

RESEARCH

Open Access



# A comparison study of adaptive scale estimation in correlation filter-based visual tracking methods

Z. L. Wang\*  and B. G. Cai\*

## Abstract

Recently, discriminative correlation filter-based method becomes one of the popular directions in the field of visual tracking because of its computational efficiency and excellent performance, which make it especially suitable for real-time application. Most of them are focused only on the transition estimation. However, accurate scale estimation of the target plays a very important role in long-term tracking task and is still a challenging problem. The principle of CF-based visual tracking is introduced first. The approaches of adaptive scale estimation in correlation filter-based visual tracking methods are summarized in this paper, and their performances are analyzed by experiment comparison. The works here can provide a better understanding on the scale estimation problem for correlation filter-based visual tracking. Furthermore, maybe with the same strategy, other factors in visual tracking, such as appearance variation, can be integrated into the framework to improve the performance of correlation filter-based method.

## Introduction

Visual object tracking is one of the core problems of computer vision and used in a wide range of applications, such as intelligent human–computer interaction, security, video surveillance and analysis, compression, augmented reality, traffic control, medical imaging and video editing [1]. It also forms a basic part of higher-level vision tasks such as scene analysis and behavior recognition. Although visual object tracking has been studied for several decades and considerable progress has been made in recent years [2, 3], robust visual object tracking is still an open research problem in the field of computer vision, and there are some challenging factors for visual tracking, such as appearance changing, scale variations, occlusions, motion blur, fast motion, some factors caused by the motion between the object and camera and some others come from the environment, such as illumination change.

In view of its wide range of applications, considerable works in the field of object tracking have been done during the past few decades, and Ref. [4] makes an

insightful review on this topic. Generally speaking, the existing tracking approaches can be classified into two groups according to the appearance model, discriminative model-based or generative model-based. Generative model-based trackers aim to build the metric model using, e.g., statistical models or templates to search the most similar patches for the tracked object [5–7]. On the other hand, discriminative model-based methods usually employ the binary classifier or machine learning techniques to distinguish the tracked object from the background. Some classifiers, such as support vector machine (SVM), structured output SVM [8], ranking SVM [9], boosting, semi-boosting and online multi-instance boosting [10], have been proposed for object tracking. SCM [11] even combines the discriminative classifier and generative model to achieve the high accuracy and robustness. But it involves with the heavy computational cost, which hinders its capability on real-time applications.

Recently, discriminative correlation filter (DCF) has successfully been applied to visual tracking [12–15] and the result performance is impressive, especially on its efficiency. As described in convolution theorem, the correlation in time domain corresponds to an element-wise multiplication in Fourier domain. Thus, the idea in

\*Correspondence: zlwang@bjtu.edu.cn; bgcai@bjtu.edu.cn  
Beijing Jiaotong University, Beijing 100044, China

nature of correlation filter is that the correlation can be calculated in Fourier domain in order to avoid the time-consuming convolution operation. Meanwhile, the correlation filter is treated as similarity measurement between the two signals in signal processing, which gives a reliable distance metric and explains the reason of the promising performance achieved by the proposed approaches. The correlation-based trackers learn a DCF for locating the target in each new frame. Some works are focused on the basic conceptions of correlation filter and exploiting the circulant structure. For example, Bolme et al. [15] trained the filter by minimizing the total squared error between the actual and the desired correlation output on a set of sample grayscale patches. By using circular correlation, the authors showed that the resulting filter can be computed efficiently using only FFTs and point-wise operations. Henriques et al. [16] further showed that the DCF formulation equivalently can be cast as learning a least squares regressor (ridge regression) on the set of all cyclic shifts of the involved training sample patches. This formulation was then used to introduce fast kernelized correlation filters. Some other basic works are conducted in [17].

Several works have recently addressed the generalization of a DCF tracker. Galoogahi et al. [18] extend the DCF with multi-channel filter and use HoG feature in CF. To improve the stability of correlation filter output, the color information is integrated in [12, 19]. However, because the multi-channel and multi-property of the target are employed in the CF, this kind of filter cannot directly apply to the online tracking problem. Alternatively, approximate formulations for learning multi-channel filters have been investigated for visual tracking [12, 13]. Danelljan et al. [12] introduced an adaptive feature dimensionality reduction technique to reduce the computational cost while preserving tracking performance.

Experiments with the benchmark dataset both in OTB [20] and VOT [21] show that DCF-based visual trackers present excellent performance, such as the capability of accurate target localization even in many different challenging tracking scenarios. Particularly, these trackers have the advantage of computing efficiency, which making them especially suitable for the real-time application. The significant gain in speed is obtained by exploiting the fast Fourier transform (FFT) at both learning and detection stages. However, most methods that employ DCFs for tracking are mainly restricted to translation estimation. This limits the performance of CF-based method; especially for long-term visual tracking, it is always the case in mobile robot and visual surveillance.

For long-term visual tracking task, many factors may affect the performance of the trackers, such as illumination variation, occlusion, scale changing and

disappearance/reappearance, and the DCF-based tracker may imply poor performance under these situations. In this paper, we will focus on the scale estimation issue of DCF-based visual tracking. It is one of the most important factors in long-term visual tracking and our works also confirmed it. The nature for this is that, for the long-term visual tracking of discriminative model-based method, a big well-known issue is the stability–plasticity dilemma [22, 23]. That is, if we use some stable samples, such as the target assigned in the first frame, to train the classifier, then the tracker is unlikely to drifting and more robust to occlusions. However, if the target appearance variation is not taken into account in this case, the tracker is doomed to work not well in long-term visual tracking process. Furthermore, the capability of accurately retrieving the target scale is beneficial in many tracking applications. Here, we first make a short summary on the existing scale estimation method, and then, experiment comparisons among these methods have been conducted to get a deep insight on this issue. And some problems on the benchmark dataset are discussed.

### Scale estimation in DCF-based visual tracking

The importance of accurate scale estimation for visual tracking has been shown in many works, especially in Ref. [14, 24]. And several works addressed the scale estimation issues in CF-based tracking method. For example, Li et al. [25] proposed a kernelized correlation translation filter with multi-resolution extension. To solve the scale estimation problem, the target in different scales are sampled first and then resized these samples into a prefixed size, and the scale with the highest correlation score is regarded as the final result. However, to get sufficient scale accuracy, the translation filter needs to be run at several resolution layers, and this brings a higher computational cost. By incorporating context information into filter learning, Zhang et al. [17] estimate the scale variation based on consecutive correlation results. In DSST tracker [14], a HOG feature-based adaptive multi-scale correlation filter is learned to cope with the scale change problem. By learning the appearance changes caused by scale variations directly, and using fused features such as raw intensity value and HOG feature, DSST tracker can estimate the target scale adaptively and track at a higher frame rate. However, this method does not address the online model updating issue. And these correlation filter-based trackers are susceptible to drifting. Danelljan et al. [26] employed an adaptive feature dimensionality reduction method as in Ref. [27] to reduce the computational cost, while tracking the performance is preserved. A collaborative correlation tracker is proposed in [28]. The experiment results [14, 24] show that by combining the scale estimation with translation filter, their approaches

outperform 19 state-of-the-art trackers in the OTB dataset.

In this section, we first briefly introduce the basic correlation filter-based visual tracking method; then, three typical strategies for scale estimation are presented, that is, multi-resolution-based [29], joint scale space filter, iterative joint scale space [14, 24].

### Basic correlation filter-based visual tracker

Correlation filters have been used in many applications such as object detection and recognition. Since the operator is readily transferred into the Fourier domain as element-wise multiplication, correlation filters have attracted considerable attention recently to visual tracking due to its computational efficiency. Bolme et al. [15] propose to learn a minimum output sum of squared error (MOSSE) filter for visual tracking on grayscale images, where the learned filter encodes target appearance with update on every frame. Many DCF-based visual tracking methods take it as a baseline approach. The basic idea of this method is as follows.

According to the convolution theorem, the correlation becomes an element-wise multiplication in the Fourier domain. To create a fast tracker, correlation is computed in the Fourier domain with fast Fourier transform (FFT). First, the 2D Fourier transform of the input image,  $F = F(f)$ , and of the filter,  $H = F(h)$ , is computed. We use the  $\odot \leftarrow$  symbol to explicitly denote element-wise multiplication and  $* \leftarrow$  to indicate the complex conjugate; then, correlation takes the form:

$$G = F \odot H^* \quad (1)$$

The correlation output is transformed back into the spatial domain using the inverse FFT. The target location corresponds to the maximum value in the correlation output. Generally, a 2D Gaussian shape is expected for the correlation output, which peak is centered on the target in training image.

For the first frame, the filter is learned according to the provided input image and the expected correlation output, and it is 2D Gaussian output. Let us  $f_i$  is a set of training images and  $g_i$  is generated from the 2D Gaussian shape; then,

$$H_i^* = \frac{G_i}{F_i}, \quad (2)$$

where the division is performed element-wise. To find a filter that maps training inputs to the desired training outputs, for the MOSSE method, a filter  $H$  is defined as that minimizes the sum of squared error between the actual output of the convolution and the desired output of the convolution. The cost function for this optimization problem is defined as

$$\min_{H^*} \sum_i |F_i \odot H^* - G_i|^2 \quad (3)$$

By solving for  $H^*$ , a closed form expression for the MOSSE filter is found:

$$H^* = \frac{\sum_i G_i \odot F_i^*}{\sum_i F_i \odot F_i^*} \quad (4)$$

In the following frames, an online update of  $H^*$  is then performed based on that new location, such as

$$H^* = \eta \frac{G_i \odot F_i^*}{F_i \odot F_i^*} + (1 - \eta)H_{i-1}^* \quad (5)$$

or in a more practical form as in MOSSE filter as

$$H_i^* = \frac{A_i}{B_i}, \quad (6)$$

where

$$\begin{aligned} A_i &= \eta G_i \odot F_i^* + (1 - \eta)A_{i-1} \\ B_i &= \eta F_i \odot F_i^* + (1 - \eta)B_{i-1} \end{aligned}$$

The computational complexity of DCF-based tracking is  $O(N \log N)$ , where  $N$  is the number of pixels in the filter. This comes from the FFTs used during the correlation operation and the online update. The advantages of DCF-based method are easy to implement and can be just accurate and much faster. Under the framework of DCF tracking, some works try to further improve its performance by taking multi-channel features, spatial constraints into consideration. But most of these works are restricted to translation estimate, and this implies poor performance when encounter with significant variations in the target scale.

### Multi-resolution-based scale estimation

For the object detection problem, a standard approach to eliminate the scale effect is to apply a detecting process at multiple resolutions. Accordingly, to tackle the problem of the fixed template size in correlation filter tracker, Li et al. [25] proposed an effective scale-adaptive scheme. Moreover, they integrate HoG and color-naming feature together to further boost the overall tracking performance. It is called SAMF (scale-adaptive multiple-feature) tracker. SAMF is the improvement in kernel-based correlation filter; to solve the scale change issue in object tracking, a sample searching strategy is implied. Here, only the scale estimation strategy in this method is briefly introduced.

Let the template size be  $S_T = (s_x, s_y)$ , and  $s_x$  and  $s_y$  denote the horizontal and vertical size, respectively, and define a scaling pool as  $S = \{t_1, t_2, \dots, t_k\}$ . Every time, the target window size  $s_i$  in the original image space is resampled  $k$  sizes in  $\{t_i s_i | t_i \in S\}$ . These samples are resized

into the same size with fixed template  $S_T$ , to match the requirement of the element-wise dot product in correlation filter. The final response is calculated by

$$\arg \max F^{-1} \hat{f}(Z^{t_i}), \quad (7)$$

where  $z^{t_i}$  is the sample patch with the size of  $t_i s_t$ , which is resized to  $S_T$ . Since the result of the response function is a vector, the max operation is employed to find its maximum scalar. As the target movement is implied in the response map, the final displacement needs to be tuned by  $t$  to get the real movement bias. The updating procedure is almost same as other DCF-based method.

In their experiments, the scale pool is set as:

$$S = \{0.985, 0.99, 0.995, 1.0, 1.005, 1.01, 1.015\}.$$

Though only seven different scale spaces are used, and all the parameters are same for the experiments, with the benchmark dataset of VOT2014, the results are impressive. One big difference to others is that the scale estimation is included in SAMF.

### Joint scale space estimation

The fused feature-based translation estimation and the scale estimation are separately processed in SAMF. The final results of translation estimation will be tuned to get a more accurate result. Instead of estimating the translation and scale separately, joint scale space-based method tries to jointly estimate the translation and scale of the target. It is achieved by computing the correlation scores in a box-shaped region of a scale pyramid representation. Both translation and scale estimates are then achieved by maximizing this score.

To update the joint scale space filter, a feature pyramid in a rectangular area around the given target location is first constructed. As shown in Fig. 1, the feature pyramid is constructed such that the target size at the current scale corresponds to the spatial filter dimensions  $M * N$ . The training sample  $f_t$  is set to the rectangular cuboid of size  $M * N * S$  centered around the target location and scale. Here,  $S$  corresponds to the number of the scale space. The joint scale space filter can be updated with formula (6), using a three-dimensional Gaussian function as the desired correlation output  $g$ .

Obviously, joint scale space-based method suffers from the computational cost and is not suitable for real-time application. Another issue is because the feature pyramid at the detection step is constructed around the predicted target location. This might result in an inclusion of a shearing component in the transformation relating the test sample  $z_t$  to the feature pyramid constructed around the actual target center. The shearing effect is caused by errors in the predicted target location. This significantly

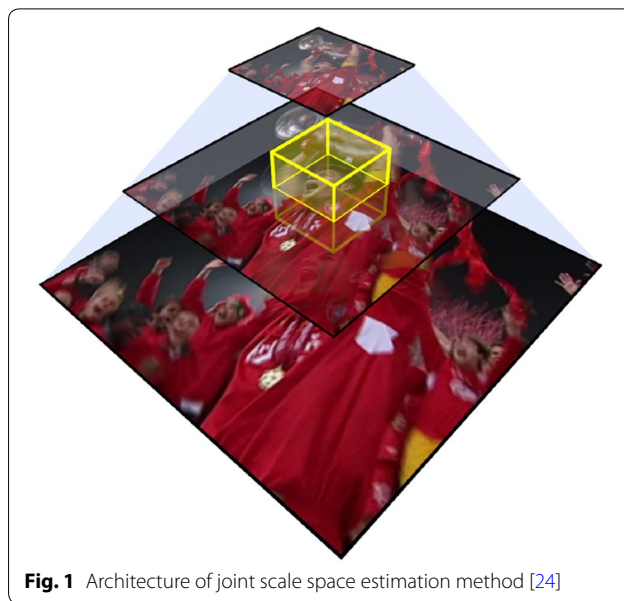


Fig. 1 Architecture of joint scale space estimation method [24]

affects the performance of the joint scale space filter by introducing a bias in the translation estimate.

### Iterative scale space estimation

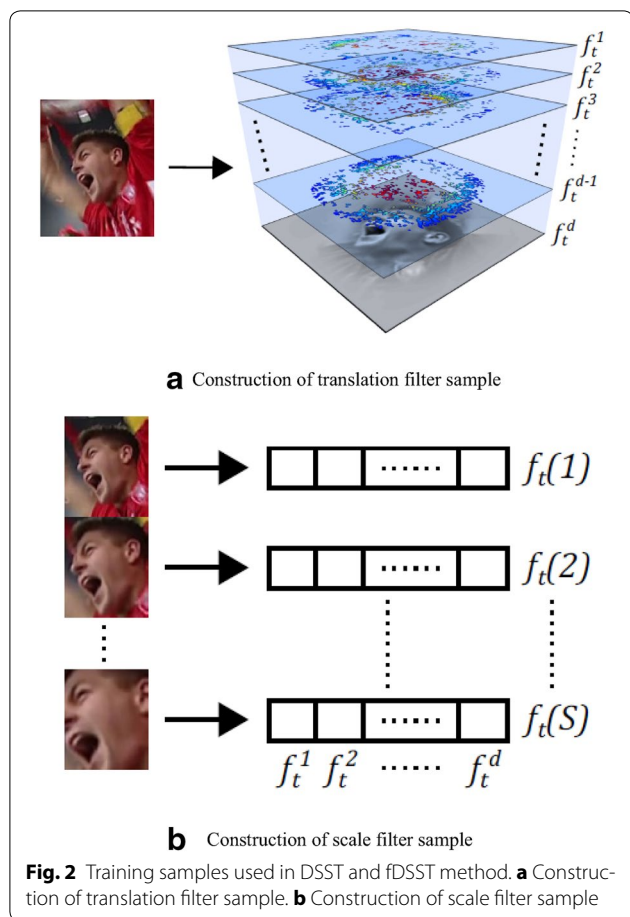
To reduce the impact of the scale space shearing distortion, the iterative scale space filter strategy can be employed. In this method, given a new frame, first using the previous target location and scale for the filter to estimate the translation, generally, a standard translation filter is used. Then the scale estimation is something a little like multi-resolution method, which uses a search strategy in scale space, and the scale is correspondent to the maximum correlation score. This procedure is performed iteratively until the convergence is achieved.

Typically, based on the observation that the target scale variation between two frames is small compared to the change in translation, the translation filter  $h_{t,trans}$  is carried out first to get the new target location; then, scale filter  $h_{t,scale}$  is applied. The test sample for scale estimation  $h_{z,scale}$  is extracted from the new location. In many cases, the iterative step may be not necessary, just as in discriminative scale space tracking (DSST) method [24].

As shown in Fig. 2, the DSST method uses a 2D multi-channel features for translation filter and a separate 1D scale filter for scale estimation. To construct the training sample  $f_{t,scale}$ , the features are extracted using variable patch sizes centered around the target. Let  $P \times R$  denote the target size in the current frame and  $S$  the size of the scale filter. For each

$$n \in \left\{ \left\lfloor -\frac{S-1}{2} \right\rfloor, \dots, \left\lfloor \frac{S-1}{2} \right\rfloor \right\},$$





an image patch  $I_n$  of size  $a^n P \times a^n R$  centered around the target. Here,  $a$  denotes the scale factor between feature layers. The training sample at scale level  $n$  is  $f_{t,\text{scale}}(n)$ , set to the  $d$ -dimensional feature descriptor of  $I_n$ . As shown in Fig. 2b, a 1D Gaussian is implied as the desired correlation output  $g$ . The updating of scale filter  $h_{t,\text{scale}}$  is like Eq. (6) with the new sample  $f_{t,\text{scale}}$ .

To estimate the translation of the target, the standard translation filter with raw pixel value and HoG feature are used in DSST [14]. To reduce the computational cost of the DSST tracker, the authors apply PCA-HoG for translation filter learning. Same to the translation filter, compressed scale filter is used without any loss of information (fast DSST, fDSST for briefly). Compared with SAMF method which only uses 7 scale space level, fDSST and DSST use  $S = 33$  and  $a = 1.02$ . This can cover a larger-scale range and more accurate results of scale estimation than in SAMF can be achieved.

Among these three strategies for scale estimation, multi-resolution-based method is simple, but the computing load is higher. Joint scale space estimation is more efficient than the multi-resolution-based method.

Iterative joint scale estimation is the most efficient one, though compressive data are used for both the translation and scale estimation, but there is no much information lost.

### The comparison experiments

Though there are few papers which take the scale estimation into consideration in correlation filter-based visual tracking, their results show that the performance of these methods is impressive. It is necessary to make a comprehensive study on the scale estimation issue.

According to the previous section, there are three strategies for scale estimation in correlation filter-based visual tracking. Actually, due to the computational cost of joint scale space estimation, it lost the advantages of the original CF-based method. fDSST is a compressed version of DSST, but there is no much information lost according to their results. Because of this, in the comparison experiments, we just take SAMF and fDSST into consideration.

All experiments are performed on an Intel Duo P8600 2.4 GHz CPU with 8 GB RAM. For the standard DSST method, the default parameter values are  $a = 1.02$ ,  $S = 33$ , and the standard deviation in the scale dimension of the desired correlation output  $g$  is set to 1/16 times the number of scales  $S$ . For the fDSST, the 32-dimensional HOG and intensity combination is reduced to 18 dimensions in our experiments. The dimensionality of the scale features from  $d \approx 1000$  to only  $S = 17$  dimensions. The parameter values of SAMF for all videos are set to the same, a Gaussian kernel type and HoG-Color feature type are used, the cell size is 4, and nine orientations are used for HoG. The padding size is set to 1.5.

The methods are quantitatively evaluated under MATLAB 2016b with the datasets of the online tracking benchmark (OTB) dataset [20]. Because we focus on the comparison of scale estimation performance, only the videos marked with scale variation are used. These videos are Biker, BlurBody, BlurCar2, BlurOwl, Board, Box, Boy, Car1, Car24, Car4, CarScale, ClifBar, Couple, Crossing, Dancer, David, Diving, Dog, Dog1, Doll, DragonBaby, Dudek, FleetFace, Freeman1, Freeman3, Freeman4, Girl, Girl2, Gym, Human2, Human3, Human5, Human6, Human7, Human8, Human9, Ironman, Jump, Lemming, Liquor, Matrix, MotorRolling, Panda, RedTeam, Rubik, Shaking, Singer1, Skater, Skater2, Skating1, Skating2.1, Skating2.2, Skiing, Soccer, Surfer, Toy, Trans, Trellis, Twinings, Vase, Walking, Walking2, Woman. There are totally 57 video sequences used.

Moreover, the tracking results about the three standard evaluation metrics namely overlap precision (OP), distance precision (DP) and tracking speed in frames per second (FPS) are reported in the existing literatures. So we mainly compare the performance on how large-scale

variation they can suffer from and can still work under this condition.

According to the ground-truth data, the referent scale value is proposed and can be calculated as follows,

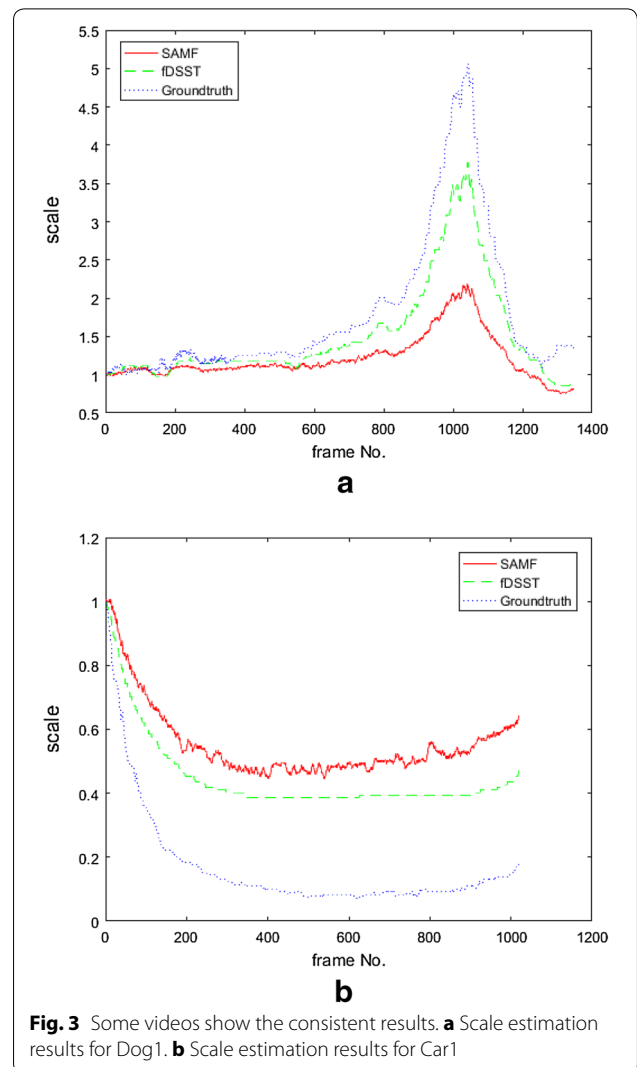
$$S_i = \frac{w_i * h_i}{w_0 * h_0}. \tag{8}$$

Here, the initial scale value  $S_0 = 1$  in first frame,  $(w_0, h_0)$  denotes the target width and height in first frame,  $(w_i, h_i)$  is the current width and height, and  $S_i$  denotes the current scale value. We have tried several other metric methods, such as the relative width/height change, but the scale value computed by (8) is seemed more practical for the comparison.

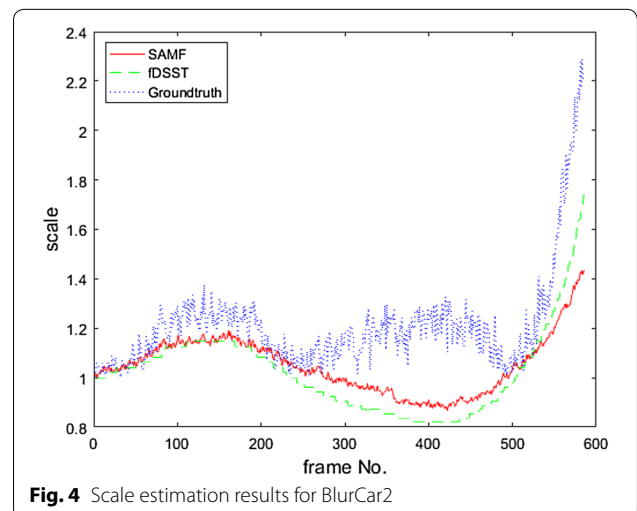
Though the video sequences used for the comparison study show a relatively large-scale variation, most of these video sequences also include other attributes. For example, video *Lemming* is included in scale variation (SV), occlusion (OCC), fast motion (FM) and out-of-plane rotation (OPR) [20]. And, moreover, different visual trackers show different performances on these attributes, so the experiment results on these video sequences are not the same. But for most of these videos, both trackers show the consistent scale change trends. Because of this, we only list some typical cases and the causes for the results are analyzed.

For the consistent results, we select the video Dog1, Car1 as examples, and the experiments show that the scale estimation results of SAMF and fDSST are consistent with the ground truth. As shown in Fig. 3, the blue dotted line is for the ground truth, the green dashed line is the results of fDSST, and the solid red line is the results of SAME. Although the magnitudes of scale values are not same, it may be caused by the different metric measurement, but the trends of the curve are almost the same.

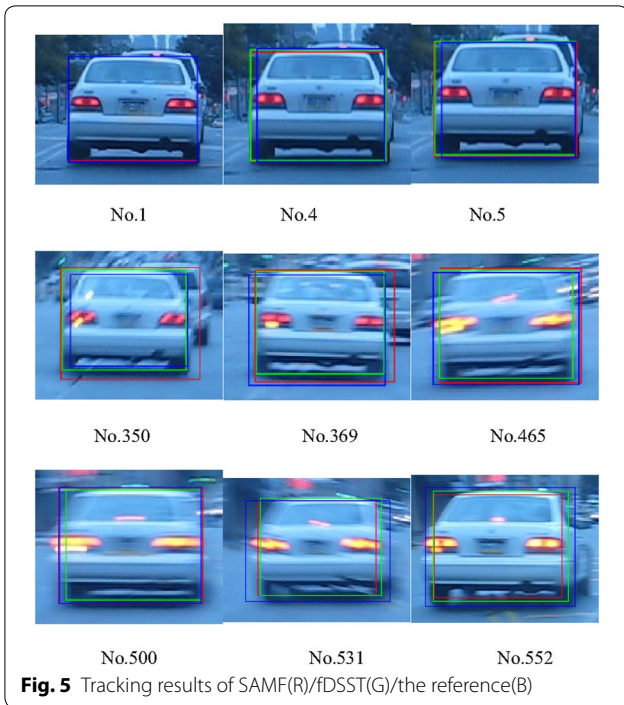
The experiment results with videos, such as BlurCar2, as shown in Fig. 4, and the results of SAMF and fDSST show a general resemblance with the ground truth, but small differences in some local sections, for example, from frame number 300–500 in Fig. 4, exist among them. For these figures, we follow the same color label rules; that is, the red color (R) indicates the result of SAME, the green color (G) presents the output of fDSST, and the blue color (B) represents the ground truth. We pick out some frames from BlurCar2 and show the tracking outputs in Fig. 5. In this experiment, we can see that the ground truth may be not accurate enough, as indicated in frame no. 5. On the other hand, because of the motion blur, it is hard to say which one is more accurate than the others, as indicated in frame no. 369/465/531. In this case, the output of SAM/fDSST and the ground truth are basically same, but there are small differences which may result from inaccuracy of the ground truth.



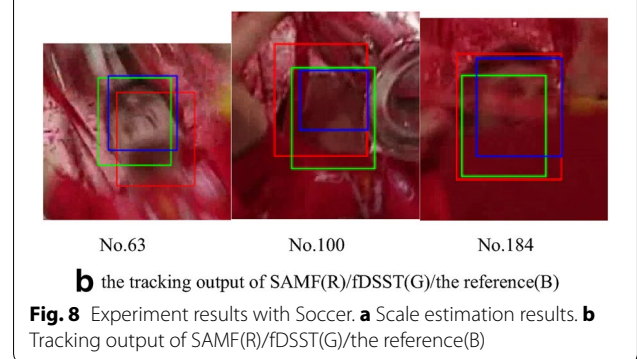
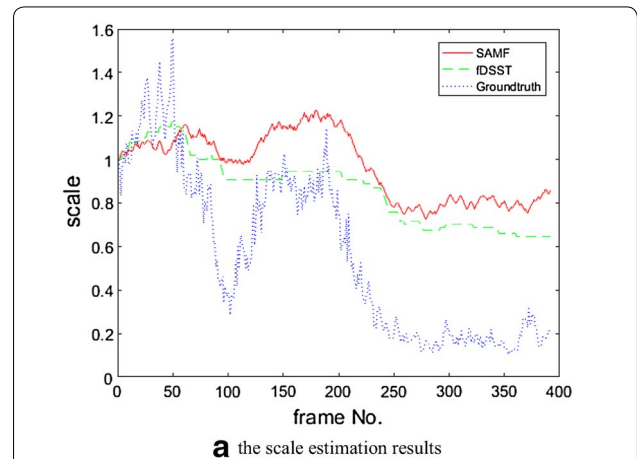
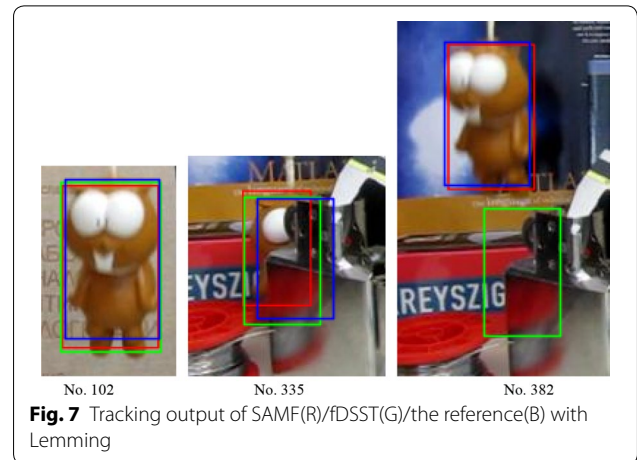
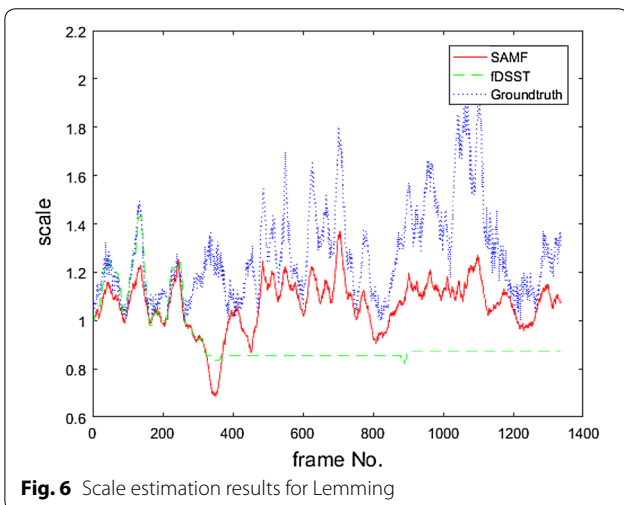
**Fig. 3** Some videos show the consistent results. **a** Scale estimation results for Dog1. **b** Scale estimation results for Car1



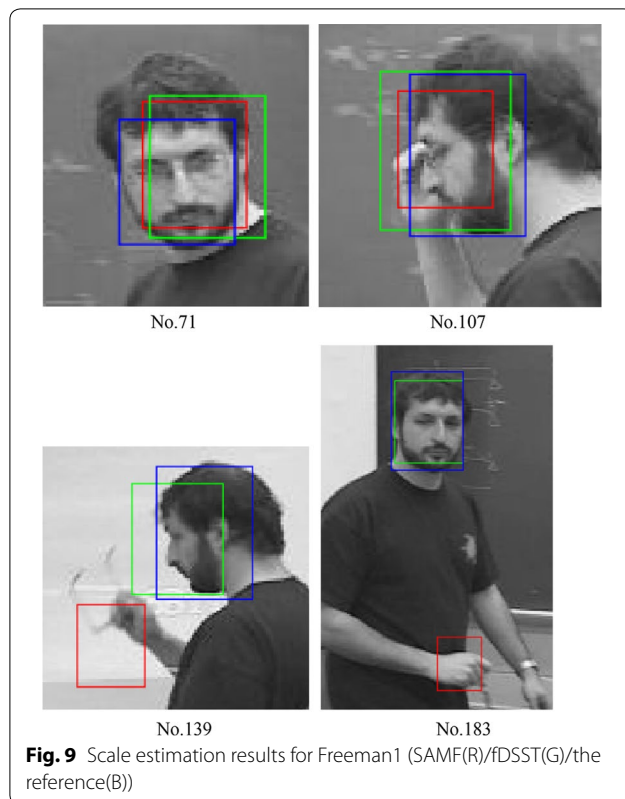
**Fig. 4** Scale estimation results for BlurCar2



For some videos, such as Lemming and Soccer, the result of SAMF is outperformed over fDSST, as illustrated in Fig. 6. In this experiment, we found that fDSST is failed since frame no. 335, as shown in Fig. 7, because of the occlusion. But SAMF can track the Lemming continuously after the target reappears. It may be benefit from the usage of color-naming feature. Figure 8 shows the results for Soccer, and it can confirm that by making use of color-naming feature, the output of CF-based tracking is more stable.



For some videos such as Freeman1, the results of fDSST are better than SAME, as shown in Fig. 9. In this experiment, we found that SAMF is failed after frame no. 139, as shown in Fig. 9b, when Freeman takes off his glass. It is interesting that in this case, however, fDSST can work well. We found that because this video is a gray-level image sequence, this makes the color of the hand



and face look like the same. But the color-naming feature is a very important part in SAMF tracker, and the tracker is distracted by hand this time.

### Conclusion and discussions

In this paper, several adaptive scale estimation methods based on correlation filter are analyzed and compared. Both the SAMF and fDSST show their high performance even in some challenging scenarios. The SAMF algorithm integrates the HoG with color-naming feature in a multi-resolution framework, and the computational cost is larger than fDSST, but it shows high performance when the color information is available, such as Soccer and Lemming. It can work more stable and accurate than fDSST and even recover from the occlusion sometime. On the other hand, when only gray-level image is available, the color-naming feature will lose the advantages, because compressive HoG is employed, and fDSST tracker is more efficient. Which tracker is suitable for the application will depend on the efficiency and the accuracy. Both trackers are capable of adaptive scale estimation and greatly improve the tracker's performance.

But it is interesting that, for the experiment with Freeman1, the tracker is distracted by the hand color. The reason for this will be further explored. Moreover, how to take the two approaches advantages to cope with the

challenging tracking scenarios is the scope of future works.

### Authors' contributions

The work presented here was carried out in collaboration between all authors. ZLW made the research theme designed methods, carried out the experiments, analyzed the data, interpreted the results and wrote the paper. BGC worked on associated data analysis, reviewed and edited the manuscript. All authors read and approved the final manuscript.

### Competing interests

The authors declare that they have no competing interests.

### Availability of data and materials

Not applicable.

### Funding

This work was supported in part by the Natural Science Foundation of China under Grant No. 61573057 and partly by the National Science and Technology Support Program (2015BAF08B01).

### Publisher's Note

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

Received: 30 August 2017 Accepted: 24 October 2017

Published online: 02 November 2017

### References

- Mihaylova L, Brasnett P, Canagarajan N, Bull D. Object tracking by particle filtering techniques in video sequences. *Adv Chall Multisens Data Inf.* 2007;8:260–8.
- Yang H, Shao L, Zheng F, Wang L, Song Z. Recent advances and trends in visual tracking: a review. *Neurocomputing.* 2011;74(18):3823–31. <https://doi.org/10.1016/j.neucom.2011.07.024>.
- Smeulders AWM, Chu DM, Cucchiara R, Caldera S, Dehghan A, Shah M. Visual tracking: an experimental survey. *IEEE Trans Pattern Anal Mach Intell.* 2014;36(7):1442–68. <https://doi.org/10.1109/TPAMI.2013.230>.
- Yilmaz A, Javed O, Shah M. Object tracking: a survey. *ACM Comput Surv.* 2006;38(4):81–93. <https://doi.org/10.1145/1177352.1177355>.
- Sevilla-Lara L, Learned-Miller EG. Distribution fields for tracking. *Comput Vis Pattern Recognit.* 2012;157(10):1910–7. <https://doi.org/10.1109/CVPR.2012.6247891>.
- Bao C, Wu Y, Ling H, Ji H. Real time robust l1 tracker using accelerated proximal gradient approach. *Comput Vis Pattern Recognit.* 2012. <https://doi.org/10.1109/CVPR.2012.6247881>.
- Chen D, Yuan Z, Wu Y, Zhang G, Zheng G. Constructing adaptive complex cells for robust object tracking. In: *International conference on computer vision.* 2013. <https://doi.org/10.1109/ICCV.2013.142>.
- Hare S, Saffari A, Torr PHS. Struck: structured output tracking with kernels. *IEEE Trans Pattern Anal Mach Intell.* 2015;38(10):2096–109. <https://doi.org/10.1109/TPAMI.2015.2509974>.
- Bai Y, Tang M. Robust tracking via weakly supervised ranking SVM. *Comput Vis Pattern Recognit.* 2012. <https://doi.org/10.1109/CVPR.2012.6247884>.
- Babenko B, Yang MH, Belongie S. Robust object tracking with online multiple instance learning. *IEEE Trans Pattern Anal Mach Intell.* 2011;33(7):1619–32. <https://doi.org/10.1109/TPAMI.2010.226>.
- Zhong W, Lu H, Yang MH. Robust object tracking via sparsity-based collaborative model. *Comput Vis Pattern Recognit.* 2012. <https://doi.org/10.1109/CVPR.2012.6247882>.
- Danelljan M, Shahbaz KM, Felsberg M, Weijer J. Adaptive color attributes for real-time visual tracking. *Comput Vis Pattern Recognit.* 2014. <https://doi.org/10.1109/CVPR.2014.143>.



13. Henriques J, Caseiro R, Martins P, Batista J. High-speed tracking with kernelized correlation filters. *IEEE Trans Pattern Anal Mach Intell.* 2015;37(3):583–96. <https://doi.org/10.1109/TPAMI.2014.2345390>.
14. Danelljan M, Hager G, Shahbaz K F, Felsberg M. Accurate scale estimation for robust visual tracking. *Proceedings of the British machine vision conference.* 2014. p. 65.1–11.
15. Bolme DS, Beveridge JR, Draper BA, Lui YM. Visual object tracking using adaptive correlation filters. *Comput Vis Pattern Recognit.* 2010;2010:2544–50. <https://doi.org/10.1109/CVPR.2010.5539960>.
16. Henriques J, Caseiro R, Martins P, Batista J. Exploiting the circulant structure of tracking-by-detection with kernels. In: *European conference on computer vision.* 2012. p. 702–15.
17. Zhang K, Zhang L, Yang MH, Zhang D. Fast tracking via spatio-temporal context learning. In: *European computer vision conference.* 2014. p. 127–41.
18. Galoogahi H, Sim T, Lucey S. Multi-channel correlation filters. *International conference on computer vision.* 2013. p. 3072–79. <https://doi.org/10.1109/ICCV.2013.381>.
19. Bertinetto L, Valmadre J, Golodetz S, Miksik O, Philip Torr H S. Staple: complementary learners for real-time tracking. In: *Computer vision and pattern recognition.* 2016. <https://doi.org/10.1109/CVPR.2016.156>.
20. [http://cvlab.hanyang.ac.kr/tracker\\_benchmark/datasets.html](http://cvlab.hanyang.ac.kr/tracker_benchmark/datasets.html). Accessed 20 Aug 2017.
21. <http://www.votchallenge.net/>. Accessed 20 Aug 2017.
22. Matthews I, Ishikawa T, Baker S. The template update problem. *IEEE Trans Pattern Anal Mach Intell.* 2004;26(6):810–5. <https://doi.org/10.1109/TPAMI.2004.16>.
23. Santner J, Leistner C, Saffari A, Pock T, Bischof H. PROST: parallel robust online simple tracking. In: *IEEE conference on computer vision and pattern recognition.* 2010. p. 723–30. <https://doi.org/10.1109/CVPR.2010.5540145>.
24. Danelljan M, Hager G, Khan FS, et al. Discriminative scale space tracking. *IEEE Trans Pattern Anal Mach Intell.* 2017;39(8):1561–75.
25. Li Y, Zhu J. A scale adaptive kernel correlation filter tracker with feature integration. In: *European conference on computer vision workshops.* 2014. p. 254–265. [https://doi.org/10.1007/978-3-319-16181-5\\_18](https://doi.org/10.1007/978-3-319-16181-5_18).
26. Danelljan M, Häger G, Fahad K, Michael F. Discriminative scale space tracking. *IEEE Trans Pattern Anal Mach Intell.* 2016;39(8):1561–75. <https://doi.org/10.1109/TPAMI.2016.2609928>.
27. Felzenszwalb P, Girshick R, McAllester D, Ramanan D. Object detection with discriminatively trained part-based models. *IEEE Trans Pattern Anal Mach Intell.* 2010;32(9):1627–45. <https://doi.org/10.1109/TPAMI.2009.167>.
28. Zhu G, Wang J, et al. Collaborative correlation tracking. In: *British machine vision conference.* 2015. p. 184.1–12.
29. Bibi A, Mueller M, Ghanem B. Target response adaptation for correlation filter tracking. In: *European conference on computer vision.* 2016. p. 419–33. [https://doi.org/10.1007/978-3-319-46466-4\\_25](https://doi.org/10.1007/978-3-319-46466-4_25).

Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

---

Submit your next manuscript at ► [springeropen.com](http://springeropen.com)

---