Deformable Part-based Tracking by Coupled Global and Local Correlation Filters

Osman Akin, Erkut Erdem, Aykut Erdem

Hacettepe University, Ankara, Turkey

Krystian Mikolajczyk

Imperial College, London, England

Abstract

Correlation filters have recently attracted attention in visual tracking due to their efficiency and high performance. However, their application to long-term tracking is somewhat limited since these trackers are not equipped with mechanisms to cope with challenging cases like partial occlusion, deformation or scale changes. In this paper, we propose a deformable part-based correlation filter tracking approach which depends on coupled interactions between a global filter and several part filters. Specifically, local filters provide an initial estimate, which is then used by the global filter as a reference to determine the final result. Then, the global filter provides a feedback to the part filters regarding their updates and the related deformation parameters. In this way, our proposed collaborative model handles not only partial occlusion but also scale changes. Experiments on two large public benchmark datasets demonstrate that our approach gives significantly better results compared with the state-of-the-art trackers.

Keywords: visual tracking, correlation filter tracking, collaborative model, deformable part-based model

1. Introduction

The main goal of object tracking is to determine the location of a given object, usually specified with a bounding box in the first frame, in the subsequent video frames. Tracking has a wide range of applications including but not limited to visual surveillance, human computer interaction, robotics, automatic navigation and action recognition. Although recent studies have reported promising results, tracking is still considered as a difficult problem due to the challenges such as occlusion, illumination variation, changes in scale, object deformations and in or out of plane rotations, etc.

Object tracking methods can be divided into two broad categories as generative and discriminative. The first group of works learns a generic model from the given initial conditions and then defines tracking as a search problem where the image region that is closest to the object of interest is determined within a neighborhood. On the other hand, the second group, which is generally referred to as tracking-by-detection, formulates tracking as a binary classification problem [1] [2] [3]. They mainly employ HoG[4], Haar-like features [1] and region covariances [5] as the object representations and use SVM[2], Multiple Instance Learning [1], Random Forests [6], and Ada-boost [7] as the visual classifiers. Recently, correlation filters [8, 9, 10, 4, 11, 12] give very promising results while offering very low processing time. These trackers simply try to maximize the correlation between a trained object template with the current frame in an online fashion, and since the processing is usually done in the Fourier domain, they are very fast.

In this paper, we propose a robust generic object based tracking algorithm. In particular, we suggest to combine recently proposed tracking-by-detection schema of correlation filters with deformable part-based model within a unified framework of coupled global and local filters. In our tracker, which we call deformable part-based correlation filter tracking (DPCF), the appearance model is implemented by a correlation filter based on histogram-of-gradient (HoG) features and color, while the deformable model is based on the relative arrangement of parts. The object is represented with several local filters, which have a deformable structure, and an additional global appearance filter that has a coupled interaction with the part-based schema. Specifically, local filters help us to track a target object by focusing on specific object parts and provide a rough solution for the target's position. Global filter, however, is learned by considering the whole object region and employs the approximate solution from the local filters to determine the final solution. Moreover, it provides feedbacks to the local filters regarding the utilized deformation model and the new parameters.

Our contributions: We make four main contributions to the long-term online visual tracking problem: i) We propose a deformable part-based tracking framework based on correlation filters. ii) We present a collaborative algorithm for tracking-by-detection with coupled local and global filters. iii) We introduce a simple yet natural model for handling scale changes, and a robust update scheme that addresses occlusion, scaling and fast motion issues while keeping the processing in real time. iv) We improve the state-of-the-art tracking results in comparison to the recent successful trackers on two large scale public benchmark datasets [13].

The rest of the paper is organized as follows. In Section 2, we review recent tracking algorithms. Section 3, we give a detailed description of correlation based tracking and the proposed coupled global and local model. In Section 4, we provide a thorough experimental analysis by comparing our tracker against the state-of-the-art trackers. In Section 5, we conclude with some discussions and directions for the future research.

2. Related work

There is a vast literature on visual object tracking, so here we briefly review some related approaches to the proposed tracking algorithm. For a comprehensive review of the tracking methods, the readers can refer to [14, 15, 13, 16].

A significant number of recent trackers use learning-based formulations. For example, MILTrack [1] tracking algorithm employs a multiple instance learning to keep track of the target object. Struck [2] was proposed as an online SVM based tracker in which online learning, sample selection and search for the best match modules works together to find the target object in a video frame. In TLD [3], tracking, learning and detection components are employed simultaneously by additionally forcing some structured constraints about the target object. TGPR [17] analyzes the probability of Gaussian Processes Regression based on a semi-supervised learning framework. On the other hand, MEEM [18] employs an ensemble learning approach which keeps track of the history of appearance changes based on a minimum entropy criterion.

Another group of algorithms employs specific object representations that have advantages for the underlying tracking process. For example, the trackers in [19, 20] use sparse representations while the recently proposed TLP [21] tracker employs And-Or graphs (AOG) to represent the target object. Deformable part-based model with discriminatively trained parts is offered in [22] for object detection and used in SPOT [23] as a tracker with SVM classifier. A coupled-layer model(LGT) is suggested in [24] to support a constraint paradigm between the adaptation of the global and the local layer that is based on visual properties such as color, shape and apparent local motion.

All the aforementioned models, to a certain extent, suffer from challenging conditions such as occlusions, changes in the illumination, etc. and do not keep up with real time performance. LGT [24] is similar to our study, but it needs a large sets of parts that leads to extra computational cost. Unlike this algorithm, our approach uses only a few correlation filters and has an efficient update mechanism which also estimate scaling changes. In addition, our approach takes into account the structural preservation of parts which is handled just like the graph structure in SPOT [23].

2.1. Correlation Filter Based Tracking

A recent trend in tracking is to employ correlation filter based formulations. A detailed experimental survey can be found in [25]. These trackers model the appearance of the target object with a specific filter that is trained from the initial or past frames, and it serves as the appearance model for the new frame. Every frame is correlated with the trained model and the maximum value of the correlation indicates the new position of the object. To construct a robust and efficient tracker, the correlation is implemented in the Fourier domain using Fast Fourier Transform (FFT). In particular, a Fourier transform of the input image $F(\mathbf{x})$ and the filter $F(\mathbf{h})$ are estimated, and then element-wise multiplication is carried out with the complex conjugate $F(\mathbf{x})^*$ as stated by the correlation theorem:

$$\mathbf{y} = F^{-1}(F(\mathbf{h}) \odot F(\mathbf{x})^*) \ . \tag{1}$$

The correlation output $\mathbf{y}$ is then transformed into the spatial domain with inverse FFT. Finally, the maximum response of the correlation gives the estimated object center. All the computations are done in the Fourier domain, and thus the resulting tracker is highly efficient, reaching a 300 frames-per-second (fps).

In MOSSE [8], a filter is trained to minimize the error between the actual and predicted output of the correlation. Affine warped windows from the first frame are used to construct a training set and an weighted average is used to quickly adapt the filters to the appearances changes:

$$F(\mathbf{h}_t) = \eta \cdot F(\mathbf{h}_t) + (1 - \eta) \cdot F(\mathbf{h}_{t-1}) \ , \tag{2}$$

where $\mathbf{h}_t$ is the current filter and $\mathbf{h}_{t-1}$ is the previous with η being the update rate. MOSSE is extended in [12] to deal with the scaling problem which makes use of scale-space pyramid representation.

Another highly efficient solution was offered in CSK [9] in which circulant matrices are employed, allowing correlation filters to be kernelized. In PBTLD[10], an extension to CSK was proposed, where the authors consider a part-based scheme to address the partial occlusion problem within the online tracking-learning-detection framework. In ACT [11], CSK tracker was extended with color attributes and an adaptive scheme to set the weights of these attributes. Similarly, another recently proposed KCF [4] tracker employs multi-channel features based on HoG instead of a raw pixel based representation. The HoG features further improve the performance of the kernelized filters while still keep the running time on the level of hundreds frames-per-second. In [26], the authors suggest an image pyramids-based framework for KCF, which integrates the HoG features with the color attributes, to tackle the scale problem.

Recently, another correlation filter based tracker (RTCF) was offered in [27], where the authors also employ a multi-part structure as in our work. While they consider adaptive weights for the filters corresponding to object parts, we consider a coupled formulation where the part-based local filters and the object-based global filter work in a unified manner to keep track of the target object. More specifically, RTCF uses the Bayesian inference framework to combine the tracking scores of different parts but not mention about determining parts initial position. It is not, meanwhile, clear that all parts represent the target entirely. This leads to missing some important parts and needs to be known initial position of prominent parts that is not suitable for general object tracking. Unlike RTCF, our framework supports structural constraints between parts using a deformation model and determines parts position according to a global filter and a deformable model. Thus, our couple model handles drifting more effectively and moreover, it provides scaling invariance in a straightforward way. In [28], a different correlation filter is proposed in which tracking is decomposed into translation and scale estimation problems thus explicitly addressing the scale changes. Very recently, a collaborative correlation [29] tracker is offered to deal with the scale variation by embedding the scale factor into the kernelized matrix and a long-term filter is offered for re-detection.

3. Proposed Tracker

In this section, we first provide the motivation and formulation of the proposed coupled global and local part based model. We then discuss the details of the model, the update scheme, scaling and our implementation details.

3.1. Problem Formulation

Traditional correlation filter-based trackers [8, 11, 4, 9, 26] provide robustness against appearance changes, at the same time, offer very fast computation times. Most of these trackers consider representation schemes which consist of global appearance models based on different visual features such raw intensities [8, 4, 9], HoG [12, 4, 26] or color [11, 26].

Employing a global appearance model for the target object makes the tracking vulnerable to partial occlusions or appearance changes of the object. Yet, in most cases, there remains some object parts that are less affected by these changes. Splitting an object into parts and modelling their a spatial arrangement provide supplementary shape knowledge and more importantly, individual tracking of parts introduces extra robustness. The approaches in [10, 27] also employ a part-based local appearance model that depends on correlation filters. They showed that using such a strategy better handles the partial occlusions, however their formulation lacks a proper mechanism to adapt the part configurations.

In this study, we develop a tracking model which aims to leverage the collaborative power of both representation schemes in which global and local correlation filters are tied together by a unified framework (Figure 1). We describe both the target object and its parts as rectangular patches for the global and local correlation filters, respectively, and train them by using a combination of the HoG [30] and color features. We denote a patch $(m \times n)$ located at l with $\mathbf{x}$ which is obtained by concatenating the individual HoG channels as $\mathbf{x} = [x_1, ..., x_C]$ with $C = 31$ representing the number of gradient orientation bins. Then, the 31 dimensional HOG vector is concatenated to 11 dimensional color attributes (CN), giving us a 42 dimensional vector.

At the training step, we consider the dense sampling strategy introduced in [4], and select the training samples automatically by using all the cyclic shifts of the input patch. For the labels of these samples, we use a Gaussian function y with $y = 1$ near the target location (the location of the object or one of its parts) and which decays to $y = 0$ as further away from it.

Parts are constructed as separate filters but they are connected with a deformable spring model that keeps track of the length and direction between the parts and the object center. Once these filter responses are computed, the target object location is updated according to the deformable model. This schema leads to keeping the shrinking or stretching parts together in a robust way. The global filter is updated according to the reliability scores of each part. We require at least one part to be reliable to use their responses as a reference point. Additionally, part filters are updated if they are considered as reliable. Otherwise, the previous state information of the collaborative model is kept fixed. Finally, compressing and stretching of parts are used as a simple model to capture the scaling.

3.2. Global Tracking

As our global tracker, we employ Kernelized Correlation Filter (KCF) tracker [4]. The objective of KCF can be expressed in the spatial domain as finding the optimal filter $\mathbf{h}$ by solving the following ridge regression problem,

$$\min_{\mathbf{h}} \sum_{i=1}^{n} (y_i - \mathbf{h}^T \mathbf{x}_i)^2 + \lambda \|\mathbf{h}\|_2^2 \,, \tag{3}$$

where $\mathbf{h}^T\mathbf{x}$ is the filter response, $y_i \in \mathbb{R}^D$ is the desired correlation response for the i-th observation $\mathbf{x}_i \in \mathbb{R}^D$, typically constructed as a Gaussian function centered at the

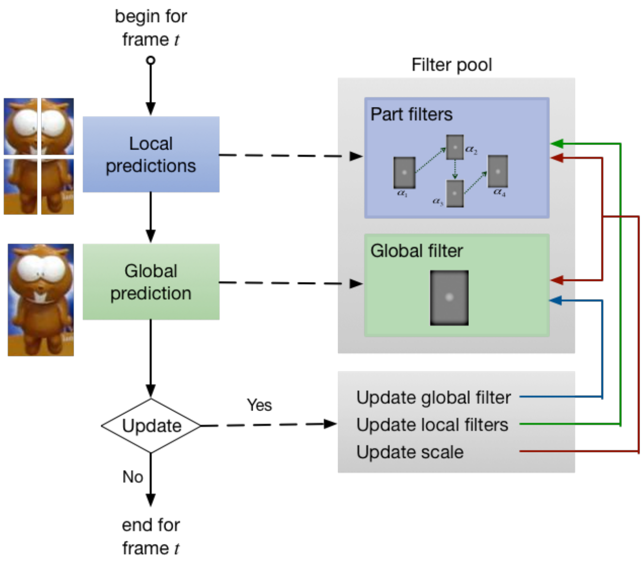

Figure 1: Deformable part-based schema with a unified framework of coupled-layer global and local filters.

location of the object to be tracked, and $\lambda \geq 0$ is a regularization parameter.

In order to achieve a better performance, Henriques [4] utilized kernel ridge regression to extend (3) to a nonlinear setting where the optimum filter is given by the following closed form solution:

$$\boldsymbol{\alpha} = (K + \lambda I)^{-1} \mathbf{y} \ , \tag{4}$$

with $\boldsymbol{\alpha}$ being the vector of coefficients α_i, representing the filter $\mathbf{h}$ in the dual space, K being the kernel matrix, with elements $K_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$, I being the identity matrix, and $\mathbf{y}$ being the vector of desired responses y_i.

The key characteristics of KCF is that Eq. 4 can be efficiently solved for a kernel matrix with a circulant structure, without building the kernel explicitly while considering all possible samples which correspond to patches around the object at different translations. Given such a kernel matrix $K = C(\mathbf{k}^{\mathbf{xx}})$, where $\mathbf{k}^{\mathbf{xx}}$ is called the kernel correlation of $\mathbf{x}$ with itself, and $C(\cdot)$ is the circulant data matrix formed by concatenating all possible cyclic shifts, then the solution is simply given by

$$\boldsymbol{\alpha} = \mathcal{F}^{-1}\left(\frac{\mathcal{F}(\mathbf{y})}{\mathcal{F}(\mathbf{k}^{\mathbf{xx}}) + \lambda}\right) . \tag{5}$$

In the experiments, we used HoG and color features, and the following multi-channel version of the Gaussian kernel, which are both shown to perform well in [4]:

$$\mathbf{k}^{\mathbf{xx}'} = \exp\left(-\frac{1}{\sigma^2}\left(\|\mathbf{x}\|^2 + \|\mathbf{x}'\|^2 - 2\mathcal{F}^{-1}(\sum_c \hat{\mathbf{x}}_c^* \odot \hat{\mathbf{x}}'_c)\right)\right) , \tag{6}$$

where $\hat{\mathbf{x}} = \mathcal{F}(\mathbf{x})$, $\hat{\mathbf{x}}^*$ is the complex-conjugate of $\hat{\mathbf{x}}$, and $\odot$ denote the element-wise product.

The same property can also be used in the detection step as follows. In a frame, a patch is extracted at the previous object location and represented with HoG and color features. Then, the response of the learned filter is computed as

$$\mathbf{y} = \mathcal{F}^{-1}\left(\mathcal{F}(\mathbf{k}^{\mathbf{xz}}) \odot \mathcal{F}(\boldsymbol{\alpha})\right) , \tag{7}$$

where $\mathbf{k}^{\mathbf{xz}}$ is the kernel correlation of $\mathbf{x}$ and $\mathbf{z}$, as defined in Eq. 6. Finally, the new object location $\boldsymbol{\ell}$ is estimated by finding the translation that maximizes the filter response $\tilde{\mathbf{y}}$.

3.3. Local Tracking

There are several different strategies to define the part-object relationships. One category of approaches (e.g., [31]), which can be considered as domain-specific trackers, use explicit part structures, e.g. a human can be represented by parts corresponding to head, limbs and torso. More generic trackers, on the other hand, consider arbitrary patches to define these parts which do not depend on any predetermined object model. Representing the target by means of a star model (e.g., [23]) or a group of arbitrary rectangular patches (e.g., [32, 19]) are the most common part-object relations for generic trackers.

Motivated by the effectiveness of the part-based trackers, we present here a generic tracking framework that considers part-specific local correlation filters to keep track of the target. Specifically, for our local tracking framework, we employ a number local correlation filters, one for each part of the object, and treat these filters as individual

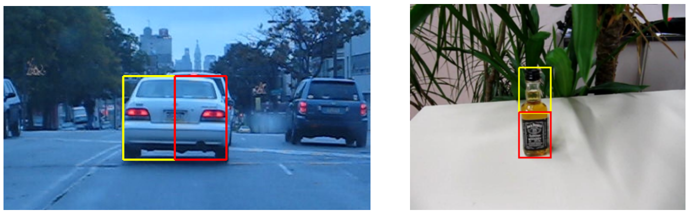

Figure 2: The initial part-object relations used in our experiments. The layout is selected according to aspect ratio of the target object. Sequences are from *blurCar2* and *liquor* [33], respectively.

trackers which are interconnected with each other through some structural constrains. While the parts-based schema can offer a generic framework, in our experiments we consider two different spatial layouts are found to be effective enough, one for the horizontally and one for the vertically aligned objects, as illustrated in Figure 2.

As in the global tracking, we employ the following closed form solution to learn the optimum coefficients of our local filters:

$$\boldsymbol{\alpha}_i = \mathcal{F}^{-1}\left(\frac{\mathcal{F}(\mathbf{y}_i)}{\mathcal{F}(\mathbf{k}_i^{\mathbf{xx}}) + \lambda}\right) , \tag{8}$$

where $\boldsymbol{\alpha}_i$ denotes the coefficient vector of the local filter that keeps track of the part i, K_{p_i} is the kernel matrix, and $\mathbf{y}_i$ represents the vector of desired responses for the part i. Once the filter coefficients $\boldsymbol{\alpha}_i$ are trained, then we use the following equation to detect the defined object parts:

$$\mathbf{y}_i = \mathcal{F}^{-1}\left(\mathcal{F}(\mathbf{k}_i^{\mathbf{xz}}) \odot \mathcal{F}(\boldsymbol{\alpha}_i)\right) , \tag{9}$$

with $\mathbf{k}_i^{\mathbf{xz}}$ denoting the kernel correlation of the learned part appearance $\mathbf{x}_i$ with the test image feature $\mathbf{z}_i$. New location of each object part $\boldsymbol{\ell}_i$ is then found by estimating the translation that maximizes the response $\mathbf{y}_i$ of the local filter $\boldsymbol{\alpha}_i$.

Moreover, for every part, we also store a displacement vector Δ_i that represents the length and direction of the vector between part i and the center of the target. Note that these structural constraints between parts are updated at each time frame in collaboration with the global filter that keeps track of the object, which further enhances the robustness of the whole tracking process as discussed in the next section.

3.4. Collaborative Deformable Part-based Model

We present a collaborative framework which employs a coupled system of global and local correlation filters. The target is represented using a part-object relationship as illustrated in Figure 2. A separate global and a number of local correlation filters are respectively trained with the whole object patch and the related part patches in the initial frame. While tracking, the filters contribute to a robust prediction of the target location by jointly aggregating their predictions through a common layer.

Specifically, each local filter $\boldsymbol{\alpha}_i^{t-1}$ is used to detect the corresponding object part. These detections are then used to estimate an initial prediction for the target using the current part-object model:

$$\hat{\boldsymbol{\ell}}^t = \frac{1}{\mathcal{Z}} \sum_i \mathbf{r}_i * \left(\boldsymbol{\ell}_i^{t-1} + \Delta_i^{t-1} \right) \ , \tag{10}$$

where $\boldsymbol{\ell}_i$ is the detected part location and Δ_i is the deformation vector between the part and the object center and the $\mathbf{r}_i$ is the parts' reliability with $\mathcal{Z} = \sum_i \mathbf{r}_i$.

To avoid erroneous estimations, we only use new information from the reliable parts. These parts are identified by comparing detection scores $\mathbf{r}_i$ against a threshold ψ. To normalize the scores between the parts, the likelihood of each part is modeled as

$$\mathbf{r}_i = \frac{1}{1 + \exp(-\mathbf{y}_i)} \ . \tag{11}$$

As a result, a part which is occluded or which has a large pose or appearance change has no effect on the final detection at frame t when combined with the global filter. This coarse result is then used to define the search neighborhood and employed in conjunction with the global filter $\boldsymbol{\alpha}$ to estimate the final target location $\boldsymbol{\ell}^t$. In particular, the local parts find the target according to the utilized deformable model, and then the estimated target position is used as a rough solution for global filter on the condition that at least one of the parts provides a reliable solution. Otherwise, the global filter takes into account the previous result of the global filter (from the preceding frame) for detection. Afterwards, this final result of the global filter provides a feedback to the part filters regarding the new reference location which is used within the deformable model. This process is summarized in Algorithm 1 and Figure 1. Consequently, the global and

Algorithm 1: Tracking

```
Tracking $t > 1$:
    for each part $i$ do
        Perform part detection (Eq. 9)
    end
    Estimate the target location $\hat{\boldsymbol{\ell}}$ suggested by the reliable parts (Eq. 10)
    Perform global detection over $\hat{\boldsymbol{\ell}}$ (Eq. 7) to obtain $\boldsymbol{\ell}$ (output of frame $t$)
    Update the global model (Eq. 12-13) and the scale (Eq. 16)
    if scaling is occured then
        re-initialize the whole system according to scaling
    end
    for each part do
        Update part location and deformation vector $\Delta_i$ between part and object
        centers
        Update local model (Eq. 14-15)
    end
end
```

the local filters play equally important roles in object tracking. For the reliable parts, an update mechanism is introduced as discussed in the next section.

3.5. Update Scheme

During tracking, the target object may become occluded by other objects, or its pose or appearance may undergo significant changes. We address these situations with an update scheme of the global and the local correlation filters. For the global filter, we train a new filter $(\boldsymbol{\alpha}^t, \mathbf{x}^t)$ at the new target location only if there exists at least one reliable part by linearly interpolating the newly estimated filter coefficients $\hat{\boldsymbol{\alpha}}^{t-1}$ (Eq. 5) and the appearance features $\hat{\mathbf{x}}^{t-1}$ with the ones from the previous frame $t-1$. If all the parts are considered unreliable, we keep the previous filters:

$$\boldsymbol{\alpha}^t = \begin{cases} (1-\gamma)\boldsymbol{\alpha}^{t-1} + \gamma\boldsymbol{\alpha} & \text{if } \exists i \mathbf{r}_i > \psi \\ \boldsymbol{\alpha}^{t-1} & \text{otherwise .} \end{cases} \tag{12}$$

$$\mathbf{x}^t = \begin{cases} (1-\gamma)\mathbf{x}^{t-1} + \gamma\mathbf{x} & \text{if } \exists i \mathbf{r}_i > \psi \\ \mathbf{x}^{t-1} & \text{otherwise ,} \end{cases} \tag{13}$$

where γ is a learning rate parameter, which determines how fast the tracker adapts itself to the changes in the scenes.

For the local filters, we consider independent update mechanisms in which a new filter $(\boldsymbol{\alpha}_i^t, \mathbf{x}_i^t)$ is trained for each part i again by adapting the learned filter and the appearance properties from the previous frame $\left(\boldsymbol{\alpha}_i^{t-1}, \mathbf{x}_i^{t-1}\right)$ using the current observations $\mathbf{x}_i$ and the estimated coefficients $\boldsymbol{\alpha}_i$:

$$\boldsymbol{\alpha}_i^t = \begin{cases} (1-\gamma)\boldsymbol{\alpha}_i^{t-1} + \gamma\boldsymbol{\alpha}_i & \text{if } \mathbf{r}_i > \psi \\ \boldsymbol{\alpha}_i^{t-1} & \text{otherwise ,} \end{cases} \tag{14}$$

$$\mathbf{x}_i^t = \begin{cases} (1-\gamma)\mathbf{x}_i^{t-1} + \gamma\mathbf{x}_i & \text{if } \mathbf{r}_i > \psi \\ \mathbf{x}_i^{t-1} & \text{otherwise ,} \end{cases} \tag{15}$$

with γ denoting the learning rate. As can be seen, these updates are performed as long as the parts are reliable, i.e. the part detection scores in the current frame is larger than a predefined threshold. Thus, for the unreliable parts we keep the previous filters. In this way, the local filters become more robust against occlusions or other significant changes. Moreover, we also update the part locations using the new target location and the part-object relation only if the part is considered reliable, which provides extra robustness. In Figure 3, we show sample frames from a video sequence, where the target object is occluded. As it can be seen, the collaborative model accurately tracks the target even if parts of the objects are not always visible through the tracking process.

3.6. Scaling

It is important to update the scale of filters to handle different sizes of the target object for visual tracking and to eliminate the drifting problem. Most of the correlation filter based trackers [28],[12],[26],[27] have tried to estimate the object size by using

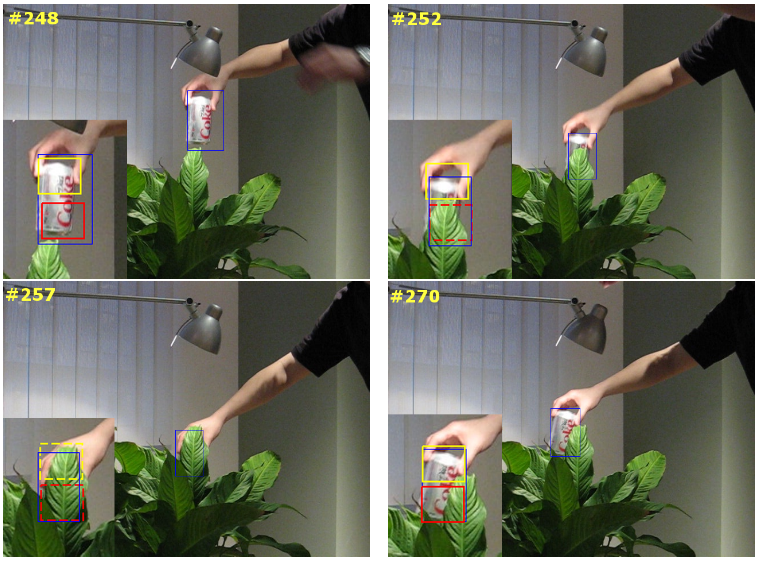

Figure 3: Tracking under significant occlusions. Occluded parts are shown with dashed line which are not updated due to low detection confidence.

a discriminative filter and a search pool that is based on pyramidal structure. Despite offering remarkable results, these trackers can not estimate the current object size. To our knowledge all previous correlation filter-based trackers did not consider changing the filter size in an adaptive way. The size of the target may change dramatically in most of the time, and therefore employing a search strategy which considers fixed scales is not enough to estimate the size accurately.

In our study, we explore adaptively changing the size of a filter using the proposed part-based schema for more accurate detection. In the suggested scheme, two reliable parts moving further away from each other naturally lead to an increase in the scale of the target object. Similarly, parts moving closer to each other indicate decreasing size of the target. We found that this scale estimation scheme is robust in that it provides a simple and efficient way to estimate the size of the target as shown in Figure 4.

Changing the size in every step without considering the previous scale information may lead inaccurate estimates. Hence, we change the original filter size $m \times n$ by a scale factor s $(ms \times ns)$ only if scaling shows a constant increase or decrease trend. In

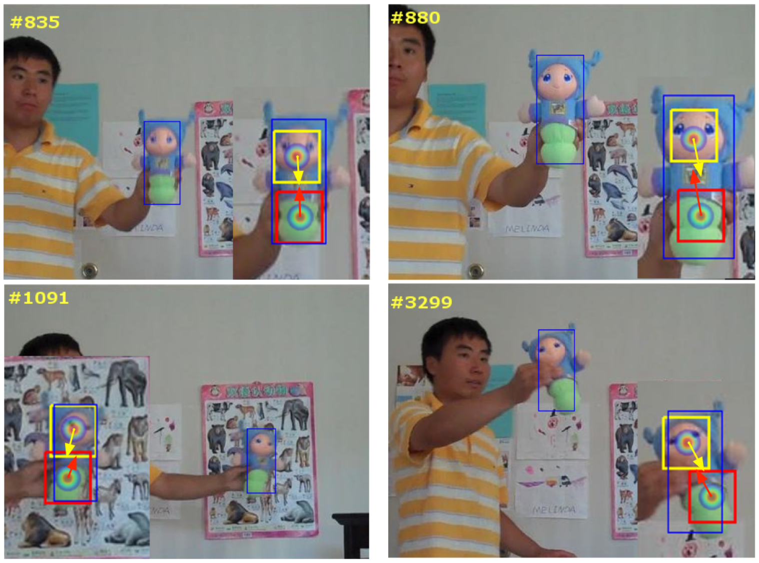

Figure 4: Tracking under large scale changes. In a particular way, when parts compress or stretch the scaling factor can be captured.

particular, if scaling is need to be updated, the system is re-initialized by changing the filter sizes according to the estimated scale factor. In this step, the previous information (template and filter) is forgotten but correlation filter strategy can handle the situation quite well. Specifically, we compute the change in the scale as follows:

$$s^t = \frac{\|\boldsymbol{\ell}_i^t - \boldsymbol{\ell}_j^t\|}{s^{t-1}} \; , \tag{16}$$

with i and j denoting the parts. We store all these scaling factors since the (re)-initialization and while changing the filter sizes, we use the mean of the scaling factors $\eta = avg(s^{0:t})$ rather than the scaling factor estimated at the current time step.

We use η to decide the scaling is necessary or not by comparing the scaling ratio with a threshold $1 - \delta$ or $1 + \delta$. Based on η, we have two options: update the sizes of the local and global filters or no update. If $\eta < 1 + \delta$ or $\eta > 1 - \delta$ and all parts are reliable, it indicates that the size of the target needs to be updated. Once the sizes are updated, we start recalculating the s from scratch. This simple scaling strategy, which takes into account not all but the latest upward or downward trend, is proved effective empirically as discussed in the experiments.

3.7. Implementation Details: Features and Parameters

Our approach is implemented in Matlab/C++ and all the algorithmic parameters are fixed throughout the experiments. Our parameters are similar to those in KCF [4] except our learning rate $\gamma = 0.18$. Since HoG representation performs poorly at low resolution (below 30x30 pixels), we use the cell size 2x2 rather than 4x4 for small targets. The filter and search size are set to 2.4 times.

Although theoretically part reliability changes between 0.5 and 1, we experimentally observed that reliability changes 0.5 and 0.75 and found that $\mathbf{r}_i$ dropping around 0.56 indicates heavy occlusion, abrupt motion, scaling or sudden pose change. We therefore set the threshold ψ in Eq. 12 - 15 to 0.55. Several latest algorithms [12, 26] use a pool of scale factors that covers 1% to 15% scale changes. Unlike these algorithms, we change the filter size if scaling is decided to be needed. This means that the more change the size, the more forget the previous information. Therefore, we set the scaling threshold δ is to 0.15.

The proposed approach results in a tracker processing videos with the speed of 20 frames per second. The correlation for individual parts takes up to 35% of computation time, combining the scores and updating the deformation scores takes nearly 50% and updating the parts models takes 15%.

4. Experiments

In this section, we first introduce the experimental setup, then evaluate the performance of our method by comparing it against different correlation based trackers as well as other state-of-the-art approaches.

4.1. Experimental Setup

To evaluate the robustness and accuracy of our approach, we carry out two comparative experiments. We first focus on six closely related correlation based trackers, and then compare to other five methods that recently produced state-of-the-art results on challenging benchmarks. The results are reported on two visual tracking benchmark datasets, OOTB [13] and TB-100 [13]. Both of these datasets contain 11 different challenges such as illumination variation (IV), scale variation (SV), occlusion (OCC), deformation (DEF), motion blur (MB), fast motion (FM), in-plane rotation (IPR), out-of-plane rotation (OPR), out-of-view (OV), background clutter (BC), and low resolution (LR). The results are also reported for these benchmark subcategories.

In all the experiments, the performance is measured with two evaluation criteria which are Center Location Error (CLE) and the Pascal VOC overlap ratio (OP). For the CLE, the pixel distance between the center of tracked bounding box and the ground truth is averaged over all frames and sequences, i.e., the lower the better. In addition, OP is used which is the average success rate for 0.5 overlap of the tracker and the ground truth bounding boxes. Given the tracking output bounding box r_t and the ground truth bounding box g_t, the overlap score $o = |r_t \cap g_t|/|r_t \cup g_t|$ is used to measure tracking success per frame, where $|\cdot|$ denotes the area.

It is worth noting that CLE has to be considered with caution, as the detected and the ground-truth bounding boxes vary in size. Furthermore, when a tracker drifts away from the target, its actual distance to the object has less importance. In addition, we present success rate and precision plots over a range of thresholds. The success plots for various methods result from [13] and represent the one pass evaluation scores while varying the overlap threshold from 0 to 1.

Both, precision and success show the average scores over all the sequences. In addition to quantitative results, we also illustrate our method strengths and weaknesses qualitatively.

4.2. Comparison to Correlation Based Trackers

To demonstrate the performance gain of our approach with the part based deformation model, we compare our tracker DPCF to the recent correlation filter based trackers that include CSK [9], PBTLD [10], KCF [4], ACT [11] and DSST [12] on the OOTB dataset. The common part of all of these trackers is the use of correlation filters or their kernelized versions.

It is apparent from the CLE and OP curves of Figure 5 that our DPCF tracker outperforms the other methods. The results also suggest that methods that explicitly

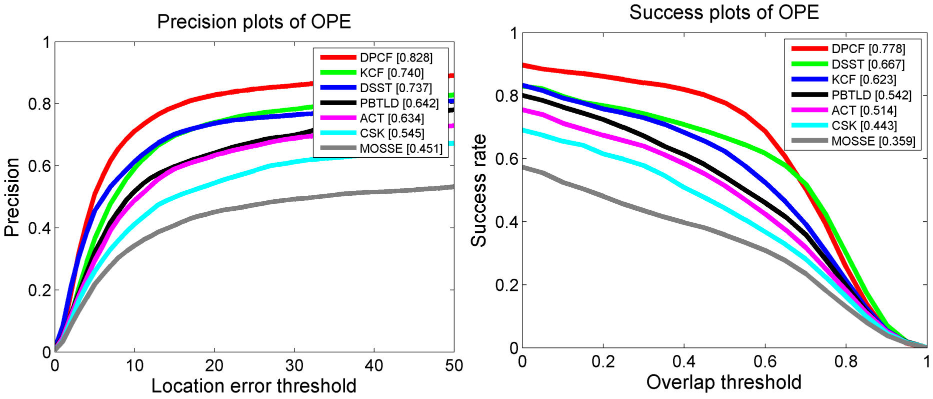

Figure 5: Comparison of the correlation filter based trackers in terms of localization error and overlap on the OOTB dataset. The legend shows the ranking of methods which is largely consistent for the two measures.

address the scale change i.e., DPCF, DSST problem have an advantage. The results also support the intuition that the part-based schema (DPCF, PBTLD) and the collaboration strategy play an important role in robust tracking. Specifically, DPCF compared with the second best DSST, OP and CLE performance gains of our method are %12 and 10 pixels error, respectively. The result also demonstrates that DPCF outperforms KCF nearly 16% for OP score and 9 pixels error for CLE score. The results in Table 1 for individual challenges also indicate that our tracker can better handle occlusion, scaling and deformation than other methods while accurately estimating the size of object.

The recent RTCF [27] tracker also employs a part-based strategy. The authors of [27], on the other hand, has performed experiments only on 16 videos from the benchmark [13] dataset. We have compared our results against RTCF and found that our approach gives nearly 6 % better than those of RTCF on this limited set. More recently, LCT is reported with the 76.9 % OP score. Our tracker also surpasses the LCT 1% score of the OP without any learning or re-detection strategy. It is noteworthy that LCT uses learning and re-detection which is proved by the authors of PBTLD [10] that learning and re-detection adds nearly 10% gain.

Moreover, we also compare our method with KCF and DSST on the TB-100 dataset. The results given in Table 2 also illustrate that our tracker outperforms both KCF and

Table 1: Tracking Benchmark Results on the OOTB dataset and its challenging subcategories for the top twelve tracking algorithms including 6 correlation filter based (*). Our method outperforms all the state-of-the-art algorithms according to overlap (OP) score (highest). Our algorithm has gained 8 the best and 2 the second best out of 11 subcategories for OP score. The column headers indicate the challenge and its number of sequences.

Tracker	All	IV-25	SV-28	OCC-29	DEF-19	MB-12	FM-17	IPR-31	OPR-39	OV-6	BC-21	LR-4
DPCF*	**77.8**	**74.7**	**68.2**	**79.9**	**82.2**	**66.5**	*63.7*	**69.7**	**69.7**	*67.7*	**75.6**	46.0
MEEM [18]	*68.5*	61.9	57.9	*65.9*	64.6	60.3	**67.4**	60.9	*66.3*	**76.7**	*73.9*	*46.3*
DSST* [12]	66.7	*68.6*	*63.0*	64.4	63.4	53.1	50.8	*67.4*	64.0	52.1	61.3	**49.7**
TGPR [17]	64.5	58.4	50.1	58.8	67.9	58.0	56.1	59.3	60.7	54.4	70.7	45.4
KCF*[4]	62.8	59.5	48.6	63.0	*69.4*	*61.6*	55.8	62.3	61.6	66.1	66.3	35.6
SCM [19]	61.6	56.8	*63.5*	59.9	56.5	33.9	33.5	56.0	57.5	44.9	55.0	30.8
PBTLD* [10]	57.8	50.7	45.8	56.3	55.9	55.0	52.5	52.9	54.7	49.3	52.4	41.4
Struck [2]	56.0	49.0	47.6	49.3	48.8	53.7	55.4	53.4	50.7	48.8	54.3	41.0
TLD [3]	52.1	46.0	49.4	46.8	45.6	48.2	47.3	47.6	49.7	51.6	38.8	32.7
ACT* [11]	51.4	46.8	42.7	49.6	55.3	49.5	43.2	55.6	51.6	47.0	53.5	39.7
ASLA [20]	51.1	50.3	54.4	45.1	45.6	28.1	26.0	48.8	49.4	35.9	46.8	16.3
CSK* [9]	45.3	41.1	37.1	42.3	39.9	38.4	38.2	47.5	45.3	41.1	49.3	39.6
LGT [24]	37.5	32.8	29.0	33.7	29.7	21.3	32.5	38.2	37.4	48.2	36.0	15.7

DSST, giving 15% and 9% better OP scores.

4.3. Comparison to State-of-the-art Trackers

In our next experiment, we have compared our approach with 36 different state-of-the-art trackers on the OOTB dataset. In addition to the 29 methods reported in the benchmark experiment in [13] we include KCF [4], TGPR [17], ACT [11], PBTLD [10], DSST [12], MEEM[18] and LGT[34].

Table 1 presents the overall scores in column (All) OP as well as individual scores for 11 challenges on OOTB from [13]. Figure 6 shows detailed results for varying overlap scores. In Table 1, the best and the second best performing methods are given in bold and italic typefaces, respectively. DPCF significantly improves upon all the trackers with the average overlap score of 77.8%. We also tested our method without the proposed deformation model, for which the score dropped by 6.5% which shows that our deformation model leads to a significant gain. Moreover, our method obtained 8 the best and 2 the second best score out of 11 subcategories according to OP score.

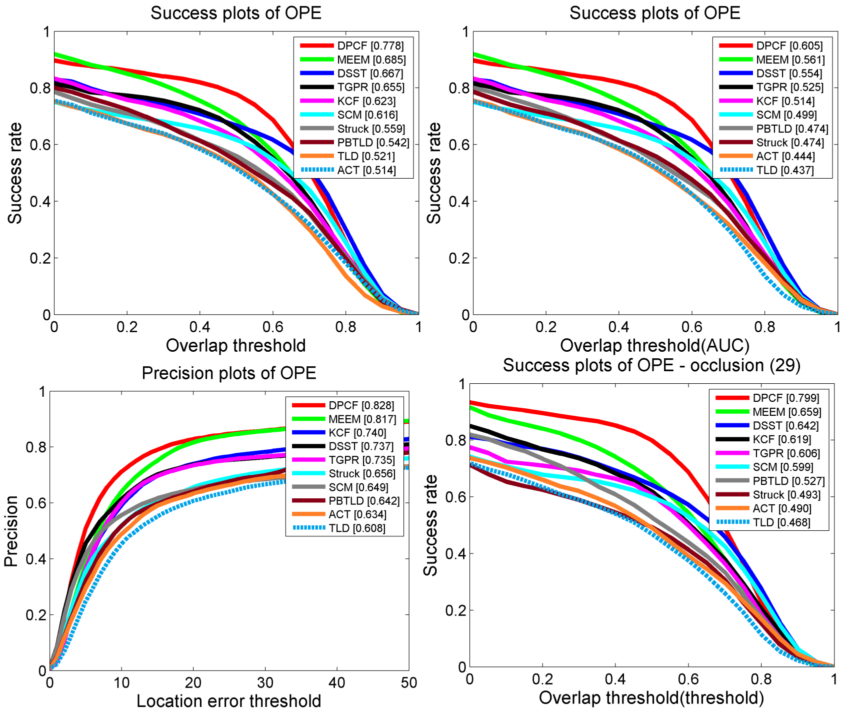

Figure 6: Quantitative analysis of the proposed tracker against the top 9 performing state-of-art trackers on the OOTB dataset. The plots are generated for OPE (threshold 0.5), OPE (AUC), precision plot and Occlusion OPE (threshold 0.5). In all plots, our DPCF algorithm gives the highest success rate as listed in the legend.

Struck [2] is reported to have shown the best performance in a recent evaluation [13], as well as KCF [4], TGPR [17] and MEEM [18] have outperformed other existing methods. Furthermore, DSST is reported to have surpassed all trackers according to equally challenging VOT2014 [35] benchmark results. Our experiments with the sequences and protocols from [13] show that our DPCF tracker outperforms Struck, KCF, TGPR, DSST and MEEM nearly 20%, 16%, 14%, 12% and 10% respectively, while keeping the real time frame rate performance. DPCF ranks the first with nearly 15%, 18% and 5% margin of OP score in 29 videos with occlusions, 19 videos deformation and 28 videos scale variation respectively. Moreover, we compared our algorithm with LGT [24] that has a similar coupled framework to ours. The results suggest that our algorithm nearly doubles the score of LGT on the OOTB dataset.

While LGT is vulnerable to occlusion, clutter background, scaling, etc., our tracker can handle the situations very well, and importantly in real time. In Figure 7, we provide sample results of the top five best performing methods. Additional results for visual tracking can be found in the supplementary video. These qualitative results also illustrate the effectiveness of the proposed approach.

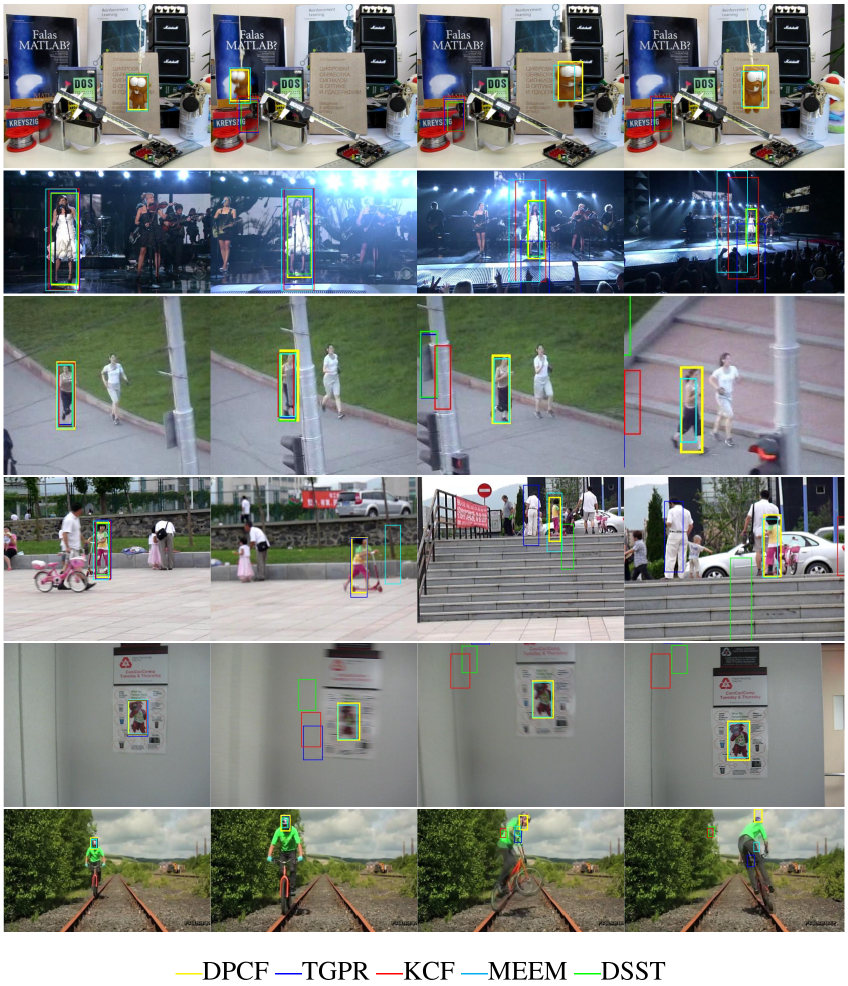

Figure 7: DPCF tracker compared with the top 5 methods that include TGPR [17], MEEM[18] , DSST [12] and KCF [4] showed in challenging frames i.e., occlusions, deformation, illumination variations and scaling. The sample sequences are from the *Lemming*, *Singer1*, *Jogging-1*, *Girl2*, *BlurOwl* and *Biker* [33] respectively. Our method handles occlusion and scale changes more accurately than the other trackers.

Table 2: Tracking benchmark results of TB-100 and challenge subcategories for top five tracking algorithms including 3 correlation filter based (*). Our method outperforms all the state-of-the-art algorithms according to overlap (OP) score (highest). Our algorithm has gained 10 best out of 11 subcategories for OP. The column headers indicate the challenge and its number of sequences.

Tracker	All	IV-37	SV-64	OCC-49	DEF-44	MB-28	FM-38	IPR-52	OPR-63	OV-14	BC-31	LR-9
DPCF*	**69.5**	**74.1**	**61.1**	**67.5**	**63.5**	**70.4**	**62.2**	**64.5**	**66.7**	*57.8*	**72.9**	**47.7**
MEEM[18]	*61.5*	58.5	49.8	*58.3*	51.2	*65.7*	*61.2*	*60.7*	*59.6*	**59.5**	*66.8*	36.8
DSST*[12]	61.3	*66.5*	*55.0*	54.5	51.4	57.0	52.6	59.7	56.9	44.3	61.1	44.4
TGPR[17]	56.2	53.1	44.5	53.3	*55.6*	56.5	53.5	57.3	56.0	43.9	55.1	*46.5*
KCF*[4]	55.1	54.7	41.6	51.3	50.3	56.7	51.4	54.3	52.7	45.7	60.9	29.5

In addition to the experiments on the OOTB dataset, we have also compared our method against the top 4 trackers on the TB-100 benchmark dataset [33]. We provide these results in Table 2. Figure 8 shows detailed results for varying overlap scores. The OP and AUC scores indicate that our tracker has an advantage on the other methods both overall and subcategories score. Our tracker is positioned first rank for overall with the 8% gain and 10 best score out 11 subcategories. These margins are 12% for deformation, 10% for occlusion, 8% for illumination variation and 6% for scale variation.

While most of the trackers suffer from significant occlusions (i.e *lemming* 345th, *girl* 434th, *coke* 257th, *skating* 85th and *jogging* 70th frames in Figure 9), our tracker can handle these cases successfully. In addition to better handling of occlusion and deformation cases, one of the main advantages of our method is more accurate estimation of the size of object which is reflected in the overlap (OP) scores in Figure 10.

Finally, the results obtained for the 28 scale varying sequences in the OOTB dataset show that our tracker outperforms ASLA by 14% as well as SCM and DSST by 5%. This margin is 6% for the 44 scaling varying sequences in the TB-100 dataset when our method is compared with DSST. It is worth to mention that these algorithms were specifically designed to deal with the scale change problem. In contrast, our approach estimates the size of object by measuring the shrinking or stretching amongst parts. This margin is nearly 10% when considering the 49 videos with occluded scenarios for TB-100 (cf. Figure 8). In addition, our method significantly boosts the score (i.e., by nearly 8%) with respect to the second best model, TGPR, for the 44 sequences in the

TB-100 dataset which include deformation.

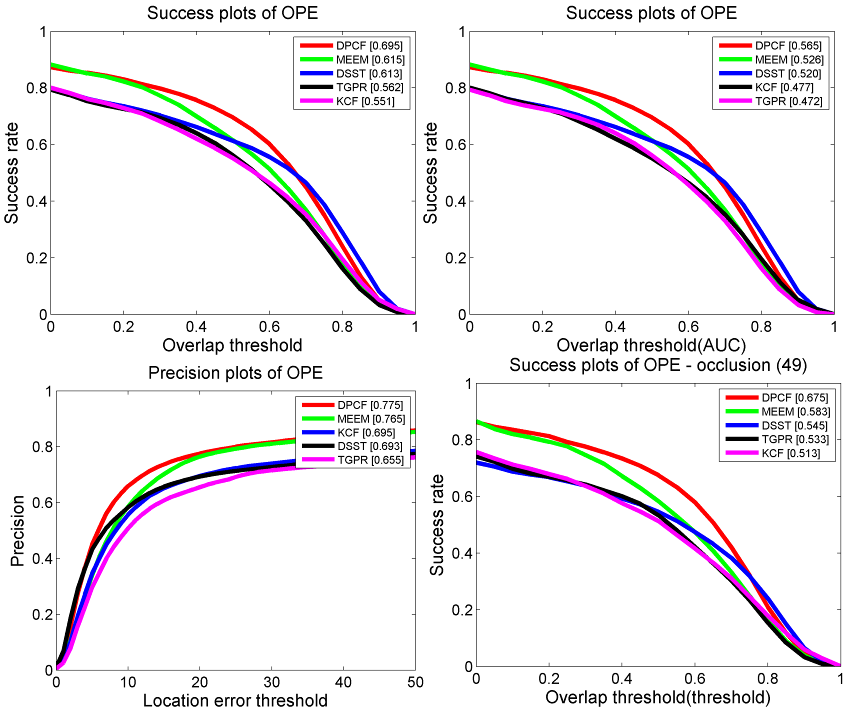

Figure 8: The success rate plots of the proposed tracker and the other top 5 best performing trackers on the TB-100 dataset. The plots are generated for OPE (threshold 0.5), OPE (AUC), OPE (presicion plot) and Occlusion OPE (threshold 0.5). In all plots our DPCF algorithm gives the highest success rate as listed in the legend.

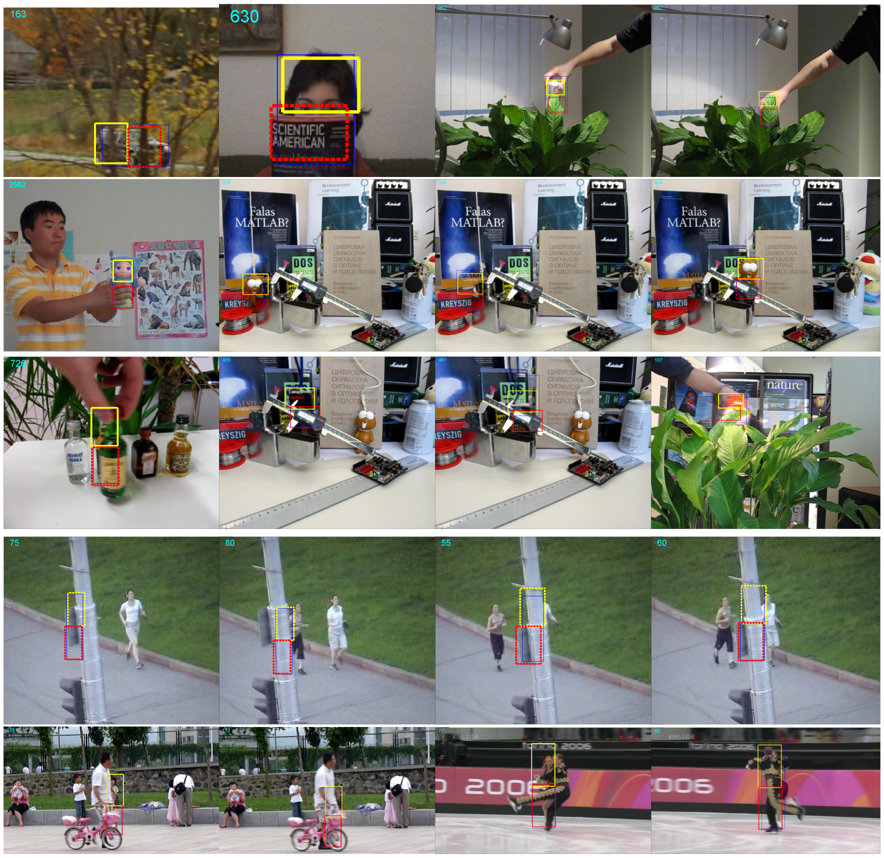

Figure 9: The proposed tracker can detect partial or heavy occlusions using the part structures. Occluded part or parts have shown with dashed lines. The sample results are from the *carscale*, *faceocc1*, *coke*, *doll*, *lemming*, *liquor*, *box*, *tiger2*, *jogging-1*, *jogging-2*, *girl2*, and *Skating-1* [33] sequences, respectively.

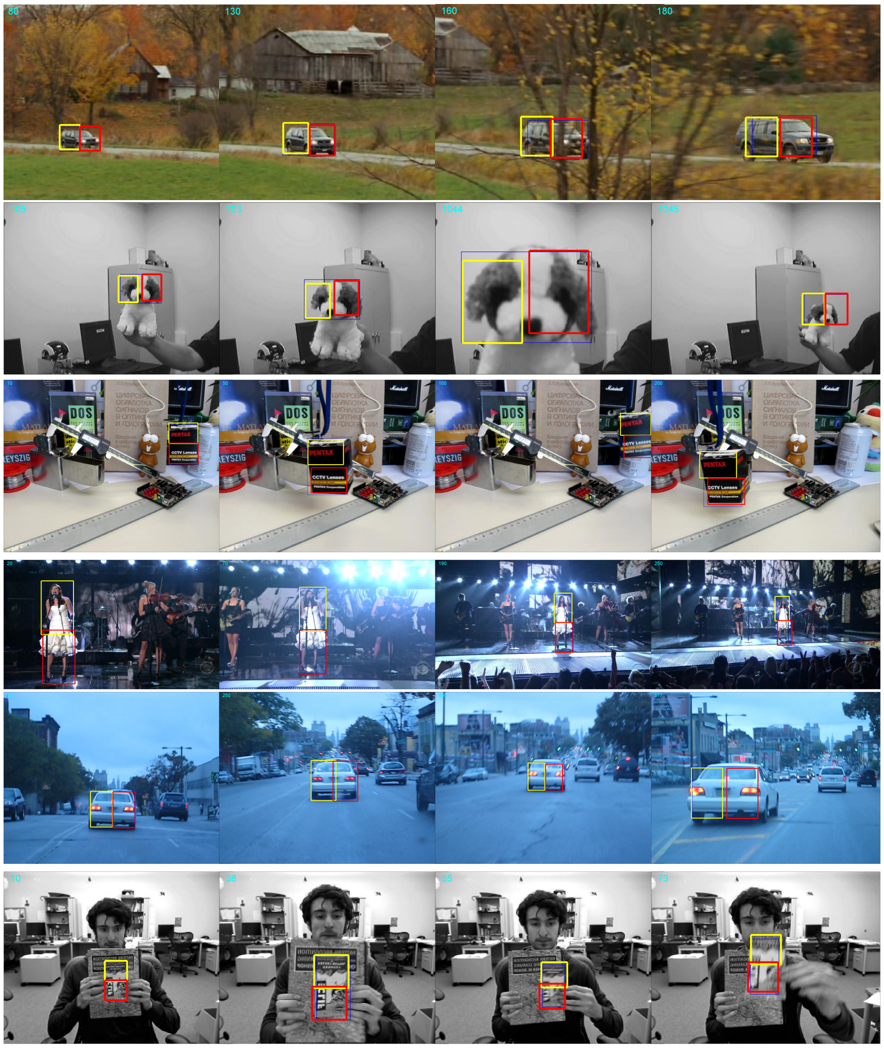

Figure 10: Shrinking or stretching of the object parts allows the tracker to cope with the changes in the scale. The sample results are from the *carScale*, *dog1*, *box*, *singer1*, *blurCar2* and *clifBar* [33] sequences, respectively.

4.4. Discussion

Our tracking algorithm combines different visual features within a collaborative deformable part-based model. In this section, we analyze the benefits of each feature and each component to the overall performance. In Table 3, we report the performances of both the proposed DPCF tracker and the baseline KCF tracker with different feature combinations. While color feature improved the score of KCF only 0.5%, our tracker performance is improved nearly 3%. This simply illustrates that our tracker performance does not mainly come from the feature that we used but the proposed coupled deformable model. Secondly, we additionally analyze each component of the proposed algorithm on the TB-100 dataset. The results in Table 4 show that our coupled model leads to a nearly 10% score gain. On the other hand, our adaptive scaling component contributes nearly 1.5% to the overall performance. The advantage of this scaling strategy is more visible for the results obtained on the 64 sequences which contains scale variations (SV-64).

Table 3: Tracking results of the proposed DPCF algorithm and KCF on the TB-100 dataset with different feature combinations.

DPCF (HoG+Color)	DPCF (HoG)	KCF (HoG+Color)	KCF (HoG)
69.5	*65.7*	55.6	55.1

Table 4: Analysis of the components of the proposed tracking algorithm on the sequences from TB-100 and SV-64 (64 scale variation sequences in the TB-100 dataset).

	DPCF	DPCF (No Scale)	DPCF (No Coupled)
TB-100	**69.5**	*68.2*	59.3
SV-64	**61.1**	*58.3*	51.1

5. Conclusions

We have introduced an effective tracker based on both correlation filters and a deformation part-based model. Our collaborative local-global model which is built up on a deformable part-based tracker improves the performance of the existing results especially under significant occlusions. In addition, our scale estimation scheme that considers the part displacements successfully addresses the problem of scale change in videos. Moreover, the results for the deformation problem and other challenges indicate that the proposed approach is more robust to such variations compared to other state-of-the-art trackers.

The overall evaluations results on the challenging benchmark datasets showed significant improvement in both accuracy and robustness while the video processing cost is still within real time range of 20fps. Our tracker outperformed all the state-of-the-art trackers, resulting in nearly 12% score gain on the OOTB dataset and 8% on the TB-100 dataset.

References

[1] B. Babenko, M.-H. Yang, S. Belongie, Visual tracking with online multiple instance learning, in: CVPR, 2009, pp. 983–990.

[2] S. Hare, A. Saffari, P. H. Torr, Struck: Structured output tracking with kernels, in: ICCV, 2011, pp. 263–270.

[3] Z. Kalal, K. Mikolajczyk, J. Matas, Tracking-learning-detection, IEEE T-PAMI 34 (7) (2012) 1409–1422.

[4] J. F. Henriques, R. Caseiro, P. Martins, J. Batista, High-speed tracking with kernelized correlation filters, IEEE T-PAMI 37 (3) (2015) 583–596.

[5] F. Porikli, O. Tuzel, P. Meer, Covariance tracking using model update based on lie algebra, in: CVPR, Vol. 1, 2006, pp. 728–735.

[6] J. Santner, C. Leistner, A. Saffari, T. Pock, H. Bischof, Prost: Parallel robust online simple tracking, in: CVPR, 2010, pp. 723–730.

[7] H. Grabner, M. Grabner, H. Bischof, Real-time tracking via on-line boosting., in: BMVC, Vol. 1, 2006, p. 6.

[8] D. S. Bolme, J. R. Beveridge, B. A. Draper, Y. M. Lui, Visual object tracking using adaptive correlation filters, in: CVPR, 2010, pp. 2544–2550.

[9] J. F. Henriques, R. Caseiro, P. Martins, J. Batista, Exploiting the circulant structure of tracking-by-detection with kernels, in: ECCV, 2012, pp. 702–715.

[10] O. Akin, K. Mikolajczyk, Online learning and detection with part-based, circulant structure, in: ICPR, 2014, pp. 4229–4233.

[11] M. Danelljan, F. Khan, M. Felsberg, J. Weijer, Adaptive color attributes for real-time visual tracking, in: CVPR, 2014, pp. 1090–1097.

[12] M. Danelljan, G. Häger, F. Khan, M. Felsberg, Accurate scale estimation for robust visual tracking, in: BMVC, 2014.

[13] Y. Wu, J. Lim, M.-H. Yang, Online object tracking: A benchmark, in: CVPR, 2013, pp. 2411–2418.

[14] X. Li, W. Hu, C. Shen, Z. Zhang, A. Dick, A. V. D. Hengel, A survey of appearance models in visual object tracking, Intelligent Systems and Technology (TIST) 4 (4) (2013) 58.

[15] H. Yang, L. Shao, F. Zheng, L. Wang, Z. Song, Recent advances and trends in visual tracking: A review, Neurocomputing 74 (18) (2011) 3823–3831.

[16] M. Kristan, J. Matas, A. Leonardis, T. Vojir, R. Pflugfelder, G. Fernandez, G. Nebehay, F. Porikli, L. Cehovin, A novel performance evaluation methodology for single-target trackers, IEEE T-PAMI PP (99) (2016) 1.

[17] J. Gao, H. Ling, W. Hu, J. Xing, Transfer learning based visual tracking with gaussian processes regression, in: ECCV, 2014, pp. 188–203.

[18] J. Zhang, S. Ma, S. Sclaroff, Meem: Robust tracking via multiple experts using entropy minimization, in: Computer Vision–ECCV, 2014, pp. 188–203.

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

[20] X. Jia, H. Lu, M. H. Yang, Visual tracking via adaptive structural local sparse appearance model, in: CVPR, 2012, pp. 1822–1829.

[21] Y. Lu, T. Wu, S. Zhu, Online object tracking, learning and parsing with and-or graphs, in: CVPR, 2014, pp. 3462–3469.

[22] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, D. Ramanan, Object detection with discriminatively trained part-based models, IEEE T-PAMI 32 (9) (2010) 1627–1645.

[23] L. Zhang, L. Maaten, Structure preserving object tracking, in: CVPR, 2013, pp. 1838–1845.

[24] L. Čehovin, M. Kristan, A. Leonardis, An adaptive coupled-layer visual model for robust visual tracking, in: ICCV, 2011, pp. 1363–1370.

[25] Z. Chen, Z. Hong, D. Tao, An experimental survey on correlation filter-based tracking, preprint arXiv:1509.05520.

[26] Y. Li, J. Zhu, A scale adaptive kernel correlation filter tracker with feature integration, in: ECCV Workshops, 2014, pp. 254–265.

[27] T. Liu, G. Wang, Q. Yang, Real-time part-based visual tracking via adaptive correlation filters, in: CVPR, 2015, pp. 4902–4912.

[28] C. Ma, X. Yang, C. Zhang, M.-H. Yang, Long-term correlation tracking, in: CVPR, 2015, pp. 5388–5396.

[29] G. Zhu, J. Wang, Y. Wu, H. Lu, Collaborative correlation tracking, in: BMVC, 2015.

[30] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: CVPR, 2005, pp. 886–893.

[31] S. S. Nejhum, J. Ho, M.-H. Yang, Visual tracking with histograms and articulating blocks, in: CVPR, 2008, pp. 1–8.

[32] A. Adam, E. Rivlin, I. Shimshoni, Robust fragments-based tracking using the integral histogram, in: CVPR, 2006, pp. 798–805.

[33] Y. Wu, J. Lim, M.-H. Yang, Object tracking benchmark, IEEE T-PAMI 37 (9) (2015) 1834–1848.

[34] L. Cehovin, M. Kristan, A. Leonardis, Robust visual tracking using an adaptive coupled-layer visual model, IEEE T-PAMI 35 (4) (2013) 941–953.

[35] M. Kristan, R. P. Pflugfelder, A. Leonardis, J. Matas, L. Cehovin, G. Nebehay, T. Vojir, G. Fernandez, A. Lukezi, A. Dimitriev, A. Petrosino, A. Saffari, B. Li, B. Han, C. Heng, C. Garcia, D. Pangersic, G. Hger, F. S. Khan, F. Oven, H. Possegger, H. Bischof, H. Nam, J. Zhu, J. Li, J. Y. Choi, J. Choi, J. F. Henriques, J. van de Weijer, J. Batista, K. Lebeda, K. Ofjall, K. M. Yi, L. Qin, L. Wen, M. E. Maresca, M. Danelljan, M. Felsberg, M. Cheng, P. Torr, Q. Huang, R. Bowden, S. Hare, S. Y. Lim, S. Hong, S. Liao, S. Hadfield, S. Z. Li, S. Duffner, S. Golodetz, T. Mauthner, V. Vineet, W. Lin, Y. Li, Y. Qi, Z. Lei, Z. Niu., The visual object tracking vot2014 challenge results, in: ECCV Workshops, 2014, p. 191217.