Pattern Recognition 69 (2017) 82–93

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Robust visual tracking via co-trained Kernelized correlation filters

Le Zhang [a], Ponnuthurai Nagaratnam Suganthan [b,*]

[a] Advanced Digital Sciences Center, 138632, Singapore
[b] School of Electric and Electronic Engineering, NanYang Technological University, 639798, Singapore

ARTICLE INFO

Article history:
Received 12 October 2016
Revised 14 March 2017
Accepted 4 April 2017
Available online 14 April 2017

Keywords:
Visual tracking
High speed
Kernelized correlation filter
Ensemble tracking
KCF tracker
COKCF tracker
Correlation filter

ABSTRACT

Recent advances in visual tracking have witnessed the importance of discriminative classifiers tasked with distinguishing the target from the background. However, a single classifier may fail to cope with complex surrounding environment and large appearance variations of the target. Motivated by multi-view learning, we equip a basic framework to train a pool of discriminative classifiers jointly in a closed-form fashion in this paper. It poses an extra regularization term in ridge regression which interacts with other base models in the ensemble. Through a simple realization of this approach, we show co-trained kernelized correlation filters (COKCF) which consist of two KCF trackers, are able to outperform the KCF tracker by a larger margin and perform favorably against other state-of-the-art trackers on 63 benchmark video sequences.

1. Introduction

As a fundamental task in computer vision, visual tracking has received a fast-growing interest due to applications such as video surveillance, human computer interaction, medical imaging and so on. As demonstrated in Fig. 1, despite significant progress made recently, it remains challenging due to factors such as lighting and background variations, occlusions, motion blur and so on.

Although visual tracking can be formulated in different settings according to different applications, most researchers focus on the *one – pass model – free single object tracking* setting, which is the most fundamental case. More specifically, without any appearance model, it only provides the bounding box of one single object in the first frame. Given this single (labeled) instance, the goal is to predict the location of the object in an online manner.

Modern trackers consist of 2 sub-models. One is motion model and the other is the appearance model. The motion model aims at estimating the posteriori probability of the target states. For the motion model, particle filter, which can be formulated as a sequential Monte Carlo importance sampling method for estimating the latent state variables of a dynamical system based on a sequence of observations, has long been regarded as the dominant approach. Moreover, recent work in [70] shows that there does not exist significant differences among various motion models such as particle filter, sliding window and radius sliding window. Appearance model plays the central role in modern tracking systems. Based on appearance model, most trackers can be divided into generative tracker [1,8,14,34,45,48,48,55,57] or discriminative tracker [4,5,24,25,74,77,86] or hybrid of them [91]. Generative trackers formulate the tracking problem as searching for the regions most similar to the target model. The rationale behind generative models for tracking is based on the reasonable assumption that the targets being tracked can be described by some generative processes. These methods are either based on template matching [1,14,34,45,48] or subspace learning [8,55] or learning more powerful descriptors for the image patch [57]. The appearance model also needs to be updated dynamically to adapt to the target appearance variations. A template update method was proposed in [48] which could reduce the drifting problem by aligning with the first template to reduce drifts. Incremental subspace learning was proposed in [55]. In order to cover a wide range of pose and illumination variations, appearance model in [34] is decomposed into several basic appearance models. Recently, sparse representation has been proposed in visual tracking [7,38,39,50,51]. It works by solving the $l1$ norm based optimization and the candidates with the lowest reconstruction error is selected as the tracking result. Sparse prototype was proposed in [66] which exploits classic principal component analysis (PCA) algorithm with recent sparse representation schemes for learning effective appearance models.

Discriminative trackers solve tracking problem by an online classification problem which distinguishes the target from the

* Corresponding author.
E-mail addresses: zhang.le@adsc.com.sg (L. Zhang), epnsugan@ntu.edu.sg (P.N. Suganthan).

http://dx.doi.org/10.1016/j.patcog.2017.04.004
0031-3203/

Fig. 1. Challenging scenarios in visual tracking. The first row shows that the object being tracked may be heavily occluded. In the second row, there exists motion blur in the video sequence. In the last row, the illumination may change significantly.

background. Discriminative trackers utilize the information from both the object and the background. Classical example includes ensemble tracking [4], where each base tracker is a least-square classifier which works directly on the image pixels. Online boosting was proposed in [24] to online update discriminative features. Extensions of online boosting to semi-supervised versions were addressed in [25] to prevent drift. Multiple instance learning [5,84], which puts all ambiguous positive and negative samples into bags to learn a discriminative model for tracking, aims at alleviating the "butterfly effect" caused by inaccurate update of the classifier. In [86], online discriminative feature selection was proposed for visual tracking. It optimized the objective function in the steepest ascent direction with respect to the positive samples while in the steepest descent direction with respect to the negative ones. The concept of "superpixel" was proposed in [79] which facilitates the tracker to distinguish between the target and background. Partial least square was employed in [74] to learn a set of appearance models for adaptive discriminative object representation. Self-paced learning and online solvers for SVMs were proposed in [62] for visual tracking. In [90], a simple convolutional neural network based tracker was proposed and updated with back-propagation to extracted discriminant features to distinguish the object from background. Deep learning and transfer learning based tracking was proposed in [69,72] where the author pre-train a deep neural network with a huge amount of labeled images and finetune in the process of tracking. Similar idea can be found in [75] where visual prior from generic real-world images can be learned and transferred for representing objects in a scene. Deep learning without pre-training can be found in [89].

According to the comprehensive results on recent proposed surveys [76], discriminative methods show their superiority over generative models. However, almost all aforementioned discriminative methods focus on characterizing the object of interest-the positive samples for the classifier. On the other hand, image patches whose overlap score is below a threshold are randomly sampled as the negative samples. However, a core tenet of discriminative trackers is to give as much importance, or more if necessary, to the relevant environment since it gives rich information about the background. Hence, the work in [27] and its kernelized successor [28] argue that undersampling the negatives is the main factor inhibiting tracking performance. As a result, they propose an analytic model for datasets of thousands of translated patches.

Though many efforts have been devoted in recent years and significant improvements can be found in the literature, visual tracking remains challenging for real-life application. On one hand, single classifier may be not powerful enough to handle complex background and significant variations of target appearance, and ensemble methods seem to be a promising approach to significant improve the tracking performance [70]. Ensemble learning can be either integrated in the learning phase [6] or as a post-processor to integrate multiple base trackers [73]. On the other hand, modern trackers suffer from relatively large computational complexity, which is a major hurdle that hinders the ensemble based visual tracking. In this work, in order to strike a good trade-off between the robustness and speed in visual tracking, we embark on KCF tracker [28], which is very efficient in training and updating the binary classifier. We jointly train an ensemble of base trackers in a closed-form fashion. We show that the realization of the proposed methods are able to outperform the KCF tracker by a larger margin and perform favorably against other state-of-the-art trackers on 63 benchmark video sequences.

Our proposed approach is also aligned with multi-view learning [3,10,61,78]. In the proposed methods, each base tracker is working on its own view and trying to correct the other one by continuous interaction through the tracking process to handle large appearance variations and avoid drifting. Proliferation of cameras, availability of cheap storage and rapid developments in computer hardware spurred the rise in multimodal data in which multi-view learning plays an inevitable role. It holds great promise for a variety of applications ranging from the Internet of Things to surveillance and assisted living. Multi-view based tracking [49,82] is not new under this umbrella. However, these methods inherit high computational complexity which severely compromises the speed of the system. Furthermore, these methods suffer from poor perfor-

mance under the challenging environment. The proposed methods strike a good trade-off between the robustness and speed in visual tracking by exploiting the circulant structure of the data sample.

To summarize, we make the following contributions in this paper:

- Motivated by multi-view learning, we propose a co-trained ensemble tracker where each base tracker is working on its own view and trying to correct the other one by continuous interaction through the tracking process.
- By exploiting the circulant structure of the data sample to alleviate the sampling ambiguity, the proposed co-trained ensemble methods have low computational cost and can be used in real-time application.
- We present two realizations of the proposed method which operates on handcrafted features and features extracted from convolutional neural network, respectively. Both of them are able to outperform the baseline tracker by a large margin and perform favorably against other state-of-the-art trackers on 63 benchmark video sequences.

In the reminder of the paper, we firstly give a brief introduction about the related methods in Section 2. Overview and details about the proposed method can be found in Section 3. Experimental results and discussions can be found in 4. In the last section, we conclude this work.

2. Related work

In this section, we discuss tracking methods closely related to the present work. A comprehensive review of visual tracking is outside the scope of this paper. We refer the interested readers to recent survey [60] for detailed information.

2.1. Correlation filters for tracking

Correlation filters belong to the well-known tracking-by-detection framework. Since the pioneering work in [9], correlation filters based tracking has attracted considerable attention due to its high computational efficiency via fast Fourier transform. They have been proven to be competitive with far more complicated tracking methods [27,28] while operating at real-time speed. Recent work shows that KCF tracker, a kernelized version of correlation filters, outperforms other trackers in this family [28]. In the Fourier domain of the training data, which is presented in a circulant matrix to avoid hard negative mining, is made diagonal [26]. Except for DFT and inverse DFT, which bounds the cost at a nearly-linear $O(n\log n)$ (n is the dimensionality of the features), the cost of the element-wise operation in the Fourier domain is $O(n)$, which is much smaller than the conventional method ($O(n^3)$) [28].

In [28], without hard negative mining, the authors assume that the cyclically shifted version of the base sample is able to approximate the dense samples over the base sample. For notational simplicity, we choose one dimensional signal to demonstrate the idea of KCF. The results generalize to two dimensional multi-channel signals [28]. Given an $1 \times n$ vector $x = [x_1, x_2, \ldots, x_n]$ representing a data sample, a cyclic shift of x can be represented as $Px = [x_n, x_1, x_2, \ldots, x_{n-1}]$. The data matrix $X = C(x)$ turns out to be a matrix which is a concatenation of all the cyclic shift visual samples, $\{P^u x | u = 0 \ldots n-1\}$. Thus, X is called circulant matrix and can be made diagonal by DFT matrix:

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

where F is known as the DFT matrix which transforms the data into Fourier domain $\hat{x}$ and F^H is the Hermitian transpose of F. The learning problem is formulated by ridge regression [54], which aims at minimizing the squared error over the output of sample x_i and its desired target y_i with a function $f(z) = w^T z$.

$$\min_w \sum_i^n (f(x_i) - y_i)^2 + \lambda \|w\|^2 \tag{2}$$

where λ is the regularization parameter. x_i is the ith row of the data matrix X. The regression labels[1] of the training sample are generated in a Gaussian function, which equals to 1 for the centered target and smoothly decays to 0 for the cyclic shifts.

It is easy to obtain the solution for the ridge regression as $W = (X^T X + \lambda I)^{-1} X^T y$. Combining with Eq. (1), we have the solution $\hat{w} = \frac{\hat{x}^* \odot \hat{y}}{\hat{x}^* \odot \hat{x} + \lambda}$, where * denotes the complex-conjugate. The solution can also be generated for the non-linear case with kernel trick,

$$f(z) = w^T z = \sum_{i=1}^n \alpha_i \kappa(z, x_i), \tag{3}$$

Most commonly used kernel functions lead to a circulant matrix [28] and the dual space coefficients α can be learnt as below:

$$\hat{\alpha} = \frac{\hat{y}}{\hat{k}^{xx} + \lambda} \tag{4}$$

where k^{xx} is called as the kernel correlation in [28] and can be computed as

$$k_i^{xx} = \kappa(x, P^{i-1}x). \tag{5}$$

Since the work in [28], there exist numerous extensions for KCF trackers. In [40], part based KCF was proposed to address occlusion. Reliable Patch Trackers (RPT) can be found in [37]. Specifically, the authors in [37] presented a tracking reliability metric to measure how reliably a patch can be tracked, where a probability model is proposed to estimate the distribution of reliable patches under a sequential Monte Carlo framework. Correlation filter was proposed in [85] to model the spatio-temporal context information. In [16], a scale pyramid representation by learning discriminative correlation filters was proposed. Multiple kernel for correlation filter based tracking can be found in [63]. The so-called MKCF tracker fully takes advantage of the invariance-discriminative power spectra of various features to further improve the performance. In [47], an online random fern classifier was employed as re-detection component for long-term tracking. A biology-inspired framework where short-term processing and long-term processing are cooperated with each other [30] in correlation filter based trackers. In [46], the authors hierarchically infer the maximum response of ensemble of correlation filters trained on the features extracted from different layers of convolutional neural networks (ConvNets). Given the set of correlation response maps f_l, targets location can be hierarchically infered. More specifically, let $(\overline{m}, \overline{n}) = argmax_{m,n}\ f_l(m,n)$ indicate the location of the maximum values on the lth layer, the optimal location of the target in the $(l-1)$th layer can be inferred as:

$$\begin{aligned} &\underset{m,n}{argmax}\ f_{l-1}(m,n) + \gamma f_l(m,n), \\ &s.t.\ |m - \overline{m}| + |n - \overline{n}| \le r \end{aligned} \tag{6}$$

The problem defined above indicates that only the $r \times r$ neighburing region of $(\overline{m}, \overline{n})$ is searched in the $(l-1)$th correlation response map. The target location is finally estimated by hierarchically maximizing Eq. (6).

[1] We follow the notions in [28]. Regression with class label can be regarded as classification. Hence, these two terms are used interchangeably in this paper.

2.2. Ensemble tracking

Ensemble methods [19,53] have lead to a great success in machine learning tasks. Theoretical explanations can be found in strength-correlation [11] bias-variance decomposition [87] and so on. Motivated by this, ensemble methods also saw heavy use in visual tracking. The pioneering work in [4] formulated each base tracker as a least-square classifier which worked directly on the image pixels. Other classical examples include boosting [24] and semi-supervised boosting [25]. Randomized ensemble tracker was proposed in [6] to model the time-varying appearance of an object for visual tracking. Instead of updating classifier ensembles with the commonly used methods in tracking-by-detection pipeline, the weight vector to combine weak classifiers is posed as a random variable. The posterior distribution for the weight vector is estimated in a Bayesian manner. In [35] the authors proposed the visual tracker sampler that can work robustly in challenging scenarios. The proposed tracking algorithm accurately tracks a target by searching for appropriate trackers in each frame. A novel tracking algorithm that combines complementary tracking modules with a new object representation model to balance between stability and adaptivity was proposed in [43]. To reduce the update error of online tracking, three complementary modules (a stable module, a soft stable module, and an adaptive module) were proposed. The authors fused them by using a biased multiplicative criterion. In [83], the authors propose a multi-expert restoration scheme to address the model drift problem in online tracking. The so-called MEEM tracker consists of its historical snapshots. The best expert is selected to restore the current tracker when needed based on a minimum entropy criterion. In [73], the authors studied the visual tracking problem in which the unknown data to be estimated is in the form of a sequence of bounding boxes representing the trajectory of the target object being tracked. As a result, they proposed a factorial hidden Markov model (FHMM) for ensemble-based tracking by learning jointly the unknown trajectory of the target and the reliability of each tracker in the ensemble. In [71], the role of each key component of the visual tracking system is analyzed and ensemble post-processing is found to boost the tracking performance significantly.

3. Proposed method

3.1. Motivation

Single classifier may be not powerful enough to handle complex background and significant variations of target appearance. A straightforward method to tackle this is the aforementioned ensemble tracking. However, ensemble tracking suffers from high computational time which makes them unsuitable for real-time applications. Thus, in this work, we strike a good trade-off between the robustness and speed in visual tracking. As a result, we embark on KCF tracker, which is very efficient in training and updating classifiers, and jointly train an ensemble of base trackers in a closed-form fashion. Extensive experiments on recently proposed benchmark dataset indicate that the proposed algorithm performs favorably against state-of-the-art methods.

3.2. Jointly training of ensemble correlation filters

The kernel matrix in Eq. (4) plays a central role in KCF tracker. Actually different kernels may correspond to different notions of similarity, and using a specific kernel may lead to a risk of bias and result in drift in tracking. Multiple kernel learning (MKL) tackles this problem by using a combination of different kernels and is reported to consistently improve the performance of kernel methods in machine learning [23] and computer vision [44,80,81]. However, though MKL or other ensemble methods can improve the stability of the tracker, they usually significantly reduce the tracking speed due to high computational complexity in the optimization process. (An example can be found in [63], where a two kernel correlation filters is proposed in the MKL framework. With the same feature, our method achieves almost the same performance with 10 times faster in fps.) Hence, in this study, we propose a simple method which jointly trains a pool of classifiers, and most importantly, inherits the efficiency of KCF tracker in the Fourier domain.

Suppose that we have M kernels, then the proposed method aims at solving the following problem:

$$\begin{aligned} T(\alpha) = \min_{\alpha} & \sum_{i=1}^{M} (\|y - K_i\alpha_i\|^2 + \lambda \alpha_i^T K_i \alpha_i) \\ & + \beta \sum_{i,j=1}^{M} \left\| K_i\alpha_i - K_j\alpha_j \right\|^2 \end{aligned} \tag{7}$$

The first term in Eq. (7) forces each individual model to have minimal squared error with respect to the desired output y. The second term $(\alpha_i^T K_i \alpha_i)$ denotes the Tikhonov regularization in a Reproducing Kernel Hilbert Space. The third term regularizes each model in a pair-wise fashion which weights the influence of pairwise disagreements. It allows high performance to be achieved since it becomes unlikely for compatible classifiers trained on independent kernels/features to agree on an incorrect label. When $\beta = 0$, the proposed method is equivalent to M independent kernel ridge regression problems. The proposed method can also be regarded as a purely supervised version of the co-regularized multiview learning [59].

It is worth noting that the proposed method can have many realizations. If we generate each kernel by using different kernel functions such as Gaussian, polynomial and linear kernel or with different kernel parameters, the resulting tracker can be regarded as an MKL tracker [44,80]. One can also generate each kernel with different features such as HOG [15], SIFT [42], Haar-like [65], or LBP [2]. Another approach could be a part-based tracker if each kernel is generated with only fixed part of the object patch.

In the present work, we simply generate two Gaussian kernels K_1, K_2 with the same parameters and with different features, respectively. This approach is named as co-trained KCF (COKCF) tracker. We have conducted two sets of experiments, one is based on handcrafted features and the other is with features extracted from ConvNets [58].

It is straight-forward to obtain the solution of Eq. (7) by setting the derivation of T with respect to α_i to be zero:

Let

$$G_i = (2\beta + 1)K_iK_i + \lambda K_i, \tag{8}$$

we can obtain the solution for Eq. (7)

$$\begin{aligned} \alpha_1 =& (G_1 - 4\beta^2 K_1K_2G_2^{-1}K_2K_1)^{-1} \\ & * (K_1y + 2\beta K_1K_2G_2^{-1}K_2y), \\ \alpha_2 =& (G_2 - 4\beta^2 K_2K_1G_1^{-1}K_1K_2)^{-1} \\ & * (K_2y + 2\beta K_2K_1G_1^{-1}K_1y), \end{aligned} \tag{9}$$

Let $\nu = 2\beta + 1$ and notice

$$G_i^{-1} = (\nu K_i + \lambda I)^{-1} * K_i^{-1}, \tag{10}$$

Substituting Eq. (8) into Eq. (9), we have

$$\begin{aligned} \alpha_1 =& ((\nu K_1 + \lambda I) - 4\beta^2 K_2(\nu K_2 + \lambda I)^{-1}K_1)^{-1} \\ & * (y + 2\beta K_2(\nu K_2 + \lambda I)^{-1}y), \end{aligned} \tag{11}$$

By applying Eq. (1), we have

$$\hat{\alpha_1} = diag(\nu k_1 + \lambda - 4\beta^2 \frac{k_2 \odot k_1}{\nu k_2 + \lambda})^{-1} * diag(1 + \frac{2\beta k_2}{\nu k_2 + \lambda}) * \hat{y}, \quad (12)$$

or better yet,

$$\hat{\alpha_1} = \left(\frac{(1+4\beta)k_2 + \lambda}{(1+4\beta)k_1 \odot k_2 + \lambda(2\beta+1)(k_1+k_2) + \lambda^2} \right) \odot \hat{y} = \frac{\hat{y}}{k_1 + \lambda + \frac{2\beta\lambda(k_1-k_2)}{(1+4\beta)k_2+\lambda}} \quad (13)$$

where k_1, k_2 can be computed in the same way as Eq. (5). The above matrix multiplication/division can be efficiently computed in an element-wise manner since they are all diagonal matrices. Similar solution can be derived for α_2.

Similar to KCF, the pipeline for the COKCF tracker is intentionally simple and does not include any heuristic for failure detection or motion modelling. In the first frame, we generate two kernels K_1, K_2 with the same Gaussian kernel and with HOG and color attributes, respectively. For each new frame, detection is performed at the previous position.

As we mentioned above, the first term in Eq. (7) forces each individual model to have the minimal squared error with respect to the desired output y. The second term $(\alpha_i^T K_i \alpha_i)$ denotes the commonly used Tikhonov regularization in a Reproducing Kernel Hilbert Space. The third term regularizes each model in a pair-wise fashion which weights the influence of pairwise disagreements. It allows high performance to be achieved since it becomes unlikely for compatible classifiers trained on independent kernels/features to agree on an incorrect label. For example, if one model becomes less accurate for a new arrival datum, the last constraint will force the output of this model to be as accurate as other models in the ensemble. At first glance, the proposed method looks contradictory to the existing Negative Correlation Learning (NCL) approach [41]. In NCL, each base model is forced to be different to each other in order to have decorrelated errors. However, NCL was originally proposed to regularize neural network ensembles. Without negative correlation constraints, all base neural network model may converge to the same local minimum solution. However, the proposed method can be regarded as the positive correlation learning. It is effective because here each kernel is constructed with either different source of features or different kernel functions. It is unlikely for those inherent diversity of kernels to have the same solutions for a given problem. The positive correlation constraint can force the output of one "poor" model to be as accurate as the other "good" models in the ensemble. We also have conducted experiments with NCL and found the results to be much worse than the proposed methods here, as we expected.

The proposed method shares the same merit with ensemble learning [20]. In machine learning and statistical learning, researchers adopt ensemble learning which employs multiple models to obtain better performance than that could be achieved by any single model amongst them [88]. From the bias-variance point of view, prediction error of a learning model is equal to the sum of bias and variance [12,21]. Ensemble method can significantly reduce the overall variance without increasing the bias significantly [12]. Hence, the overall error will be reduced. From statistical point of view, learning can be posed as searching a space $\mathcal{H}$ of hypotheses to identify the best hypothesis in the space for data fitting. However, when the amount of training data available is limited compared to the size of the hypothesis space, such as visual tracking where only the bounding box of the first frame is provided, statistical problem may arise. Without sufficient data, one learning algorithm may end up with many different hypotheses in $\mathcal{H}$ which all fit the training data well. In this case, the algorithm can manipulate their votes and reduce the risk of choosing the "wrong" classifier by generating an ensemble. The other reason is representational. It is possible that the true function $\mathbf{g}$ may not be well represented by any of the hypotheses in $\mathcal{H}$. For example, function $\mathbf{g_1}$ working on features extracted from higher layers of ConvNets capture semantic concepts on object categories while function $\mathbf{g_2}$ working on features extracted from lower layers of ConvNets encode more discriminative features to capture intra class variations. Either function $\mathbf{g_1}$ or $\mathbf{g_2}$ may fail to cope with simultaneous challenging cases such as illumination variations and background clutter. However, it may still be possible to expand the space of representable functions, thereby enhancing the representation ability, by combing hypotheses of $\mathbf{g_1}$ and $\mathbf{g_2}$. Conventional ensemble tracking methods inherit high computational complexity and thus can only be operated in an extremely low frame rate. Furthermore, these methods suffer from poor performance under challenging environments. The proposed method here strike a good trade-off between the robustness and speed in visual tracking. As we demonstrated in Section 4.5, each base tracker is both accurate and fast. The co-trained methods allow them to correct the other one by continuous interaction through the tracking process.

The proposed method is also well-aligned with the popular multi-view learning [3,61,78]. Proliferation of cameras, availability of cheap storage and rapid developments in computer hardware has spurred the rise in multi-view data. In these cases, real-time operation requires low-complexity processing in a multi-view framework. There is therefore a critical need to develop efficient machine learning algorithm for computer vision tasks. When it comes to multi-view tracking, there exists a large performance gap between current multi-view trackers and other state-of-the-art trackers (See Section 4.4 for details). The proposed method fills in the gap by letting each base tracker work on its own view in a consensus manner.

3.3. Model update

We adopt a similar approach as KCF to update the co-trained models for visual tracking. More specifically, the solution of the correlation filters for each base model are updated with moving average:

$$\alpha_t = (1-\eta)\alpha_{t-1} + \eta\alpha_t; x_t = (1-\eta)x_{t-1} + \eta x_t; \quad (14)$$

where $x \in [x_1, x_2]$, $\alpha \in [\alpha_1, \alpha_2]$, t is the frame index and η is the learning rate.

3.4. Overall tracking pipeline

The pipeline for the tracker is intentionally simple and we do not include any heuristics for failure detection, motion modeling or model averaging. We **train** a model with two feature sets from two different views (x_1 x_2) of the image patch which includes some context at the initial position of the target via Eq. (13). During the tracking process, we **detect** via Eq. (13) over the patch at the previous position for each new video frame, and the target position is updated to the result of the base tracker which yields the maximum value. In order to provide the tracker with some memory, we **update** a new model via Eq. (14) at the new position when it is necessary by linearly interpolating the obtained values of α_1, x_1, α_2 and x_2 with the ones from the previous frame, respectively. The bulk of the functionality of the proposed method is presented as Matlab codes in Algorithm 1.

Algorithm 1 Matlab pseduo-codes of the proposed method.

Details of the proposed method.

Input:

- x_1: training features from the first view.
- x_2: training features from the second view.
- z_1: testing features from the first view.
- z_2: testing features from the second view.
- y : regression target.
- λ, β: regularization parameter in Eq. (7).
- σ: kernel parameters (We provide Gaussian kernel here for an example.)

Output:

- responses: detection score for each location.

function $[\alpha_1, \alpha_2]$ = **train**($x_1, x_2, y, \sigma, \lambda, \beta$)
k_1 = kernel correlation(x_1, x_1, σ);
k_2 = kernel correlation(x_2, x_2, σ);
$\hat{y}$=**fft2**(y);
$\hat{\alpha_1} = \frac{\hat{y}}{k_1+\lambda+\frac{2\beta\lambda(k_1-k_2)}{(1+4\beta)k_2+\lambda}}$
$\hat{\alpha_2} = \frac{\hat{y}}{k_2+\lambda+\frac{2\beta\lambda(k_2-k_1)}{(1+4\beta)k_1+\lambda}}$
end

function responses = **detect**($\alpha_1, \alpha_2, x_1, x_2, z_1, z_2\ \sigma$)
k_1 = kernel_correlation(z_1, x_1, σ);
k_2 = kernel_correlation(z_2, x_2, σ);
responses_1 = **real**(**ifft2**(α_1 .* **fft2**(k_1)));
responses_2 = **real**(**ifft2**(α_2 .* **fft2**(k_2)));
if **max**(responses_1(:))≥ **max**(responses_2(:))
responses=responses_1;
else
responses=responses_2;
end
end

function k = **kernel_correlation**($x1$, $x2$, σ)
c = **ifft2**(**sum**(**conj**(**fft2**($x1$)) .*fft2($x2$), 3));
$d = x1(:) * x1(:) + x2(:) * x2(:) - 2 * c$;
$k = \mathbf{exp}(-1/\sigma^2 * \mathbf{abs}(d)/\mathbf{numel}(d))$;
end

Tracking Pipeline

If Current frame is the first frame

- Get multi-view features x_1, x_2 with the input frame and the given bounding box.
- Train the model with **train** module and save template x_1, x_2

else

- Get multi-view features z_1, z_2 with the input frame from previous results.
- Get the tracking results with **detect** module.
- Update the model with **update** module.

End If

4. Experiment

4.1. Benchmark evaluations

Firstly, we evaluate our tracker by using a recent benchmark which has 51 video sequences [76]. This benchmark collects many video sequences used in previous works to avoid the danger of overfitting to a small subset. For the performance criteria, we did not adopt the commonly used average location error or other measures that are averaged over frames. Because once the tracker lose the target, they will impose an arbitrary penalty to the tracker and lead to an unfair measurement. In this work, we adopt the measurement in the benchmark: the precision curve. The precision curve shows the percentage of frames whose estimated location is within the given threshold distance of the ground truth. As done in [76], we set the threshold for precision curve to be 20 pixels.

4.2. CoKCF_HC (CoKCF with handcrafted features)

In this section, we implement the CoKCF tracker based on two handcrafted features: HOG [15] and color features [64]. We name the proposed method with these two features as CoKCF_HC. For the HOG feature, we adopt the same configurations used in [28] which sets the cell size to be 4 and number of orientations to be 9. We use the same color features in [64]. For this part, the two base trackers operate in a competitive manner, which means the one with larger peak-to-sidelobe ratio (PSR) will give the tracking result as the position of detection with the maximum value based on Eq. (3). Finally, we update the tracker by linearly interpolating the resulting α and x with the ones from the previous frame. The learning rate is set as $0.002*PSR$. The regularization parameter λ and β are set to be e^{-4} and 0.05 respectively and the width parameter for the Gaussian kernel is 0.5 (the same as KCF) throughout all the experiments. As mentioned earlier, the overall complexity of each base KCF tracker is dominated by the DFT, which is $O(n \log n)$. Thus, the overall complexity of the whole ensemble tracker is around $O(Mn \log n)$, where M is the number of the base trackers. In our regular PC with Intel-i7-3770 CPU (3.4 GHz) and 32GB RAM, KCF tracker operates at an average fps of around 260 while CoKCF runs at around 130 fps (including feature extraction).[2]

We firstly summarize the results over 51 video sequences in Table 1. We can see that the proposed method demonstrates its superiority over the KCF tracker in overall performance and most of the challenging scenarios.

4.3. CoKCF_CNN (CoKCF with features extracted from ConvNets)

Handcrafted features, such as HOG [15], SIFT [42] and LBP [2], saw heavy use in the last decade, but then fell out of fashion with the rise of deep ConvNets trained on large-scale datasets [33]. In [33], the authors rekindled interest in ConvNets by demonstrating significantly better image classification accuracy on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [56].

In [52], the authors show generic descriptors extracted from the ConvNets are very powerful for a wide range of recognition tasks. Motivated by this work, Chao Ma et al. propose to train correlation filters with the features extracted from different layers of ConvNets. They hierarchically infer the maximum response of each layer to locate targets [46]. We name this method as KCF_CNN. In this section, we follow the same pipeline as done in [46]. More specifically, we use the VGG-19 model [58] to encode target appearance. We train two sets of correlations filters with features

[2] Source code will be available at the authors' homepage.

Table 1
Performance (Mean precision (20 px)) comparisons of CoKCF_HC with KCF.

Methods	Overall	IV	SV	Occ	Def	MB	FM	IR	OR	OV	BC	LR	Frame Rate
CoKCF_HC	**77.5**	72.0	**76.2**	**68.5**	**77.7**	**79.8**	**66.0**	57.9	**73.2**	62.9	70.2	**38.8**	130
KCF	74.3	**73.3**	73.3	67.9	75.4	74.8	60.9	**61.0**	72.9	**65.0**	**75.3**	38.1	**260**

Some keys are: IV=Illumination variation, SV=Scale variation, Occ=Occlusion, Def=Deformation, MB=Motion Blur, FM=Fast Motion, IR=In-Plane Rotation, OR=Out-of-Plane Rotation, OV=Out of View, LR=Low Resolution.

Table 2
Performance (Mean precision (20 px)) comparisons of CoKCF_CNN with KCF [28] and the KCF_CNN [46].

Methods	Overall	IV	SV	Occ	Def	MB	FM	IR	OR	OV	BC	LR	Frame Rate
KCF	74.3	73.3	73.3	67.9	75.4	74.8	60.9	61.0	72.9	65.0	75.3	38.1	**260**
KCF_CNN	**89.2**	**84.6**	**87.0**	88.0	**87.9**	**88.3**	**84.8**	**79.2**	**87.0**	69.5	88.5	89.7	6
CoKCF_CNN	88.9	83.4	86.2	**88.9**	87.1	84.7	81.3	76.4	86.4	**70.2**	**89.2**	**93.8**	9

Some keys are: IV=Illumination variation, SV=Scale variation, Occ=Occlusion, Def=Deformation, MB=Motion Blur, FM=Fast Motion, IR=In-Plane Rotation, OR=Out-of-Plane Rotation, OV=Out of View, LR=Low Resolution.

extracted from $conv4-4$ and $conv5-4$ layers (which correspond to the 28th and 37th layers of the VGG-19 model [58]). We resize those two output maps with bilinear interpolation to be the same as the window size (e.g., 1.8 times of the target size). The regularization parameters λ and β are set to be 10^{-5} and 10^{-3}, respectively. We find that the tracking performance is not sensitive to these settings. Linear kernel is used. Moreover, the learning parameter is set to be 0.01 for these two models. We name the proposed method on features extracted from ConvNets as CoKCF_CNN to differentiate from the previous section. The current implementation of the CoKCF_CNN tracker runs at around 9 fps at our K40 GPU, while the work in [46] operates at about 6 fps on the same machine.

In the same way, we summarize the detailed results of the proposed method in Table 2. We also summarize the detailed result of KCF_CNN in [46]. Table 2 clearly demonstrates the superiority of tracking with ConvNet features. The proposed method achieves competitive performance compared with [46]. Note that the proposed method is both conceptually simpler and computational cheaper (around 1.5 times faster) than the method in [46]. In [46], the location of target is hierarchically inferred from the output of 3 sets of different correlation filters. However, in CoKCF_CNN, the response can be easily obtained by only 2 sets of correlation filters.we give the overall precision plot under one-pass evaluation [76] in Fig. 2. We can see that the proposed algorithm perform competitively against the state-of-the-art method in all cases.

4.4. Comparisons with more state-of-the-art

In this section, we compare the overall performance (distance precision rate at 20 pixels) with more state-of-the-art methods published recently.[3] The results are summarized in Table 3. From the table, we can see that the proposed method on features extracted from ConvNets is able to perform favorably against those state-of-the-art solutions. Moreover, the proposed methods outperform the state-of-the-art multiple view and ensemble based IMT tracker by a large margin in terms of both accuracy and speed (IMT operates less than 5 frames in our setting).

4.5. Ablation study

In order to demonstrate the feasibility of the proposed co-trained methods, we also summarize the performance of the base trackers and other ensemble variants in Table 4. For ensemble variants, we compare the proposed method with average voting and maximum voting of two models trained separately. For average voting, the response is the average of two tracking responses. Similarly, maximum voting means selecting one response map from these two with the maximum prediction value. We also evaluate a simple tracker which works on concatenation of two different features.

We can see other ensemble variants such as averaging or choosing the maximum output cannot further boost the performance. This is because those two trackers are trained separately and the correct tracker can sometimes be corrupted by the other tracker. However, when trained jointly in the proposed method, the last term in our objective function (Eq. (7)) regularize each model in a pair-wise fashion which weighs the influence of pairwise disagreements. It is more likely for each base tracker to work on its own features and trying to correct the other one by continuous interaction through the tracking process because first term and the third

Table 3
Overall performance (distance precision rate at 20 pixels) comparisons with other state-of-the-art methods.

Method	Reference	DP rate(%)
SAMF	[36]	77.4
SRDCF	[18]	83.8
DeepSRDCF	[17]	84.9
MEEM	[83]	84.0
IMT	[82]	65.2
TGPR	[22]	75.9
DML	[32]	60.3
LCT	[47]	85.4
CNN-SVM	[29]	85.2
MUSTer	[31]	86.5
RPT	[37]	81.2
LHF	[68]	81.2
MKKCF	[63]	78.1
FCNT	[71]	85.6
CRVFL	[90]	86.2
CoKCF_HC	Proposed in This work	77.5
CoKCF_CNN	Proposed in This work	88.9

Table 4
Average precision scores of base models and other ensemble variants.

Method	Score	Method	Score
KCF_HOG	74.3	KCF_Conv5	84.9
KCF_Color	63.6	KCF_Conv4	81.3
KCF_HCAve	73.4	KCF_54Ave	84.2
KCF_HCMax	75.4	KCF_54Max	82.0
KCF_HCCon	74.1	KCF_54Con	81.4
CoKCF_HC	77.5	CoKCF_CNN	88.9

[3] Due to proprietary issues, results of some methods (such as LHF [68], DML [32] and so on) are not publicly available. Those results are obtained through personal communications with the authors.

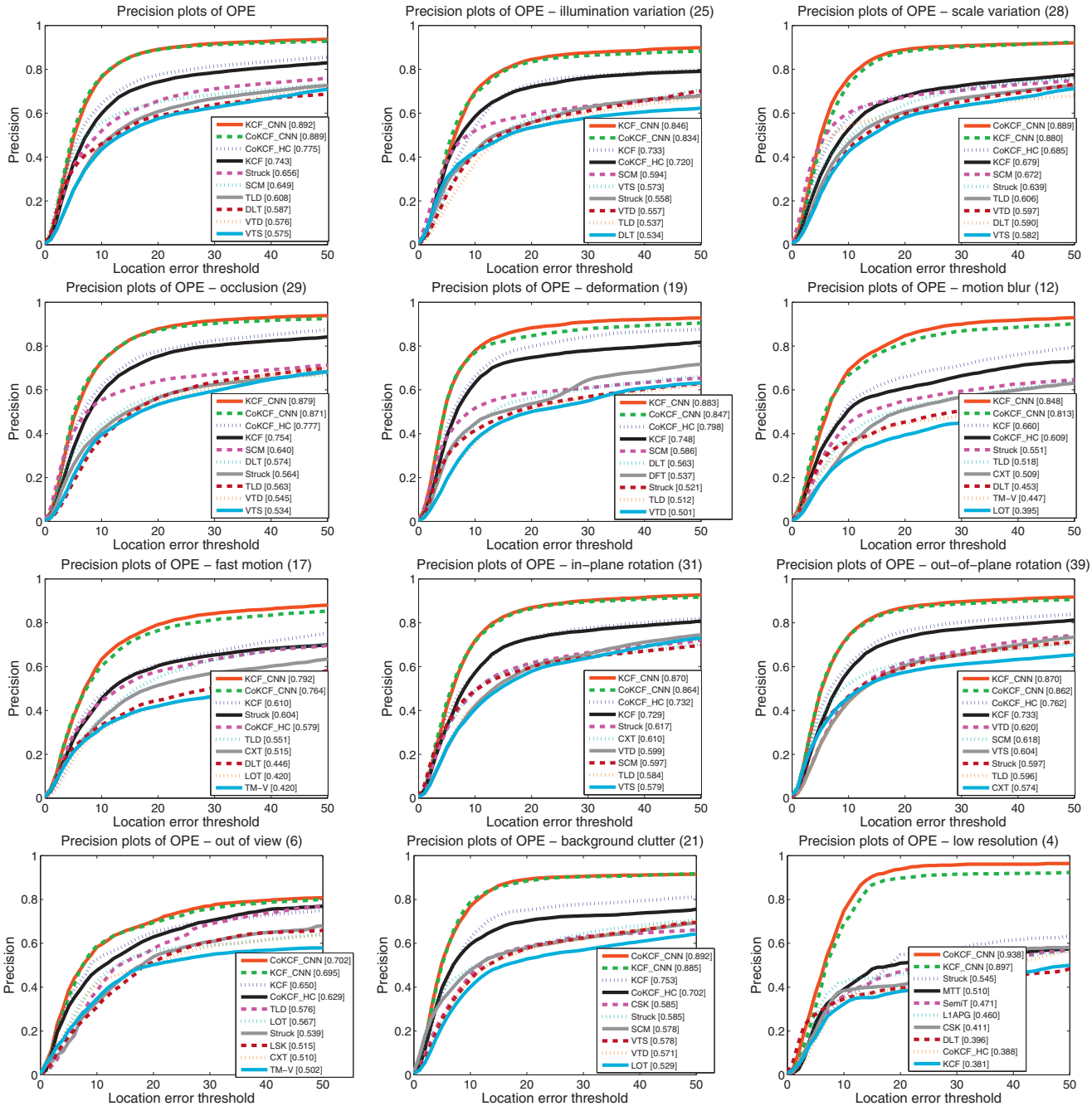


Fig. 2. The precision plots of the results.

term In Eq. (7) force each base tracker to be as accurate as possible. Moreover, simple concatenation of the features is not good because it may distort the intrinsic geometric structure of the data. Different sources contain different and partly independent information. In this case, designing a proper learning method is beneficial by reducing the noise, as well as by improving statistical significance and leveraging the interactions and correlations between data sources to obtain more refined and higher level information, which is also known as data fusion or data integration [67].

The parameter *beta* is set to be 10^{-4} for our experiment. In order to investigate the effectiveness of this parameter, we also report the overall performance of the proposed method with different numerical values for β in Fig. 3. Generally we found an inverted "*U*" shape. Setting β to be a very large value will force the co-trained model to ignore the first constrains in Eq. 7 and thus result in inaccurate solution. On the other hand, setting β to be a tiny value is equivalent to train two models independently. Please note the $x-axis$ in Fig. 3 is in log_{10} scale.

4.6. Results on more challenging video sequences

We now further analyze and illustrate the benefits of the proposed CoKCF tracker on another set of 12 challenging video datasets to reflect the real world scenarios of frequent occlusions and repetitive appearance variations. These data were used in [13,83] and is shown to be more challenging than those in the

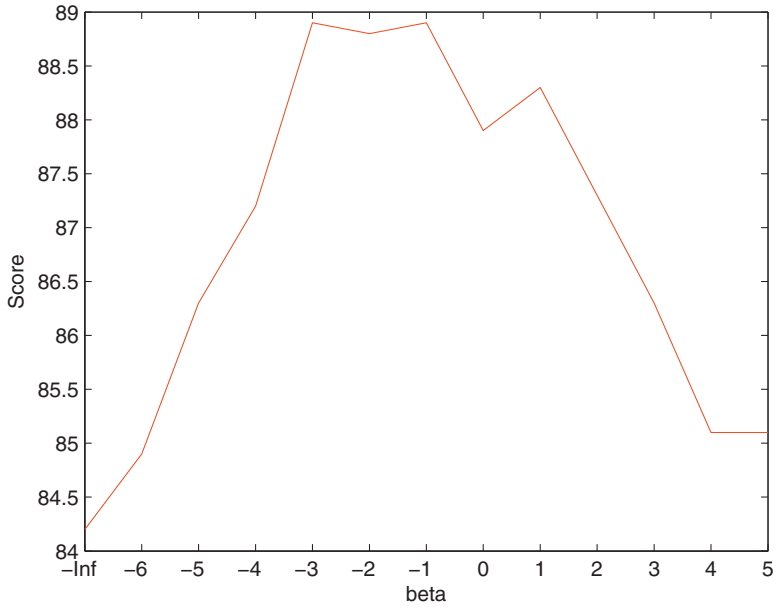


Fig. 3. Overall performance analysis of the β parameter. Note the $x-axis$ is in log_{10} scale.

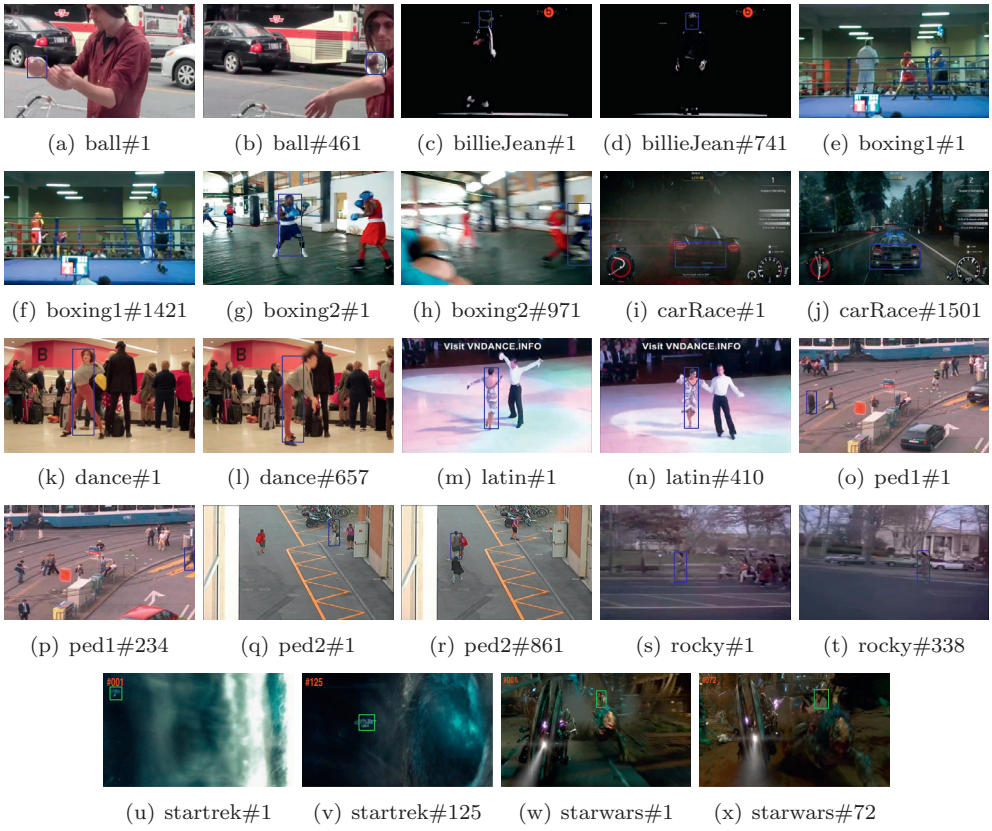


Fig. 4. Annotated image sequences for the 12 additional challenging datasets. Only the first and last frames are displayed for each video sequance. Best viewed in color.

commonly used benchmark [76]. The first and last frames with ground-truth bounding box are shown in Fig. 4.

We compare the proposed method with the baseline KCF and the KCF_CNN as it also achieves competitive results according to previous sections. Those results are summarized in Table 5.

Form Table 5 we can see that the proposed method KCF_HC and CoKCF_CNN outperform KCF tracker. Our proposed method CoKCF_CNN, with only two sets of correlation filters, outperforms the state-of-the-art KCF_CNN which consists of three sets of correlation filters.

Table 5
Performance comparisons of different methods on another 12 challenging datasets.

Data	KCF	KCF_CNN	CoKCF_HC	CoKCF_CNN
ball	96.1	58.4	97.8	96.7
billieJean	71.9	100.0	85.6	99.6
boxing1	26.5	58.7	36.6	30.3
boxing2	47.9	78.1	63.5	80.7
carRace	33.5	33.5	89.9	33.5
dance	22.8	40.2	25.4	37.9
latin	44.6	44.9	44.6	44.6
ped1	55.1	59.4	99.6	58.5
ped2	12.3	15.4	12.5	15.1
rocky	100.0	100.0	100.0	100.0
startrek	96.0	100.0	51.2	100.0
starwars	100.0	100.0	70.8	100.0
mean	58.89	65.72	64.79	66.41

4.7. Extension to more than 2 models

It is straightforward to get the solutions when we have more than 2 models in the same way. More specifically, let

$$G_i = (2\beta(M-1)+1)K_iK_i + \lambda K_i, \tag{15}$$

then

$$\nabla_{\alpha_i} T(\alpha) = 2G_i\alpha_i - 2K_iy - 4\lambda \sum_{j\neq i} K_iK_j\alpha_j, \tag{16}$$

It is straight-forward that the following equation must hold when α is at its optimum:

$$(\nabla_{\alpha_1} E(\alpha), \nabla_{\alpha_2} E(\alpha), \cdots, \nabla_{\alpha_M} E(\alpha))^\top = \boldsymbol{0}, \tag{17}$$

hence, we have

$$\begin{pmatrix} G_1 & -2\beta K_1K_2 & \cdots \\ -2\beta K_2K_1 & G_2 & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \end{pmatrix} = \begin{pmatrix} K_1y \\ K_2y \\ \vdots \end{pmatrix}, \tag{18}$$

However, for visual tracking, increasing the number of kernels used will further increase the computation complexity without too much benefit in overall performance. We have used the features from the 3rd, 4th and 5th layers of the same ConvNet, it performs almost the same (89.5% in mean precision) while operate with only 4fps on average.

5. Conclusion

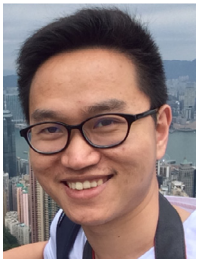
Robustness and speed have been regarded as a dilemma in visual tracking system for a long time. On one hand, single classifier may not be powerful enough to handle complex backgroundings and significant variations of target appearance, and ensemble methods seem to be a promising approach to significant improve the tracking performance [70]. On the other hand, modern trackers suffer from relatively large computational time, which is a major hurdle that hinders the ensemble based visual tracking.

This work extends the recently developed KCF tracker and makes a good trade-off between the robustness and speed. It presents a novel and effective joint training approach for ensemble tracking. We choose CoKCF, which is a simple realization of the proposed method with only 2 base trackers, to demonstrate our point. We demonstrate the effectiveness of the proposed method with handcrafted features and features extracted from ConvNets. Besides the recent proposed visual tracking benchmark, the proposed method achieves competitive results on another 12 challenging video sequences.

References

[1] A. Adam, E. Rivlin, I. Shimshoni, Robust fragments-based tracking using the integral histogram, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1, IEEE, 2006, pp. 798–805.
[2] T. Ahonen, A. Hadid, M. Pietikainen, Face description with local binary patterns: application to face recognition, IEEE Trans. Pattern Anal. Mach. Intell. 28 (12) (2006) 2037–2041.
[3] M. Amini, N. Usunier, C. Goutte, Learning from multiple partially observed views-an application to multilingual text categorization, in: Advances in Neural Information Processing Systems, 2009, pp. 28–36.
[4] S. Avidan, Ensemble tracking, IEEE Trans. Pattern Anal. Mach. Intell. 29 (2) (2007) 261–271.
[5] B. Babenko, M.-H. Yang, S. Belongie, Robust object tracking with online multiple instance learning, IEEE Trans. Pattern Anal. Mach. Intell. 33 (8) (2011) 1619–1632.
[6] Q. Bai, Z. Wu, S. Sclaroff, M. Betke, C. Monnier, Randomized ensemble tracking, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV)„ IEEE, 2013, pp. 2040–2047.
[7] T. Bai, Y.F. Li, Robust visual tracking with structured sparse representation appearance model, Pattern Recognit. 45 (6) (2012) 2390–2404.
[8] M.J. Black, A.D. Jepson, Eigentracking: robust matching and tracking of articulated objects using a view-based representation, In. J. Comput. Vision 26 (1) (1998) 63–84.
[9] D.S. Bolme, J.R. Beveridge, B. Draper, Y.M. Lui, Visual object tracking using adaptive correlation filters, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2010, pp. 2544–2550.
[10] U. Brefeld, T. Gärtner, T. Scheffer, S. Wrobel, Efficient co-regularised least squares regression, in: Proceedings of the 23rd International Conference on Machine Learning, ACM, 2006, pp. 137–144.
[11] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.
[12] L. Breiman, et al., Arcing classifier (with discussion and a rejoinder by the author), Ann.Stat. 26 (3) (1998) 801–849.
[13] D.M. Chu, A.W. Smeulders, Thirteen hard cases in visual tracking, in: Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance, IEEE, 2010, pp. 103–110.
[14] D. Comaniciu, V. Ramesh, P. Meer, Kernel-based object tracking, IEEE Trans. Pattern Anal. Mach. Intell. 25 (5) (2003) 564–577.
[15] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1, IEEE, 2005, pp. 886–893.
[16] M. Danelljan, G. Häger, F. Khan, M. Felsberg, Accurate scale estimation for robust visual tracking, in: Proceedings of the British Machine Vision Conference, BMVA Press, 2014.
[17] M. Danelljan, G. Hager, F. Shahbaz Khan, M. Felsberg, Convolutional features for correlation filter based visual tracking, in: Proceedings of the IEEE International Conference on Computer Vision Workshops, 2015a, pp. 58–66.
[18] M. Danelljan, G. Hager, F. Shahbaz Khan, M. Felsberg, Learning spatially regularized correlation filters for visual tracking, in: Proceedings of the IEEE International Conference on Computer Vision, 2015b, pp. 4310–4318.
[19] T.G. Dietterich, Ensemble methods in machine learning, in: Multiple Classifier Systems, Springer, 2000, pp. 1–15.
[20] T.G. Dietterich, Ensemble learning, in: The Handbook of Brain Theory and Neural Networks, 2, 2002, pp. 110–125.
[21] P. Domingos, A unified bias-variance decomposition, in: Proceedings of 17th International Conference on Machine Learning, 2000, pp. 231–238.
[22] J. Gao, H. Ling, W. Hu, J. Xing, Transfer learning based visual tracking with gaussian processes regression, in: Proceedings of the European Conference on Computer Vision, Springer, 2014, pp. 188–203.
[23] M. Gönen, E. Alpaydın, Multiple kernel learning algorithms, J. Mach. Learn. Res. 12 (2011) 2211–2268.
[24] H. Grabner, H. Bischof, On-line boosting and vision, in: Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition, 1, IEEE, 2006, pp. 260–267.
[25] H. Grabner, C. Leistner, H. Bischof, Semi-supervised on-line boosting for robust tracking, in: Proceedings of the European Conference on Computer Vision, Springer, 2008, pp. 234–247.
[26] R.M. Gray, Toeplitz and Circulant Matrices: A Review, NOW Publishers INC, 2006.
[27] J.F. Henriques, R. Caseiro, P. Martins, J. Batista, Exploiting the circulant structure of tracking-by-detection with kernels, in: Proceedings of European Conference on Computer Vision, Springer, 2012, pp. 702–715.
[28] J.F. Henriques, R. Caseiro, P. Martins, J. Batista, High-speed tracking with kernelized correlation filters, IEEE Trans. Pattern Anal. Mach. Intell. 37 (3) (2015) 583–596.
[29] S. Hong, T. You, S. Kwak, B. Han, Online tracking by learning discriminative saliency map with convolutional neural network, arXiv preprint arXiv:1502.06796 (2015a).
[30] Z. Hong, Z. Chen, C. Wang, X. Mei, D. Prokhorov, D. Tao, Multi-Store Tracker (MUSTer): a cognitive psychology inspired approach to object tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015b, pp. 749–758.
[31] Z. Hong, Z. Chen, C. Wang, X. Mei, D. Prokhorov, D. Tao, Multi-Store Tracker (MUSTer): a cognitive psychology inspired approach to object tracking, in: The IEEE Conference on Computer Vision and Pattern Recognition, 2015c.

[32] J. Hu, J. Lu, Y.-P. Tan, Deep metric learning for visual tracking, IEEE Trans. Circuits Syst. Video Technol. PP (99) (2015) 1–1. doi:10.1109/tcsvt.2015.2477936.
[33] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.
[34] J. Kwon, K.M. Lee, Visual tracking decomposition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2010, pp. 1269–1276.
[35] J. Kwon, K.M. Lee, Tracking by sampling and integrating multiple trackers, IEEE Trans.Pattern Anal. Mach. Intell. 36 (7) (2014) 1428–1441.
[36] Y. Li, J. Zhu, A scale adaptive kernel correlation filter tracker with feature integration, in: European Conference on Computer Vision, Springer, 2014, pp. 254–265.
[37] Y. Li, J. Zhu, S.C. Hoi, Reliable patch trackers: robust visual tracking by exploiting reliable patches, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 353–361.
[38] B. Liu, J. Huang, L. Yang, C. Kulikowsk, Robust tracking using local sparse appearance model and k-selection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2011, pp. 1313–1320.
[39] B. Liu, L. Yang, J. Huang, P. Meer, L. Gong, C. Kulikowski, Robust and fast collaborative tracking with two stage sparse optimization, in: Proceedings of the European Conference on Computer Vision, Springer, 2010, pp. 624–637.
[40] T. Liu, G. Wang, Q. Yang, Real-time part-based visual tracking via adaptive correlation filters, in: Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 4902–4912.
[41] Y. Liu, X. Yao, Ensemble learning via negative correlation, Neural Netw. 12 (10) (1999) 1399–1404.
[42] D.G. Lowe, Object recognition from local scale-invariant features, in: Proceedings of the IEEE Conference on Computer vision, 2, IEEE, 1999, pp. 1150–1157.
[43] H. Lu, S. Lu, D. Wang, S. Wang, H. Leung, Pixel-wise spatial pyramid-based hybrid tracking, IEEE Trans. Circuits Syst. Video Technol. 22 (9) (2012) 1365–1376.
[44] H. Lu, W. Zhang, Y.-W. Chen, On feature combination and multiple kernel learning for object tracking, in: Proceedings of the Asian Conference on Computer Vision, Springer, 2011, pp. 511–522.
[45] B.D. Lucas, T. Kanade, et al., An iterative image registration technique with an application to stereo vision., in: Proceedings of the International Joint Conference on Artificial Intelligence, 81, 1981, pp. 674–679.
[46] C. Ma, J.-B. Huang, X. Yang, M.-H. Yang, Hierarchical convolutional features for visual tracking, in: Proceedings of the IEEE International Conference on Computer Vision, 2015a.
[47] C. Ma, X. Yang, C. Zhang, M.-H. Yang, Long-term correlation tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015b, pp. 5388–5396.
[48] I. Matthews, T. Ishikawa, S. Baker, The template update problem, IEEE Trans. Pattern Anal. Mach. Intell. 26 (6) (2004) 810–815.
[49] X. Mei, Z. Hong, D. Prokhorov, D. Tao, Robust multitask multiview tracking in videos, IEEE Trans. Neural Netw. Learn. Syst. 26 (11) (2015) 2874–2890.
[50] X. Mei, H. Ling, Robust visual tracking using l1 minimization, in: Proceedings of the IEEE International Conference on Computer Vision, IEEE, 2009, pp. 1436–1443.
[51] X. Mei, H. Ling, Y. Wu, E. Blasch, L. Bai, Minimum error bounded efficient *l*1 tracker with occlusion detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2011, pp. 1257–1264.
[52] A.S. Razavian, H. Azizpour, J. Sullivan, S. Carlsson, Cnn features off-the-shelf: an astounding baseline for recognition, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW),, IEEE, 2014, pp. 512–519.
[53] Y. Ren, L. Zhang, P. Suganthan, Ensemble classification and regression-recent developments, applications and future directions [review article], IEEE Comput. Intell. Mag. 11 (1) (2016) 41–53.
[54] R. Rifkin, G. Yeo, T. Poggio, Regularized least-squares classification, Nato Sci. Ser. Sub Ser. III Comput.Syst. Sci. 190 (2003) 131–154.
[55] D.A. Ross, J. Lim, R.-S. Lin, M.-H. Yang, Incremental learning for robust visual tracking, Int. J. Comput. Vision 77 (1–3) (2008) 125–141.
[56] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., Imagenet large scale visual recognition challenge, Int. J. Comput. Vision (2014) 1–42.
[57] L. Sevilla-Lara, E. Learned-Miller, Distribution fields for tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2012, pp. 1910–1917.
[58] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556 (2014).
[59] V. Sindhwani, P. Niyogi, M. Belkin, A co-regularization approach to semi-supervised learning with multiple views, in: Proceedings of ICML Workshop on Learning with Multiple views, Citeseer, 2005, pp. 74–79.
[60] A.W. Smeulders, D.M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, M. Shah, Visual tracking: an experimental survey, IEEE Trans. Pattern Anal. Mach. Intell. 36 (7) (2014) 1442–1468.
[61] S. Sun, A survey of multi-view machine learning, Neural Comput. Appl. 23 (7–8) (2013) 2031–2038.
[62] J.S. Supancic, D. Ramanan, Self-paced learning for long-term tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2013, pp. 2379–2386.
[63] M. Tang, J. Feng, Multi-kernel correlation filter for visual tracking, in: Proceedings of the IEEE International Conference on Computer Vision, 2015.
[64] J. Van De Weijer, C. Schmid, J. Verbeek, D. Larlus, Learning color names for real-world applications, IEEE Trans. Image Process. 18 (7) (2009) 1512–1523.
[65] P. Viola, M.J. Jones, D. Snow, Detecting pedestrians using patterns of motion and appearance, in: Proceedings of the IEEE International Conference on Computer Vision, IEEE, 2003, pp. 734–741.
[66] D. Wang, H. Lu, M.-H. Yang, Online object tracking with sparse prototypes, IEEE Trans. Image Process. 22 (1) (2013a) 314–325.
[67] H. Wang, F. Nie, H. Huang, Multi-view clustering and feature learning via structured sparsity., in: Proceedings of the International Conference on Machine Learning, 2013b, pp. 352–360.
[68] L. Wang, T. Liu, G. Wang, K.L. Chan, Q. Yang, Video tracking using learned hierarchical features, IEEE Trans. Image Process. 24 (4) (2015a) 1424–1435.
[69] N. Wang, S. Li, A. Gupta, D.-Y. Yeung, Transferring rich feature hierarchies for robust visual tracking, arXiv preprint arXiv:1501.04587 (2015b).
[70] N. Wang, J. Shi, D.-Y. Yeung, J. Jia, Understanding and diagnosing visual tracking systems, in: Proceedings of the IEEE International Conference on Computer Vision, 2015c, pp. 3101–3109.
[71] N. Wang, J. Shi, D.-Y. Yeung, J. Jia, Understanding and diagnosing visual tracking systems, in: Proceedings of the IEEE International Conference on Computer Vision, 2015d.
[72] N. Wang, D.-Y. Yeung, Learning a deep compact image representation for visual tracking, in: Advances in Neural Information Processing Systems, 2013, pp. 809–817.
[73] N. Wang, D.-Y. Yeung, Ensemble-based tracking: aggregating crowdsourced structured time series data, in: Proceedings of the International Conference on Machine Learning, 2014, pp. 1107–1115.
[74] Q. Wang, F. Chen, W. Xu, M.-H. Yang, Object tracking via partial least squares analysis, IEEE Trans. Image Process. 21 (10) (2012a) 4454–4465.
[75] Q. Wang, F. Chen, J. Yang, W. Xu, M.-H. Yang, Transferring visual prior for online object tracking, IEEE Trans. Image Process. 21 (7) (2012b) 3296–3305.
[76] Y. Wu, J. Lim, M.-H. Yang, Online object tracking: a benchmark, in: Proceedings of the IEEE Conference on Computer vision and pattern recognition, IEEE, 2013, pp. 2411–2418.
[77] Z. Xiao, H. Lu, D. Wang, L2-RLS-based object tracking, IEEE Trans. Circuits Syst. Video Technol., 24 (8) (2014) 1301–1309.
[78] C. Xu, D. Tao, C. Xu, A survey on multi-view learning, arXiv preprint arXiv:1304.5634 (2013).
[79] F. Yang, H. Lu, M.-H. Yang, Robust superpixel tracking, IEEE Trans. Image Process. 23 (4) (2014a) 1639–1651.
[80] F. Yang, H. Lu, M.-H. Yang, Robust visual tracking via multiple kernel boosting with affinity constraints, IEEE Trans. Circuits Syst. Video Technol. 24 (2) (2014b) 242–254.
[81] J. Yang, Y. Li, Y. Tian, L. Duan, W. Gao, Group-sensitive multiple kernel learning for object categorization, in: IEEE 12th International Conference on Computer Vision, IEEE, 2009, pp. 436–443.
[82] J.H. Yoon, M.-H. Yang, K.-J. Yoon, Interacting multiview tracker, IEEE Trans. Pattern Anal. Mach. Intell. 38 (5) (2016) 903–917.
[83] J. Zhang, S. Ma, S. Sclaroff, MEEM: robust tracking via multiple experts using entropy minimization, in: Proceedings of the European Conference on Computer Vision, Springer, 2014a, pp. 188–203.
[84] K. Zhang, H. Song, Real-time visual tracking via online weighted multiple instance learning, Pattern Recognit. 46 (1) (2013) 397–411.
[85] K. Zhang, L. Zhang, Q. Liu, D. Zhang, M.-H. Yang, Fast visual tracking via dense spatio-temporal context learning, in: Proceedings of the European Conference on Computer Vision, Springer, 2014b, pp. 127–141.
[86] K. Zhang, L. Zhang, M.-H. Yang, Real-time object tracking via online discriminative feature selection, IEEE Trans. Image Process. 22 (12) (2013) 4664–4677.
[87] L. Zhang, P. Suganthan, Oblique decision tree ensemble via multisurface proximal support vector machine,, IEEE Trans. Cybern. 45 (10) (2015) 2165–2176, doi:10.1109/TCYB.2014.2366468.
[88] L. Zhang, P.N. Suganthan, Random forests with ensemble of feature spaces, Pattern Recognit. 47 (10) (2014) 3429–3437.
[89] L. Zhang, P.N. Suganthan, Visual tracking with convolutional random vector functional link network, IEEE Trans. Cybern. PP (99) (2016a) 1–11, doi:10.1109/TCYB.2016.2588526.
[90] L. Zhang, P.N. Suganthan, Visual tracking with convolutional random vector functional link neural network, IEEE Trans. Cybern. (2016), doi:10.1109/TCYB.2016.2588526.
[91] W. Zhong, H. Lu, M.-H. Yang, Robust object tracking via sparse collaborative appearance model, IEEE Trans. Image Process. 23 (5) (2014) 2356–2368.



Le Zhang received his B.Eng. degree from University of Electronic Science and Technology of China in 2011. He got his M.sc and PhD degree from Nanyang Technological University in 2012 and 2016, respectively. He is now working as a researcher in Advanced Digital Sciences Center (a Singapore-based research center of the University of Illinois at Urbana-Champaign). His research interest includes pattern classification and computer vision.

Ponnuthurai Nagaratnam Suganthan received the B.A degree, Postgraduate Certificate and M.A degree in Electrical and Information Engineering from the University of Cambridge, UK in 1990, 1992 and 1994, respectively. After completing his PhD research in 1995, he served as a pre-doctoral Research Assistant in the Dept of Electrical Engineering, University of Sydney in 1995-96 and a lecturer in the Dept of Computer Science and Electrical Engineering, University of Queensland in 1996-99. He moved to NTU in 1999. He is an Editorial Board Member of the Evolutionary Computation Journal, MIT Press. He is an associate editor of the IEEE Trans on Cybernetics (2012 -), IEEE Trans on Evolutionary Computation (2005 -), Information Sciences (Elsevier) (2009 -), Pattern Recognition (Elsevier) (2001 -) and Int. J. of Swarm Intelligence Research (2009 -) Journals. He is a founding co-editor-in-chief of Swarm and Evolutionary Computation (2010 -), an SCI Indexed Elsevier Journal. He was selected as one of the highly cited researchers by Thomson Reuters in 2015 and 2016 in computer science, also known as the World's Most Influential Scientists 2015 and 2016. He served as the General Chair of the IEEE SSCI 2013. He has been a member of the IEEE (S'90, M'92, SM'00, F'15) since 1990 and an elected AdCom member of the IEEE Computational Intelligence Society (CIS) in 2014–2016.