Visual Tracking via Correlation Filter using Luminance Histogram and Adaptive Model

Zhaoqian Tang and Kaoru Arakawa
Graduate School of Advanced Mathematical Sciences, Meiji University, Japan

Abstract— **Visual trackers based on the framework of kernelized correlation filter (KCF) need to learn information on the object from each frame, thus the state change of the object affects the tracking performances. In this paper, we propose a novel KCF tracker using luminance histogram and adaptive model, to deal with the change of the object's state. This method firstly takes skipped scale pool method which utilizes variable window size at every two frames. Secondly, the location of the object is estimated using the combination of the filter response and the similarity of the luminance histogram at multiple points in the filter response map. Thirdly, the learning rate to obtain the tracking model is adjusted, using the filter response and the similarity of the luminance histogram, considering the state of the object. Experimentally, the proposed tracker (CFHA) achieves outstanding performance for the challenging benchmark sequence (OTB100).**

I. INTRODUCTION

Visual tracking, which automatically detects moving objects in image sequences, is one of the hot topics in computer vision. Theoretical research on visual tracking has made great progress and been applied to many applications, but still, it faces enormous difficulties such as deformations, fast motions, occlusions, background clutter, and scale variations and so on [4, 5, 6, 9,11].

It is a tough problem for visual tracking to accurately track a moving object and simultaneously achieve real-time performance. Recently, the emergence of a group of correlation filters (CF) in tracking algorithms has enabled accurate and high-speed tracking [1, 2, 3, 8, 12, 13, 18].

In 2010, D. S. Bolme proposed a new type of tracking algorithm with CF, named as MOSSE[14]. MOSSE introduces correlation filtering to visual tracking for the first time and it is famous for high-speed tracking. Compared to MOSSE which utilizes sparse samples as input, Henriques et al. proposed a tracking-by-detection method (CSK) [1] which showed higher speed in visual tracking. In CSK, a novel cyclic sampling method was proposed, and a cyclic matrix is constructed using cyclic samples. Using the cyclic structure, the training of the classifier becomes quite fast in frequency-domain. At the same time, the classifier detects the object at the maximum point of the response map of CF. In [3], M. Danelljan proposed discriminative scale-space tracker (DSST) which achieves accurate scale estimation in tracking. Moreover, Yang Li proposed a scale pool technique to deal with scale variation [8]. Although scale pool technique can achieve scale adaptive tracking and improve tracking accuracy, it needs more calculation time.

To improve the robustness of trackers, robust features of the moving objects should be utilized in visual tracking. Henriques et al. proposed kernelized correlation filter tracker (KCF) which uses the histogram of gradient (HOG) to achieve state-of-the-art performance in tracking [2]. M. Danelljan extended the CSK tracker with color attributes [16]. Utilization of color attributes was shown to be effective to get excellent results for object recognition and object detection [19, 20]. Yang Li employed various powerful features to improve the performance of trackers, such as the raw grayscale, HOG and color attributes (CN) [8]. At present, conventional handcraft features (HOG and CN) [21, 22] are major features for visual tracking.

In this paper, a new method for visual tracking based on KCF is proposed to deal with the state change of the object, such as deformation, occlusion, and so on. This method is named as a correlation filter using the luminance histogram and adaptive model, CFHA for short. To realize robust detection for the moving objects with the variant state, we proposed before a tracking method based on KCF using adaptive model (KCFAMSR)[17]. The proposed CFHA newly utilizes the information on both the correlation response and the luminance histogram to estimate the location of the object in KCFAMSR. Moreover, the information at multiple points in the response map are taken into consideration

An adaptive method, such as a scale pool method and adaptive model update, which are adopted in KCFAMSR, are also included in CFHA, but they are improved as follow. First, skipped scale pool method, which apply scale pool method at every two frames is proposed in order to save computation time. Second, the model update is more precisely adjusted at each frame considering the state of the object.

Experimental results demonstrate that the proposed tracker shows the state-of-the-art performance for the standard benchmark (OTB-100) [7, 15].

II. KERNELIZED CORRELATION FILTER TRACKER (KCF)

We briefly review the principle of KCF [1][2] which is the original form of our proposed CF-based trackers.

A. *Linear Regression*

KCF estimates the location of the object by a correlation filter $\boldsymbol{w}$. Suppose that y_i is the target response of the filter for the i-th input image patch which is expressed as a vector $\boldsymbol{x}_i$, KCF obtains the filter $\boldsymbol{w}$ which approximates y_i as $\boldsymbol{w}\boldsymbol{x}_i$ by linear ridge regression as follows.

$$\min_{\boldsymbol{w}} \sum_i^n (f(\boldsymbol{x}_i) - y_i)^2 + \lambda\|\boldsymbol{w}\| \qquad (1)$$

Here $f(\boldsymbol{x}_i) = \boldsymbol{w}\boldsymbol{x}_i$ and λ is a regularization parameter.
The solution of $\boldsymbol{w}$ which satisfies (1) is obtained as

$$\boldsymbol{w} = (X^H X + \lambda I)^{-1} X^H \boldsymbol{y} \qquad (2)$$

where X is a matrix composed of rows $\boldsymbol{x}_i$'s, $\boldsymbol{y}$ a vector, the elements of which are y_i's, and I an identity matrix. X^H denotes the Hermitian transpose of X.

B. *Circulant Matrix*

KCF assumes that a cyclic shift version of $\boldsymbol{x}_i$ corresponds to a shifted image patch. Thus, image patch samples which are densely collected around the estimated location can be expressed as cyclic shifts of the base image sample $\boldsymbol{x}$. Thus, the matrix X can be expressed as a circulant matrix of the base sample as shown in Fig. 1.

Fig. 2 shows examples of the dense image patches around the base sample (0,0), obtained by shifting the base sample vertically and horizontally by one pixel. Since the shifting range becomes wide, the size of the matrix X becomes large, requiring a lot of computation time to obtain the solution $\boldsymbol{w}$.

KCF introduces the discrete Fourier transform (DFT) to reduce the computation time, using the characteristics of circulant matrices that all circulant matrices are made diagonal by DFT. The circulant matrix X can be expressed as follows,

$$X = F\mathit{diag}(\hat{x})F^H \qquad (3)$$

where F is the DFT matrix, $\hat{\boldsymbol{x}}$ is the DFT of the base sample $\boldsymbol{x}$, and $\mathit{diag}(\hat{x})$ denotes a diagonal matrix, the diagonal element of which is $\hat{\boldsymbol{x}}$.

Thus, the solution $\boldsymbol{w}$ in Eq. (2) is obtained by inverse DFT of $\hat{\boldsymbol{w}}$ which is obtained by element-wise product $\odot$ as follows.

$$\hat{w} = \frac{\hat{x} \odot \hat{y}}{\hat{x}^* \odot \hat{x} + \lambda} \qquad (4)$$

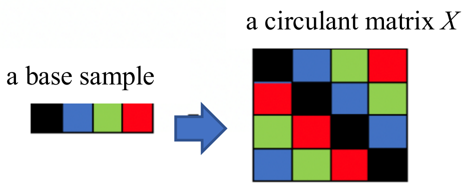

Fig.1 Circulant matrix X obtained by cyclic shift of a base sample.

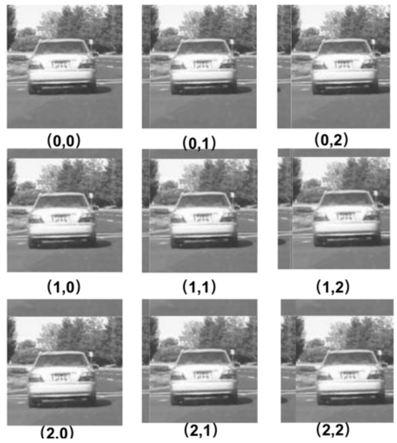

Fig.2 Example of the cyclic shifts of a base sample to approximate the dense samples around the base sample

Here, $\hat{\boldsymbol{y}}$ denotes the DFT of $\boldsymbol{y}$, and all the calculation is performed element-wisely. Since inverse matrix is not required to calculate Eq. (2), KCF can obtain the solution $\boldsymbol{w}$ with a small amount of calculation.

C. *Non-linear Regression*

The fundamental idea of KCF can be extended to no-linear regression by applying kernel trick to project the sample $\boldsymbol{x}_i$ into high-dimensional feature space $\varphi(\boldsymbol{x}_i)$. More powerful classifier can be realized by this non-linear regression. In non-linear regression, the regression function f for the input sample $\boldsymbol{z}$ becomes,

$$f(\boldsymbol{z}) = \boldsymbol{w}\boldsymbol{z} = \sum_i^n \alpha_i \kappa(\boldsymbol{z}, \boldsymbol{x}_i) \qquad (5)$$

where $\kappa(\boldsymbol{z}, \boldsymbol{x}_i)$ is a kernel function, such as Gaussian as

$$\boldsymbol{\kappa}(\boldsymbol{z}, \boldsymbol{x}_i) = \exp\left(-\frac{1}{\sigma^2}(\|\boldsymbol{z}\|^2 + \|\boldsymbol{x}_i\|^2) - 2F^{-1}(\hat{\boldsymbol{z}} \odot \hat{\boldsymbol{x}}_i)\right) \qquad (6)$$

The variable to be optimized is a vector $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ instead of $\boldsymbol{w}$. The solution of $\boldsymbol{\alpha}$ is obtained by ridge regression as follows.

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

Here K is the kernel matrix with elements $K_{ij} = \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$, which also becomes a circulant matrix, if the Gaussian kernel as Eq. (6) is adopted. Thus, Eq. (7) can be solved by making the matrix diagonal and the following solution is obtained in frequency domain.

$$\hat{\boldsymbol{\alpha}} = \frac{\hat{y}}{\hat{k}^{xx} + \lambda} \qquad (8)$$

Here, $\boldsymbol{k}^{xx}$ denotes the first row of the kernel matrix K, that is $\kappa(\boldsymbol{x}, \boldsymbol{x})$, and again a hat ˆ denotes the DFT. Also, λ is added to all the elements and the division is performed

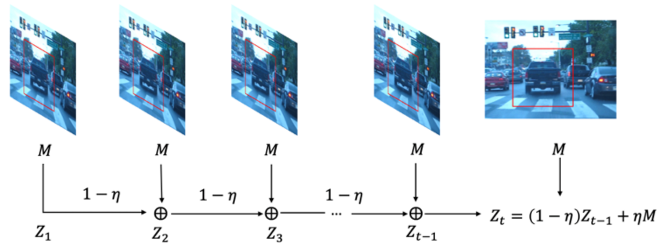

Fig.3 Update the model of the object using online learning method. This model obtains the information on the object in each frame.

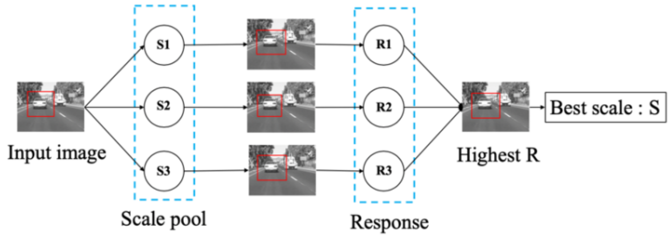

Fig.4 Tracker gets three response maps using three scales from scale pool, and according to the maximum response, the best scale is selected.

element-wisely. Computation time can be also reduced owing to the diagonalization. Finally, the regression for the input z is obtained from the inverse DFT of the following $\hat{f}(\boldsymbol{z})$.

$$\hat{f}(X) = (\hat{\boldsymbol{k}}^{zx})^* \odot \hat{\boldsymbol{\alpha}} \tag{9}$$

D. Model update

KCF tracks an object while updating the model of the object and the filter. Suppose that the visual information on the object is denoted as z_t and the filter as $\boldsymbol{\alpha}_t$ at frame t, they are updated as follows, where M is the new input information on the object and $\boldsymbol{\alpha}$ the newly obtained filter at current frame, and η is the learning rate .

$$Z_t = (1-\eta)Z_{t-1} + \eta M \tag{10}$$

$$\boldsymbol{\alpha}_t = (1-\eta)\boldsymbol{\alpha}_{t-1} + \eta\boldsymbol{\alpha} \tag{11}$$

As shown in Fig. 3, the model of frame 3 includes model information from frame 1 and frame 2. Such online learning method can reduce the effect of background and improve the robustness of the model.

III. Correlation Filter using Luminance Histogram and Adaptive Model

In CFHA which is proposed here, three types of improvement are introduced to KCF; one is skipped scale pool, one is estimating the object location using both the filter response and the luminance histogram at multiple points in the response map, and the other is adopting adaptive model update considering the state of the object.

A. Skipped Scale pool

Scale variation of the object is a tough problem in visual tracking. Scale pool is a method to adapt the tracking system to scale variation, but it is time-consuming to prepare various sizes of scales.

In this paper, we propose a new method of scale pool to solve scale variation, while saving the computation time. Since the scale of the object does not change drastically in two adjacent frames, a skipped scale pool is proposed here which applies scale pool $S = \{S_1, S_2, S_3\}$ in every two frames. Here, S_k is the value of the scale to be multiplied to the search patch size, and we set S as {0.95, 1.0, 1.05}. Object tracker obtains three response map using this scale pool. The scale which gives the maximum filter response is judged as the proper scale. The process of scale pool method is as shown in fig. 4. The scale gap in S is set larger than that of SAMF, because SAMF applies scale pool at every frame. The scale change is larger at two frames ahead in our method. In applying scale pool, when the size of the object $S_T = (Height, Width)$ is too small, the size of the search patch $\{S_T(S_i)|S_i \in S\}$ does not change at all, because the difference of the patch sizes $0.05 \times S_T$ become less than 1.0, which becomes zero in quantizing the patch size to an integer. In this case, it makes no sense to use the scale pool. In order to avoid this situation, we change the width or the height of the search patch size by at least 1. If height is less than width, patch sizes as {*Height*-1, *Width*-*Ratio*}, {*Height*, *Width*}, and {*Height*+1, *Width*+*Ratio*} are applied. Here, *Height* and *Width* are the height and the width of image patch, and *Ratio* is as follows.

$$Ratio = \frac{\max(Height, Width)}{\min(Height, Width)} \tag{12}$$

If width is less than height, width is changed by 1, and height is by *Ratio*.

B. Combination with Luminance Histogram

Correlation filter trackers belong to template matching tracking methods. Since KCF uses HOG feature to describe object, it is difficult for HOG to maintain robustness to non-rigid object deformation. When the object has deformation, the response map will change dramatically, causing tracking failure. In order to obtain the correct location, we proposed to use luminance histogram of the image as well as the response map of HOG. The luminance histogram describes gray distribution of the object, which can visually show the amount of each gray level in the image. Thus, the same object can be easily recognized by the luminance histogram, although the object gets deformed. We adopt cosine similarity [10] to obtain the luminance similarity between the histogram vectors as follows:

$$\cos\theta = \frac{\sum_{i=1}^{n}(T_i \times N_i)}{\sqrt{\sum_{i=1}^{n}(T_i)^2} \times \sqrt{\sum_{i=1}^{n}(N_i)^2}} \tag{13}$$

Where T_i is the $i-th$ element in the luminance histogram vector of the initial object, and N_i is *the i-th* element in the luminance histogram vector of the current object.

The response map directly reflects the relationship between the model and the samples, and the maximum point of the map corresponds to the location of the object. However the maximum point is sometimes ambiguous when the object changes its state, such as deformation and so on. In order to increase the reliability, we propose to select the highest four values $\{R_i | i \in \{1,2,3,4\}\}$ in the response map, and judge the location of the object from the products of R_i and the luminance histogram similarity H_i at the four positions as $Y = \{(R_i \times H_i) | i \in \{1,2,3,4\}\}$. The location of the object is obtained as the position of i which takes the maximum value in Y.

C. Adaptive Model Update

The model of the object and the filter is a significant factor for correlation filter trackers. The quality of the model directly affects the results of tracking. In this paper, we update the model adaptively by analyzing the state of the object. Fig. 5 and Fig. 6 show the maximum filter response and the similarity of luminance histogram at each frame in the image sequence [*jogging*] and [*car2*] in OTB100 [15] respectively. In Fig. 5, deformation of the object occurs in the first few frames, and accordingly the maximum value of the filter response drops drastically. However, in this case, the similarity of the luminance histogram is kept steady, because the luminance histogram does not influenced by the deformation. At this time, the state of the object is to be recognized as 'deformation' and the learning rate of the model should be increased. In the frames from number 60 to number 80, both the maximum filter response and the histogram similarity drop largely. In this case, the object is occluded by an obstacle, and the model should not be updated by the new input. When the object is rigid as cars, the maximum response and the similarity of luminance histogram are kept steady in early stage as shown in Fig. 6. However, when the illumination around the object changes in frames 60-80 and 120-140, both of these values become smaller. Especially, since luminance histogram is not robust to illumination, the value of the similarity drops largely, while the drop of the maximum filter response is not so large. In this case, the learning rate of the model should be reduced in order to avoid the effects of illumination.

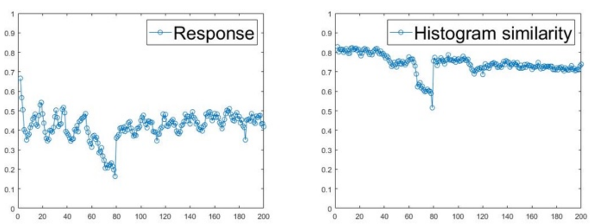

Fig.5 The maximum response (left) and the similarity of luminance histogram (right) at each frame in the image sequence *[jogging]*. The horizontal axis represents the frame number.

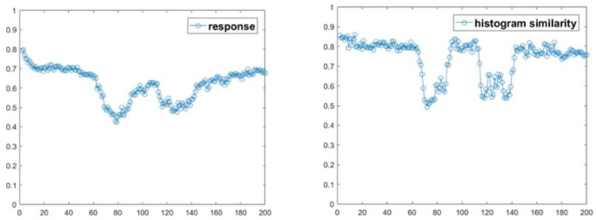

Fig.6 The maximum response (left) and the similarity of luminance histogram (right) of each frame in the image sequence *[cars]*. The horizontal axis represents the frame number.

The judgment coefficient σ is introduced here to adjust the learning rate, using the information on both the maximum filter response and the similarity of the luminance histogram, as follows.

$$\sigma = \begin{cases} 0 & |R_n - R| > 0.1, H < 0.7 \\ 1 & |R_n - R| \le 0.1, H \ge 0.7 \\ 2 & |R_n - R| > 0.1, H \ge 0.7 \\ 0.5 & |R_n - R| \le 0.1, H < 0.7 \end{cases} \tag{14}$$

Here, $|R_n\text{-}R|$ is the change of the maximum filter response and H the similarity of luminance histogram. When $\sigma = 0$, the search region is enlarged to relocate the object. The judgment coefficient σ is applied to the model update as follows.

$$Z_t = (1 - \sigma\eta)Z_{t-1} + \sigma\eta M \tag{15}$$

$$\alpha_t = (1 - \sigma\eta)\alpha_{t-1} + \sigma\eta\alpha \tag{16}$$

The overall algorithm is summarized into Algorithm 1.

Algorithm 1 CFHA tracking algorithm

Input: Initial target position P_{old}, initial target scale $best\_scale$, initial target luminance histogram M.
Output: Target position of each frame $\{P_t\}_2^T$, target scale $best\_scale$.

1: **Repeat**
2: For each scale in S do.
3: Crop a searching region at the last location P_{t-1}.
4: Extract HOG feature map.
5: Calculate the dual space coefficient α with Eq.7.
6: Calculate the response $\hat{f}(z)$ via Eq.8.
7: End
8: Get beat scale according to max $(\hat{f}(z))$.
9: Combine similarity H_i and four maximum values of response $\hat{f}(z)$, $Y = \{(R_i \times H_i) | i \in \{1,2,3,4\}\}$.
10: Detect the target position P_{new}.
11: Analyze the state of object σ with Eq.13.
12: Adjust the learning rate η with Eq.14 and Eq.15.
13: Update the tracking model.
14: **Until** end of video sequence.

IV. EXPERIMENTS

Experiment was performed to the OTB-100 benchmark database in order to evaluate the performance of the proposed CFHA tracker. Firstly, we compare our proposed tracker with eight other trackers (SAMF, DSST, KCF, KCFAMSR, CSK, IVT, CT, DFT). SAMF [8], DSST [3], CSK [1], KCFAMSR [17] and KCF [2] are correlation filter trackers. IVT [23], CT [25] and DFT [24] are tested in the benchmark database [7]. All sequences in OTB-100 are annotated with 11 attributes which cover challenging factors, including fast motion (FM), background clutters (BC), motion blur (MB), deformation (DEF), illumination variation (IV), in plane rotation (IPR), low resolution (LR), occlusion (OCC), out of plane rotation (OPR), out of view (OV), scale variation (SV). We use one-pass evaluation (OPE) to fully evaluate our tracker. The precision scores indicate the percentage of the frames in which the estimated locations are within 20 pixels from the ground-truth positions. The success scores are the area under curve (AUC) of each success plot.

A. Details and parameters

All the experiments are conducted on an Intel Xeon Silver 4112 CPU (2.60GHZ) PC with 16GB memory. Our proposed CFHA tracker runs at about 54 frames per second. The learning rate η is set to 0.015. The gap in scale pool S is set to 0.05. The cell size of HOG is 4x4 and the orientation bin number of HOG is 9.

B. Evaluation of CFHA

The precision scores and the success scores of the proposed tracker CFHA and the other eight trackers are shown in figure 7 for all the 100 image sequences in OTB-100. The results show that CFHA obtains the best performance both in robustness and accuracy. CFHA is based on KCF, but CFHA significantly improves KCF by an average improvement of 20% in the average AUC scores. At the same time, CFHA gains 11% improvement upon KCF in precision. DSST and SAMF focus on solving scale variation as CFHA, but CFHA gets better performance than DSST and SAMF in the computation time. Especially, frame rate is three times higher in CFHA than SAMF. Comparing to non-correlation filter trackers IVT, CT and DFT, the gain of the precision score of CFHA are 81%, 121% and 87% respectively, and as to the success score, 81%, 113% and 74% respectively. Table 1 and Table 2 shows the detail performance of the nine trackers on 11 challenging attributes in OTB-100. The results demonstrate that CFHA obtains robust performance on most attributes, especially on fast motion, motion blur, deformation, low resolution and scale variation. Although SAMF uses more robust features, such as HOG and CN, than CFHA, CFHA gets better performance.

V. Conclusions

In this paper, a new type of coefficient-filter tracking method CFHA is proposed using luminance histogram and an adaptive model, considering the state change of the object.

TABLE I OTB-100 the precision scores of 11 changing factors

	FM	BC	MB	DEF	IV	IPR	LR	OCC	OPR	OV	SV
CT	0.227	0.387	0.197	0.337	0.353	0.364	0.384	0.324	0.362	0.298	0.333
IVT	0.217	0.446	0.204	0.321	0.439	0.433	0.543	0.400	0.429	0.297	0.407
DFT	0.305	0.455	0.272	0.412	0.411	0.434	0.421	0.415	0.440	0.353	0.352
CSK	0.397	0.574	0.355	0.451	0.482	0.514	0.445	0.428	0.489	0.276	0.448
KCF	0.621	0.713	0.601	0.617	0.719	0.701	0.671	0.630	0.677	0.501	0.633
DSST	0.554	0.657	0.555	0.526	0.688	0.684	0.738	0.554	0.622	0.483	0.644
SAMF	0.665	0.689	0.623	0.679	0.710	0.721	0.766	**0.716**	0.728	**0.660**	0.694
KCFAMSR	0.659	0.691	0.589	0.679	0.712	0.705	0.733	0.652	0.691	0.501	0.669
CFHA	**0.753**	**0.724**	**0.701**	**0.714**	**0.748**	**0.767**	**0.785**	0.714	**0.762**	0.648	**0.725**

TABLE II OTB-100 success scores of 11 changing factors

	FM	BC	MB	DEF	IV	IPR	LR	OCC	OPR	OV	SV
CT	0.225	0.297	0.187	0.242	0.269	0.270	0.148	0.264	0.271	0.268	0.246
IVT	0.196	0.325	0.209	0.278	0.333	0.301	0.334	0.288	0.300	0.234	0.286
DFT	0.272	0.369	0.263	0.329	0.334	0.327	0.224	0.332	0.335	0.291	0.262
CSK	0.329	0.410	0.308	0.337	0.368	0.379	0.234	0.331	0.354	0.250	0.318
KCF	0.459	0.498	0.459	0.436	0.479	0.469	0.290	0.443	0.453	0.394	0.394
DSST	0.462	0.491	0.460	0.409	0.529	0.497	0.412	0.426	0.457	0.389	0.473
SAMF	0.512	0.524	0.496	0.501	0.526	0.518	0.425	0.531	0.525	**0.495**	0.484
KCFAMSR	0.501	0.504	0.450	0.486	0.514	0.503	0.470	0.475	0.489	0.393	0.478
CFHA	**0.567**	**0.542**	**0.533**	**0.520**	**0.561**	**0.557**	**0.521**	**0.532**	**0.553**	0.493	**0.536**

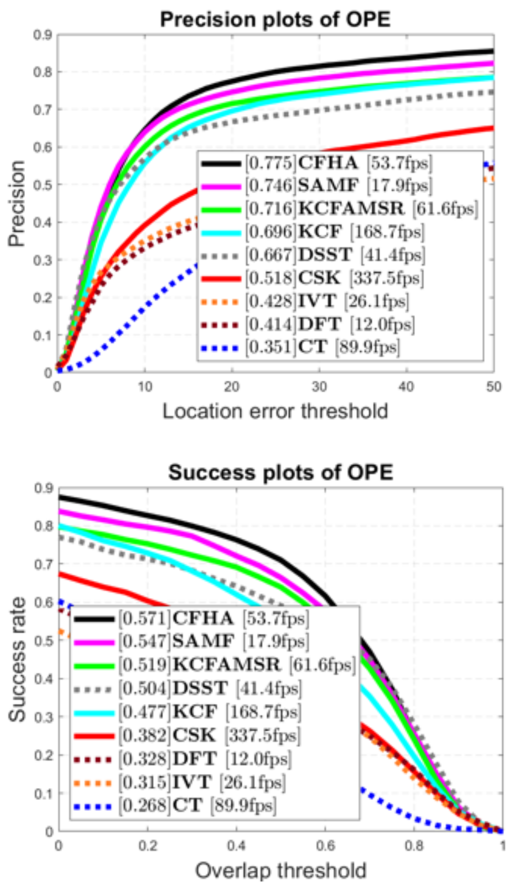

Fig 7. OTB-100 benchmark comparison. The precision plot (upper) and the success plot (lower).

CFHA is realized by adding three types of new methodologies to KCF to consider the state change; one is a skipped scale pool method to deal with scale variation with less calculation time, one is introducing the similarity of luminance histogram to estimate the location of the object, and the other is adjusting the learning rate on the basis of the state of the object which is recognized by the combination of the change of the maximum response and the similarity of luminance histogram. Especially, the second factor is new, compared with KCFAMSR which was proposed by authors before. From the experimental results for OTB-100, the proposed method is shown to obtain more robust and effective performance than conventional method. Improvement of our method using color attribute is for further research.

REFERENCES

[1] João F. Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. Exploiting the Circulant Structure of Tracking-by-detection with Kernels. In ECCV LNCS, vol.7575, pp.702-715,2012.

[2] João F. Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. High-speed tracking with kernelized correlation filters. TPAMI, vol. 37, no. 20, pp. 583-596, 2014.

[3] M. Danelljan, G. Häger, F. Khan, M. Felsberg. Accurate Scale Estimation for Robust Visual Tracking. In BMVC, 2014.

[4] Z. Hong, C. Wang, X. Mei, D. Prokhorov, and D. Tao. Tracking using multilevel quantizations. In ECCV LNCS, vol.8694, pp.155-171,2014.

[5] J. Kwon, J. Roh, K.-M. Lee, and L. Van Gool. Robust visual tracking with double bounding box model. In ECCV LNCS, vol.8689, pp.377-392,2014.

[6] H.Nam, S. Hong, and B. Han. Online graph-based tracking. In ECCV LNCS, vol.8693, pp.112-126,2014.

[7] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Online object tracking: A benchmark. In CVPR, pp.2411-2418,2013.

[8] Li Y, Zhu J. A scale adaptive kernel correlation.filter tracker with feature integration. In ECCV LNCS, vol.8926, pp.254-265,2014.

[9] Zhong, W., Lu, H., Yang, M.H.: Robust object tracking via sparsity-based collaborative model. In: CVPR. pp. 1838–1845, 2012.

[10] E.Garcia. Cosine Similarity Tutorial. Published: 2015.

[11] H. Kiani Galoogahi, T. Sim, and S. Lucey. Correlation filters with limited boundaries. In *CVPR*, pages 4630–4638, 2015.

[12] B. Babenko, M.-H. Yang, and S. Belongie. Robust object tracking with online multiple instance learning. *TPAMI*, 33(8), 2011.

[13] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah. Visual tracking: An experimental survey. *TPAMI*, 36(7):1442–1468, 2014.

[14] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Yui M. Lui. Visual object tracking using adaptive correlation filters. In CVPR, 2010.

[15] Wu, Y., Lim, J., Yang, M.H.: Object tracking benchmark. TPAMI 37(9), 1834– 1848 (2015)

[16] Danelljan, M., Khan, F.S., Felsberg, M., van de Weijer., J.: Adaptive color attributes for real-time visual tracking. In: CVPR (2014)

[17] Zhaoqian Tang, Kaoru Arakawa. Kernel Correlation Filter via Adaptive Model. In ISPACS, 2018.

[18] V.N.Boddeti,T.Kanade,andB.V.K.V.Kumar,"Correlationfilters for object alignment." in *CVPR*, 2013.

[19] F. S. Khan, J. van de Weijer, and M. Vanrell. Modulat- ing shape features by color attention for object recognition. *IJCV*, 98(1):49–64, 2012.

[20] Khan, F.S., van de Weijer, J., Vanrell, M.: Modulating shape features by color attention for object recognition. IJCV 98(1), 4964 (201)

[21] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-learning-detection. *IEEE transactions on pattern analysis and ma- chine intelligence*, 34(7):1409–1422, 2012.

[22] J. Kwon and K. M. Lee. Visual tracking decomposition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1269–1276. IEEE, 2010.

[23] D. Ross, J. Lim, R.-S. Lin, and M.-H. Yang. Incremental Learning for Robust Visual Tracking. IJCV, 77(1):125–141, 2008.

[24] L. Sevilla-Lara and E. Learned-Miller. Distribution Fields for Tracking. In CVPR, 2012.
K. Zhang, L. Zhang, and M.-H. Yang. Real-time Compressive Tracking. In ECCV, 2012.