A Comparison Study on the Adaptive Scale Estimation of Correlation Filter-based Visual Tracking Methods *

Z. L. Wang, B. G. Cai, Member, IEEE, Y. L. Wang, G. Y. Lv, R. C. Shan, M. Zhang

Abstract— Recently, discriminative correlation filter (DCF) based visual tracker shows the advantages of being computationally efficient and excellent performance, which make them especially suitable for real-time application. On the other hand, accurate scale estimation of a target plays a very important role in visual object tracking and is obviously a challenging task for tracking. The approaches of adaptive scale estimation in correlation filter based visual tracking methods are summarized in this paper, and their performances are analyzed by experimental comparison. The works here can provide a better understanding on the scale estimation problem. Furthermore, maybe with the same strategy, other factors in visual tracking, such as appearance variation et al, can be integrated into the framework to improve the performance of correlation filter-based method.

I. INTRODUCTION

Visual object tracking is one of the core problems of computer vision and widely 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, video editing [1], et al. 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, 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, some others come from the environment, such as illumination changes.

Considerable works in the field of object tracking has been done during the past few decades, reference [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][6][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 adapted for object tracking. SCM [11] combines the discriminative classifier and generative model to achieve the high accuracy and robustness. However, 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][13][14][15]. As described in convolution theorem, the correlation in time domain corresponds to an element-wise multiplication in Fourier domain. Thus, the idea in 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 previous approaches.

To locate the target in every frame, a correlation process in Fourier domain is carried out and the position of the peak value in the response map is taken as the target position for the correlation-based trackers. Bolme et al. [15] first introduces the CF into visual tracking, he trained the tracking filter by minimizing the output sum of squared error between the actual and the desired correlation output on a set of gray scale samples. By using the tool of circulant matrix, the resulting filter can be computed efficiently using point-wise operations in frequent domain. Henriques et al [16] further showed that the DCF formulation can be equivalently cast as learning a ridge regression on the set of all cyclic shifts of the involved training sample patches. This formulation was then further extended to the fast kernelized correlation filters.

Several works have recently addressed the generalization of a DCF tracker. Galoogahi et al [17] extend the DCF with multi-channel filter. However, 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 and VOT show that discriminative correlation filter (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

\*Research supported by the National Natural Science Foundation of China (Grant No. 61573057) and partly by National Science and technology supporting project under Grant No.2015BAF08B01.

Z. L. Wang and B. G. Cai are with the Beijing Jiaotong University, Beijing, 100044 China (corresponding author: +86610-51684361; fax: +8610-51687111; e-mail: [2lwang,bgcai]@bjtu.edu.cn).

Yinlin Wang, Guoyan Lv, Rongchen Shan and Min Zhang are with CSR Qindao Sifang Co. Ltd. Qindao 266111, China (e-mail: [wyil, lvguoyan]@cqsfc.com).
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.

For long term visual tracking task in mobile robot application, if there are significant variations in the target scale, the DCF-based tracker may imply poor performance. So in this paper, we focus on the scale estimation of DCF-based visual tracking. It’s one of the most important factors in visual tracking. Furthermore, the capability of accurately retrieving the target scale is beneficial in many tracking applications. In this paper, we first make a short summary on the existing scale estimation method, and then an experimental comparison among these methods have been conducted to get a deep insight on this issue.

II. SCALE ESTIMATION IN DCF-BASED VISUAL TRACKING

The importance of accurate scale estimation for visual tracking has been shown in many works, especially, in reference [18]. By combining the scale estimation with translation filter, this approach outperforms 19 state-of-the-art trackers in the OTB dataset. In this section, we first briefly review the basic correlation filter-based visual tracking method, then several typical approaches of scale estimation are introduced. Currently, based on correlation filter, three main strategies for scale estimation are proposed, multi-resolution based [19], joint scale space filter, iterative joint scale space [14][18].

A. 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 gray-scale 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: \( H = F(h) \) are computed. We use the \( \odot \) symbol to explicitly denote element-wise multiplication and * to indicate the complex conjugate, then correlation takes the form:

\[
G = F \odot H^* \tag{1}
\]

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, that is 2D Gaussian output. Let’s \( f_i \) is a set of training images, \( g_i \) is generated from the 2D Gaussian shape, then

\[
H_i^* = \frac{g_i}{F_i} \tag{2}
\]

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

\[
\min_{H^*} \sum_i |F_i \odot H^* - g_i|^2 \tag{3}
\]

By solving for \( H^* \) a closed form expression for the MOSSE filter is found:

\[
H^* = \sum_i \frac{G_i}{F_i} \odot F_i^* \tag{4}
\]

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

\[
H_i^* = \eta \frac{G_i}{F_i} \odot F_i^* + (1-\eta)H_{i-1}^* \tag{5}
\]

Or in a more practical form as in MOSSE filter as

\[
H_i^* = \frac{A_i}{B_i} \tag{6}
\]

where

\[
A_i = \eta G_i \odot F_i^* + (1-\eta)A_{i-1}
\]

\[
B_i = \eta F_i \odot F_i^* + (1-\eta)B_{i-1}
\]

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, can be just as 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, this implies poor performance when encounter with significant variations in the target scale.

B. Multi-resolution based scale estimation

For the object detection problem, a standard approach to eliminate the scale effect, is to apply a detection at multiple resolutions. Accordingly, to tackle the problem of the fixed template size in correlation filter tracker, Li et al [20] 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
of 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 is \( S_r = (s_x, s_y) \), \( s_x \) and \( s_y \) denote the horizontal and vertical size respectively, and define a scaling pool as \( S = \{ t_1, t_2, \ldots, t_r \} \). Every time, the target window size \( s_x \) in the original image space is resampled \( k \) sizes in \( \{ t_i, t_j \in S \} \). These samples are resized into the same size with fixed template \( S_r \), to match the requirement of the element-wise dot-product in correlation filter. The final response is calculated by

\[
\arg \max \ F^{-1} \hat{f}(Z^r),
\]  

(7)

where \( Z^r \) is the sample patch with the size of \( t_i s_x \), which is resized to \( S_r \). 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.

C. Joint scale space estimation

The fused feature based translation estimation and the scale estimation are separated 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 try 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 showed in figure 1, the feature pyramid is constructed such that the target size at the current scale corresponds to the spatial filter dimensions \( M^sN^s \). The training sample \( f_t \) is set to the rectangular cuboid of size \( M^sN^sS \) 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 as same as in (6), using a 3-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 that, 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 \( z_t \) with the feature pyramid constructed around the actual target center. The shearing effect is caused by errors in the predicted target location. This significantly affects the performance of the joint scale space filter by introducing a bias in the translation estimate.

![Figure 1. The architecture of joint scale space estimation method [18]](image)

D. 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 use 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 \( h_{t,\text{trans}} \) is carried out first to get the new target location, then scale filter \( h_{t,\text{scale}} \) is applied. The test sample for scale estimation \( z_{t,\text{scale}} \) is extracted from the new location. In many cases, the iterative step may be not necessary, just as in discriminative scale space tracking method (DSST) [18].

As showed in figure 2, the DSST method use a 2D multi-channel features for translation filter, and a separate 1D scale filter for scale estimation. To construct the training sample \( f_{t,\text{scale}} \), 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 \( S \) is the size of the scale filter. For each

\[
n \in \{ \text{floor}(\frac{S-1}{2}), \ldots, \text{floor}(\frac{S-1}{2})\},
\]

473
an image patch $I_a$ of size $a^P \times a^R$ centered around the
target. Here, $a$ denotes the scale factor between feature layers.
The training sample at scale level $n$ is $f_{r, \text{scale}}(n)$, is set to the
$d$-dimensional feature descriptor of $I_a$. As illuminated in
figure 2.b, a 1D Gaussian is implied as the desired correlation
output $g$. The updating of scale filter $h_{r, \text{scale}}$ is like equation (6)
with the new sample $f_{r, \text{scale}}$.

![Translation filter sample](image1)

a). Translation filter sample

![Scale filter sample](image2)

b). Scale filter sample

Figure 2. Training samples used in DSST and fDSST method

To estimate the translation of the target, the standard
translation filter with raw pixel value and HoG feature are used in
DSST [14]. To reducing the computational cost of the DSST, 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 spaces, fDSST and
DSST use $S = 33$ and $a = 1.02$. This can cover a larger scale
range and more accurate scale estimation results than in SAMF.

III. THE COMPARISON EXPERIMENTS

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

According to last section, there are three kinds of
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
method. Because of this, in the comparison experiments, we
just take SAMF and fDSST into consideration. fDSST is a
compressed version of DSST, but there is no much
information lost according to their results.

All experiments are performed on an Intel Duo P8600
2.4GHz CPU with 8GB RAM. For the standard DSST method,
the default parameter values are $a = 1.02, S = 33$, 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, nine orientations are used for HoG. The padding size
is set to 1.5.

The methods are quantitatively evaluated under matlab2016b
with the datasets of the Online Tracking Benchmark (OTB)
dataset [21]. 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,
Skating, Soccer, Surfer, Toy, Trans, Trellis, Twinnings, 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 calculated as follows,

$$ S' = \frac{w_1 \cdot h_1}{w_0 \cdot h_0} . \tag{8} $$

Here, the initial scale value $S_1' = 1$ in first frame, $(w_1, h_1)$
denote the target width and height in first frame, $(w_0, h_0)$ is
the current width and height, $S_{n}$ 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 more practical for our comparison study.

The experiments show that the scale estimation results of
SAMF and fDSST are consistent with the ground truth for some
videos, such as Dog1, BlurBody, CarScale, Doll. As
showed in figure 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 SAMF. Although the magnitudes of scale
values are not the same, it may be caused by the different scale
estimation methods, but the trends of the curve are almost the
same.
The experiment results with videos, like BlurCar2, as showed in figure 4, 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 to 500 in figure 4, exist among them. For these figures, we follow the same color rules, that is, the red color (R) indicates the result of SAMF, 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 figure 5. We can see that, the ground truth may be not accurate, as indicated in frame no.5. On the other hand, because of the motion blur, it’s hard to say which one is more accurate as indicated in frame no. 369/465/531.

For some videos, such as Lemming, Soccer, the result of SAMF is outperformed with fDSST, as illustrated in figure 6. In this experiment, we found that fDSST is failed after frame no 335, as showed in figure 7, because of the occlusion. But SAMF can track the lemming stably. It may be benefit from the usage of color-naming feature. Figure 8 is the results for Soccer.

For some videos like Freeman1, the results of fDSST are better than SAMF, as showed in figure 9. In this experiment, we found that SAMF is failed after frame no 139, as showed in figure 9.b, when Freeman takes off his glass. But in this case, fDSST can work well.

Different videos in SV dataset show their different properties, it results in the different performances of two scale adaptive estimation approaches.

IV. CONCLUSION AND DISCUSSIONS

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, the computational cost is larger
than fDSST, but it shows the advantages in the experiments with Soccer, Lemming, et al. For Soccer experiment, SAMF works more stable and accurate than fDSST. It can even recover from the occlusion. But it is interesting that, for the experiment with Freeman1, the tracking is distracted by the hand color. The reason for this will be further explored. Moreover, how to fuse the two approaches advantages to cope with the challenging tracking scenarios is the next works.

![Tracking result](image)

**Figure 7. Some typical results for Lemming**

![Scale estimation result](image)

**Figure 8. The experiment results with Soccer**

**Figure 9. The scale estimation results for Freeman1 (SAMF(R)/fDSST(G)/the reference(B))**

**ACKNOWLEDGMENT**

This work was supported in part by the Natural Science Foundation of China under Grant No.61573057, partly by the national science and technology support program (2015B0108B01).

**REFERENCES**

[1] L. Mihaylova, P. Brasnett, N. Canagarajah and D. Bull, “Object tracking by particle filtering techniques in video sequences,” In *Advances and Challenges in Multisensor Data and Information. NATO Security Through Science Series*. 8. Netherlands: IOS Press. 2007, pp.260–268.

[2] H. Yang, L. Shao, F. Zheng, L. Wang, and Z. Song, “Recent advances and trends in visual tracking: an review,” In *Neurocomputing*, Vol.74, No.18, pp.3823–3831, 2011.

[3] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, “Visual tracking: an experimental survey,” *IEEE Transaction on Pattern Anal. Mach. Intell.*, Vol.36, No.7, pp.1442–1468, 2014.

[4] Yilmaz, Alper, O. Javed, and M. Shah, “Object tracking: a survey,” *ACM Computing Surveys,* 38.4, pp.81–93, 2006.

[5] L. Sevilla-Lara and E. G. Learned-Miller, “Distribution fields for tracking,” in *CVPR*, 2012.

[6] C. Bao, Y. Wu, H. Ling, and H. Ji, “Real-time robust I1 tracker using accelerated proximal gradient approach,” in *CVPR*, 2012.

[7] Chen, D., Yuan, Z., Wu, Y., Zhang, G., Zheng, N.: Constructing adaptive complex cells for robust object tracking. In *ICCV*, 2013.

[8] S. Hare, A. Saffari, and P. H. S. Torr. Structured output tracking with kernels. *International Conference on Computer Vision. IEEE Trans Pattern Anal Mach Intell.*, pp.263–270, 2011.

[9] Y. Bai and M. Tang. Robust tracking via weakly supervised ranking SVM. *Computer Vision and Pattern Recognition, IEEE*, pp.1854-1861, 2012.

[10] B. Babenko, M.-H. Yang, and S. Belongie. Robust object tracking with online multiple instance learning. *IEEE Trans Pattern Anal Mach Intell.*, Vol.33, No.7 pp.1619–1632, 2011.

[11] Zhong, Wei, H. Lu, and M. H. Yang. “Robust object tracking via sparsity-based collaborative model.” In *Computer Vision and Pattern Recognition, IEEE*, pp.1838-1845, 2012.

[12] M. Danelljan, F. Shahtabz Khan, M. Felsberg, and J. van de Weijer, “Adaptive color attributes for real-time visual tracking,” in *CVPR*, 2014.

[13] J. Henriques, R. Caseiro, P. Martins, and J. Batista, “High-speed tracking with kernelized correlation filters,” *IEEE Trans Pattern Anal Mach Intell.*, 2015.

[14] M. Danelljan, G. Hagger, F. Shahtabz Khan, and M. Felsberg, “Accurate scale estimation for robust visual tracking,” in *BMVC*, 2014.

[15] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, “Visual object tracking using adaptive correlation filters,” in *CVPR*, 2010.

[16] J. Henriques, R. Caseiro, P. Martins, and J. Batista, “Exploiting the circulant structure of tracking-by-detection with kernels,” in *ECCV*, 2012.

[17] H. Galoogahi, T. Sim, and S. Lucey, “Multi-channel correlation filters,” in *ECCV*, 2013.

[18] M. Danelljan, et al. “Discriminative Scale Space Tracking.” *IEEE Transactions on Pattern Analysis & Machine Intelligence*, pp.99:1-1, 2016.

[19] A. Bibi, M. Mueller, and B. Ghanem. “Target Response Adaptation for Correlation Filter Tracking”, *European Conference on Computer Vision*, 2016.

[20] Li, Yang, and J. Zhu. “A Scale Adaptive Kernel Correlation Filter Tracker with Feature Integration.” In: *European Conference on Computer Vision Workshops*, pp. 254-265, 2014.

[21] http://cvlab.hanyang.ac.kr/tracker_benchmark/datasets.html