Machine Vision and Applications
https://doi.org/10.1007/s00138-019-01009-9

ORIGINAL PAPER

Beyond feature integration: a coarse-to-fine framework for cascade correlation tracking

Dongdong Li[1] · Gongjian Wen[1] · Yangliu Kuai[1] · Fatih Porikli[2]

Received: 4 May 2018 / Revised: 11 October 2018 / Accepted: 13 January 2019

Abstract

Discriminative correlation filters (DCF) have achieved enormous popularity in the tracking community. Recently, the performance advancement in DCF-based trackers is predominantly driven by the use of convolutional features. In pursuit of extreme tracking performance, state-of-the-art trackers (e.g., cascade correlation tracking [1] and HCF [2]) equip DCF with hierarchical convolutional features to capture both semantics and spatial details of the target appearance. While such methods have been shown to work well, multiple feature integration results in high model complexity which significantly increases the over-fitting risk and computational burden. In this paper, we present a coarse-to-fine framework for cascade correlation tracking (CCT). Instead of integrating hierarchical features, this framework decomposes a complicated tracker into two low-complexity modules, a coarse tracker $\mathcal{C}$ and a refined tracker $\mathcal{R}$, working in a coarse-to-fine manner. The coarse tracker $\mathcal{C}$ employs low-resolution semantic convolutional features extracted from a large search area to cope with large target displacement and appearance change between adjacent frames. By contrast, the refined tracker $\mathcal{R}$ employs high-resolution handcraft features extracted from a small search area to further refine the coarse location of $\mathcal{C}$. Our CCT tracker enjoys the strong discriminative power of $\mathcal{C}$ and the high efficiency of $\mathcal{R}$. Experiments on the OTB2013 and TC128 benchmarks show that CCT performs favorably against state-of-the-art trackers.

Keywords Correlation filter · Coarse to fine · Visual tracking

1 Introduction

Visual tracking is a classical and rapidly evolving research topic in computer vision with a variety of applications such as video surveillance [3], augmented reality [4] and human–computer interaction [5]. It is the task of continuously locating a target given only its initial state (generally an axis-aligned rectangle) in a video sequence. Recently, discriminative correlation filters (DCF)-based trackers have achieved enormous popularity in the tracking community. With the circular assumption, standard DCF achieves extremely high tracking frame rates with a closed-form element-wise solution. Different variants of closed-form DCF have been proposed to boost tracking performance using multi-dimensional features [6], robust scale estimation [7], nonlinear kernels [8], long-term memory components [9], target response adaptation [10], complementary cues [11] and context learning [12].

To achieve further performance improvement, an emerging trend is to use deep features for their strong discriminative power. Danelljan et al. [13] first introduce shallow convolutional features into the DCF-based tracking framework. Shallow convolutional features capture high-resolution spatial details for precise localization, but are not robust to target appearance variation. Later, Ma et al. [2] exploit the semantic information of last layers to handle large appearance changes and alleviate drifting by using features of earlier layers for precise localization. CCOT [1] employs the integration of multi-resolution features in the continuous domain and achieves the top rank on the VOT2016 challenge [14]. Compared with traditional closed-form DCF which employs handcraft features and a restricted search area, both HCF [2] and CCOT [1] maintain a larger filter size or filter dimension due to the larger search area or multiple feature integration. The large filter size and filter dimension lead to massive

✉ Dongdong Li
moqimubai@sina.cn

[1] College of Electronic Science, National University of Defense Technology, Changsha, China

[2] Research School of Engineering, Australian National University, Canberra, Australia

Published online: 13 February 2019

trainable parameters in the tracking model, which increases the model complexity, over-fitting risk and computational burden. For instance, CCOT runs with a frame rate of 0.2 fps, which is insufficient for computationally constrained platforms, such as aerial tracking using unmanned aerial vehicle (UAV).

In this paper, instead of integrating hierarchical features, we propose to reduce the model complexity by decomposing a complicated tracker into low-complexity collaborative modules. This idea is inspired by the following observations:

Observation 1. Closed-form DCF with handcraft features are suitable for accurate and efficient tracking, but suffers from large target displacement between adjacent frames due to the restricted search area originated from boundary effect. Moreover, handcraft features are not robust to target appearance variation.

Observation 2. Semantic convolutional features extracted from the last layers of a deep neural network is suitable for robust coarse tracking, which can compensate closed-form DCF for large target displacement. Meanwhile, complementary to handcraft features, semantic convolutional features capture abstract semantics which cope well with large appearance variation.

Observation 3. Closed-form DCF employs high-resolution handcraft features, but can be efficiently trained with an element-wise solution. Semantic convolutional features contain limited values due to the small feature map size and put little burden on model training.

Based on the above observations, we propose a coarse-to-fine framework for cascade correlation tracking. Our CCT consists of two components, a coarse tracker $\mathcal{C}$ and a refined tracker $\mathcal{R}$, working in a coarse-to-fine manner. The coarse tracker $\mathcal{C}$ employs low-resolution semantic convolutional features extracted from a large search area to cope with large target displacement and appearance variation. On contrast, the refined tracker $\mathcal{R}$ employs high-resolution handcraft features extracted from a small search area to locate the target accurately with a closed-form DCF solution.

Implementing a CCT algorithm needs three parts: a coarse tracker $\mathcal{C}$, a refined tracker $\mathcal{R}$ and a scale module $\mathcal{S}$. For $\mathcal{C}$, we choose the CREST tracker [15] to track with semantic convolutional features extracted from a large search area. For $\mathcal{R}$ and $\mathcal{S}$, we choose the *discriminative scale space tracking* (DSST) algorithm [16] for efficient precise target localization and scale estimation.

The proposed CCT algorithm is evaluated thoroughly on two popular tracking benchmarks OTB2013 [17] and TC128 [18]. In these experiments, CCT achieves favorable tracking performance against state-of-the-art trackers.

In summary, our first main contribution is the novel coarse-to-fine tracking framework to decompose a complicated tracker into basic collaborative modules. With this framework, we made a second contribution to implement a tracking solution that combines closed-form DCF (DSST) with a deep learning-based tracker (CREST). Then, our solution shows very promising results on OTB2013 and TC128 in comparison with state of the arts. Moreover, it is worth noting that CCF is a very flexible framework and our implementation is far from optimal. We believe there are great rooms for future improvement and generation.

2 Related works

There are extensive surveys on visual tracking in the literature. We refer readers to [19] and [14] for a thorough review of existing tracking algorithms. In this section, we only focus on the most related work.

Closed-form correlation filters. Recently, discriminative correlation filters (DCF) have drawn increasing attention in visual tracking. Conventional correlation filters transform spatial correlation into efficient element-wise multiplication in the frequency domain and achieve extremely high computational efficiency with a closed-form solution. The pioneer MOSSE tracker [20] attracted considerable attention with a tracking speed of over 600 fps. Henriques et al. [21] introduce kernel space into correlation filter and propose a circulant structure with kernel (CSK) method for tracking. CSK is then extended in [8] for further improvement and results in the well-known kernelized correlation filters (KCF). Later, discriminative scale space tracker (DSST) [7] is proposed by Danelljan et al. to achieve real-time scale adaptive tracking. Bertinetto et al. [11] combine a correlation filter and a global color histogram to achieve robustness to both deformation and color change. Recently, Valmadre et al. [22] interpret closed-form DCF as a differential layer in a deep neural network to learn end-to-end convolutional features.

Conceptual improvement in filter learning. Despite the extreme efficiency, closed-form DCF significantly suffers from boundary effects which lead to a restricted search area. Several approaches have been proposed to address the problem of boundary effects. SRDCF [23] learn a correlation filter with large spatial support which leads to a larger search area in the detection stage. Filter values outside the object bounding box are penalized with higher regularization weights to highlight the central area of the correlation filter. Within the SRDCF framework, CCOT [1] employs the integration of multi-resolution features in the continuous domain and achieves the top rank on the VOT2016 challenge [14]. BACF [24] trains correlation filters from real negative samples densely extracted from the background and ensures a correct filter size. Further, Song et al. [15] reformulate the correlation filter as a convolutional kernel in a deep neural network and propose convolutional residual learning for visual tracking (CREST). Both CCOT and CREST exploit the high-dimensional shallow convolutional features and maintain massive trainable parameters in the tracking model.

Cooperative mechanism in visual tracking. The idea of decomposing a complicated tracker into basic cooperative modules is not new in visual tracking. A pioneering example is the TLD tracker [25] which consists of a tracker, a detector and a learner. The long-term correlation tracker (LCT) [9] combines closed-form DCF with a re-detection module to achieve robust long-term tracking. The parallel tracking and verifying (PTAV) framework consists of a tracker and a verifier, working in parallel on two separate threads to achieve accurate and real-time tracking. The verifier in PTAV is never updated and is activated only occasionally based on a fixed threshold. This manually designed activation mechanism is highly video dependent and suffers from poor generalization.

3 Building blocks

A typical coarse-to-fine tracking framework consists of two basic components: a coarse tracker $\mathcal{C}$ and a refined tracker $\mathcal{R}$.

3.1 Refined tracker $\mathcal{R}$

The refined tracker $\mathcal{R}$ is responsible for precise target localization. We choose the discriminative scale space tracker (DSST) to implement the refined tracker. The aim of DSST is to learn a multi-dimensional $M \times N \times d$ correlation filter R from a d-dimensional $M \times N \times d$ feature f. We denote feature layer $l \in \{1, \ldots, d\}$ of f by f^l. y is the designed output for each location in the feature f, which is a predefined sampled Gaussian with a standard deviation proportional to the target size. The desired correlation filter R is obtained by minimizing the following target function,

$$\varepsilon(R) = \left\| \sum_{l=1}^{d} f^l * R^l - y \right\|^2 + \lambda \sum_{l=1}^{d} \left\| R^l \right\|^2 . \tag{1}$$

Here, $*$ denotes the convolution operator and the regularization scalar λ controls the impact of the regularization term.

Equation (1) can be transformed into the Fourier domain as:

$$\varepsilon(\hat{R}) = \left\| \sum_{l=1}^{d} \hat{f}^l \cdot \hat{R}^l - \hat{y} \right\|^2 + \lambda \sum_{l=1}^{d} \left\| \hat{R}^l \right\|^2 . \tag{2}$$

Here, $\cdot$ denotes point-wise multiplication and the hat denotes the DFT of a function.

According to [7], the solution to (2) is

$$\hat{R}^l = \frac{\hat{f}^{l*} \cdot \hat{y}^l}{\sum_{l=1}^{d} \hat{f}^{l*} \cdot \hat{f}^l + \lambda}. \tag{3}$$

Here, $\hat{f}^l$ means the Fourier transform of f^l and $\hat{f}^{l*}$ means the complex conjugation of $\hat{f}^l$. The product and division in (3) is point-wise.

Let f denote the $M \times N \times d$ feature extracted in the current frame and R denote the $M \times N \times d$ correlation filter learned in the previous frame. In the detection stage, the correlation scores S_f at all locations in the image patch are computed as follows,

$$S_\text{f} = \mathscr{F}^{-1} \left\{ \sum_{l=1}^{d} \hat{f}^l \cdot \hat{R}^l \right\} . \tag{4}$$

Here, $\mathscr{F}$ denotes the Fourier transform of a function and its inverse denotes the inverse Fourier transform.

3.2 Coarse tracker $\mathcal{C}$

We choose the CREST tracker [15] as the coarse tracker $\mathcal{C}$ in CCT. Different from DSST which has a closed-form solution, CREST reformulates the correlation filter as a convolutional kernel in a one-layer deep neural network and performs statistical gradient to train the tracking model. Once we have the convolutional kernel trained, target localization is simply finding the maxima on the response map

$$y_x = \phi(x) * C \tag{5}$$

where C is the convolutional kernel, $*$ is the convolution operation and ϕ is a feature extractor, e.g., CNN. x is an image patch centered on the target which is usually larger than the target object to provide enough background context.

Different from (1), the objective of CREST is defined as follows

$$\varepsilon(C) = \sum_{(h,w) \in P} \left\| e^{y(h,w)} [y_x(h,w) - y(h,w)] \right\|^2 + \lambda \left\| C \right\|^2 . \tag{6}$$

where $y_x(h, w)$ represents the element of y_x in (h, w) coordinates and $P = \{(h, w) | |y_x(h, w) - y(h, w)| > 0.1\}$ is a set of coordinates where the difference of $y_x(h, w)$ and y_x is above a fixed threshold. $e^{y(h,w)}$ performs as active weight for P in hard negative mining.

Besides reformulating the correlation filter as a convolutional kernel, CREST also inserts spatial–temporal residual modules to avoid target model degradation by large appearance changes. CREST also devises sophisticated initialization, learning rates, and weight decay regularization.

4 Our tracking framework

Our tracking pipeline includes four steps: coarse location, refined location, scale estimation and model update. The flowchart of our coarse-to-fine tracking pipeline is shown in Fig. 1 and summarized in Algorithm 1. The details are discussed below.

4.1 Coarse-grained location

For the coarse tracker $\mathcal{C}$, we adopt imagenet-vgg-verydeep-16 network [26] using the implementation in the MatConvNet library [27] for feature extraction. The network is trained on the ImageNet dataset for the image classification task.

Given an input frame and the target location, we extract a large search patch (five times the target size) centered on the target object. This patch is fed into imagenet-vgg-verydeep-16 for feature extraction. To produce semantic convolutional features for the coarse tracker, we employ the activations produced after the relu4_3 layer. The semantic convolutional feature map has small spatial size, but captures coarse-grained semantics. To further reduce the model complexity of $\mathcal{C}$, the high-dimensional semantic convolutional features are compressed with the PCA dimensionality reduction.

Different from closed-form correlation filters which suffer from the restricted search region, our coarse tracker holds a large search region due to the large input image patch. The large search region enables the coarse tracker to cope better with fast motion and heavy occlusion. Besides taking advantage of the high-level semantics, the coarse tracker is robust to significant target appearance variation originated from self-deformation, illumination change and background clutter.

4.2 Fine-grained location

Due to the spatial strides in the convolutional neural networks, the coarse tracker $\mathcal{C}$ can only locate the target coarsely with a 8×8 cell size. Therefore, semantic convolutional features are insufficient to capture fine-grained spatial details which are effective for precise localization. This is attributed to the decreased spatial resolution in the deeper layers. Intuitively, better spatial resolution alleviates the task of accurately locating the target, which is crucial for the tracking problem.

The refined tracker $\mathcal{R}$ is responsible for precise target localization. Compared with the coarse tracker $\mathcal{C}$, $\mathcal{R}$ employs handcraft features (e.g., HOG [28]) from a smaller target search area (two times the target size) centered at the coarse location of $\mathcal{C}$. Due to the boundary effects, $\mathcal{R}$ is not able to cope with large displacement of the target between the previous frame and current frame. However, thanks to $\mathcal{C}$, most of the large displacement has been compensated by the coarse-grained location. Therefore, $\mathcal{R}$ only need to perform a second-round fine-grained location around the initial coarse-grained location. In this work, $\mathcal{R}$ employs the HOG feature with a 1×1 cell size. In this way, with the fine-grained location, the location accuracy is refined from 8 cell size to 1 cell size.

4.3 Model update

In our tracking pipeline, it is computationally efficient to extract the semantic convolutional features with the labor of

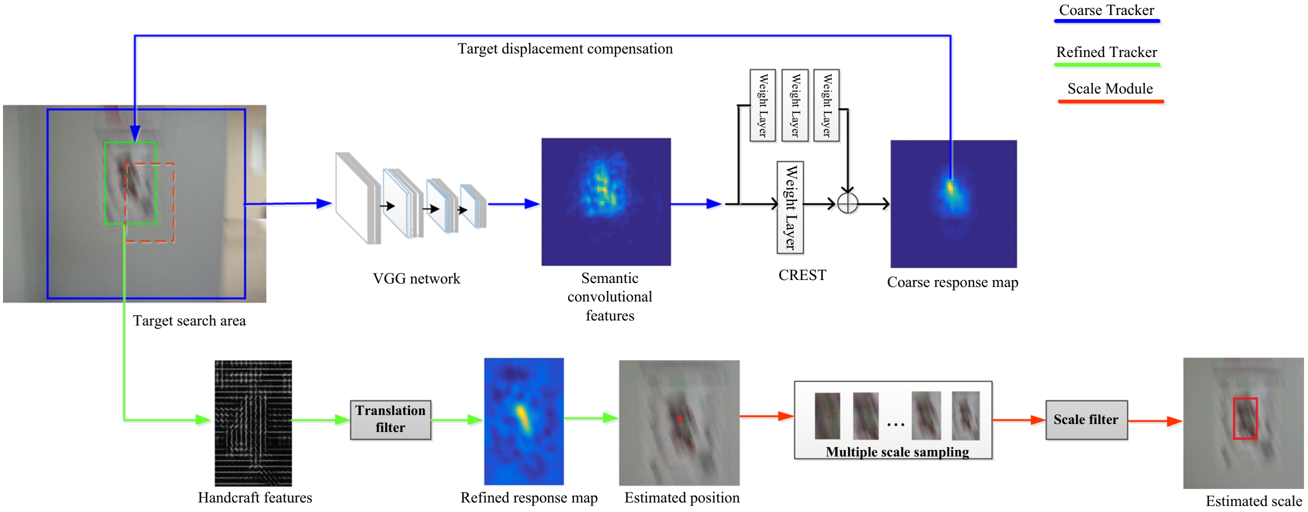

Fig. 1 Flowchart of our coarse-to-fine tracking framework

GPU. The coarse-grained location is also efficient with the fast CNN forward propagation in the coarse tracker $\mathcal{C}$. The fine-grained location is beyond real time with the element-wise solution of standard DCF. Therefore, the bottleneck for real-time performance lies in model update of the tracking pipeline.

With the closed-form solution, the refined tracker $\mathcal{R}$ can be efficiently trained element by element in the Fourier domain. On contrast, formulated as a convolutional kernel, the coarse tracker $\mathcal{C}$ has to be updated with the exhaustive stochastic gradient descent (SGD) method. Thanks to the semantics captured in the semantic convolutional features, we do not need to update $\mathcal{C}$ in each frame. Instead, we can set the updating frequency to make a good balance between computational efficiency and model adaptation.

4.4 Scale estimation

In the source codes provided in [15], scale variation is estimated by processing the search image at three scales. With no doubt, searching scale at multiple resolutions significantly increases the computational cost. To achieve fast scale adaptive tracking, our framework removes scale estimation from CREST and follows the scale filter in DSST and uses patch pyramid with the scale factors $\{a^s | s = [-\frac{S-1}{2}], [-\frac{S-3}{2}], ..., [\frac{S-3}{2}], [\frac{S-1}{2}]\}$. Instead, the coarse tracker $\mathcal{C}$ and the refined tracker $\mathcal{R}$ share the common one-dimensional scale correlation filter after the coarse-to-fine translation estimation process.

Algorithm 1 Coarse-to-fine Tracking

```
Input:
    Target state $X_{t-1} = (x_{t-1}, y_{t-1}, s_{t-1})$ in frame $t-1$. $(x_{t-1}, y_{t-1})$ is the target location while $s_{t-1}$ is the scale size.
Output:
    Estimated target state $X_t = (x_t, y_t, s_t)$ in frame $t$.
Tracking:
1: Crop the large image patch centered at $(x_{t-1}, y_{t-1})$ and feed it into the convolutional neural network to extract the semantic convolutional features.
2: Feed semantic convolutional features into the coarse tracker $\mathcal{C}$ to generate the coarse response map. Search for the coarse target location $(x_t^c, y_t^c)$ on the coarse-grained response map.
3: Crop a small image patch centered at $(x_t^c, y_t^c)$ and extract handcraft features.
4: Feed handcraft features into the refined tracker $\mathcal{R}$ to generate the fine-grained response map. Search for the refined target location $(x_t, y_t)$ on the fine-grained response map.
5: Estimate the new target scale $s_t$ with the scale filter.
6: Crop a large image patch centered at $(x_t, y_t)$ and extract semantic convolutional features for updating $\mathcal{C}$ with statistical gradient descent.
7: Crop a large image patch centered at $(x_t, y_t)$ and extract handcraft features for training $\mathcal{R}$ with the closed-form solution.
```

5 Experiments

Here, we present a comprehensive evaluation of the proposed tracker (CCT). Results are reported on two popular tracking benchmarks: OTB2013 [17] and TC128 [18].

5.1 Implementation details

For convenience reasons, we follow the default parameter setting of CREST and DSST as reported in [15] and [16], respectively. The CREST tracker employs the convolutional features extracted from the relu4_3 layer in the imagenet-vgg-verydeep-16 model for feature representation. This model can be downloaded from http://www.vlfeat.org/matconvnet/pretrained/. The target search area of CREST is set to be square and five times the target size. On contrast, DSST employs the 31-dimensional HOG features with 1 × 1 cell size. The target search area of DSST is set to be two times proportional to the target size. CREST is updated every two frames, while DSST is updated in each frame. Parameters are fixed for all videos in each dataset. Our tracker is implemented in Matlab and uses Matconvnet [27] for deep feature extraction. The comparison experiments of CCT are performed on a 4-core Intel Core -7-6700 CPU at 3.4GHz with a GeForce GTX TITAN GPU. The source codes for our approach are available at https://github.com/moqimubai/CCT.

5.2 Experiments on OTB2013

5.2.1 Overall performance

OTB2013 is a popular tracking benchmark which contains 50 videos. We evaluate CCT on OTB2013 in comparison with 8 state-of-the-art trackers from three typical categories: (1) correlation filter-based trackers, including CCOT [1], DeepSRDCF [13], SRDCF [23] and DSST [16]; (2) deep trackers, including CRSET [15], Siamfc3s [29] and cfnet [22]; (3) other trackers with collaborative modules, including Staple [11] and PTAV [30].

Following the protocol in OTB2013, we report the results in one-pass evaluation (OPE) using distance precision rate (DPR) and overlap success rate (OSR) as shown in Fig. 2 and Table 1. Overall, CCT performs favorably against all other state-of-the-art trackers in both rates. On OTB2013, our tracker achieves an DPR of 90.9% and an OSR of 67.8%. Though CCOT utilizes multiple feature integration to enhance feature representation, our approach performs better compared with its DPR of 90.8% and OSR of 67.7%. Our tracker performs better than both trackers, CRSET and DSST, in both the precision rate and success rate. Moreover, with our coarse-to-fine tracking framework, CCT (7 fps) achieves

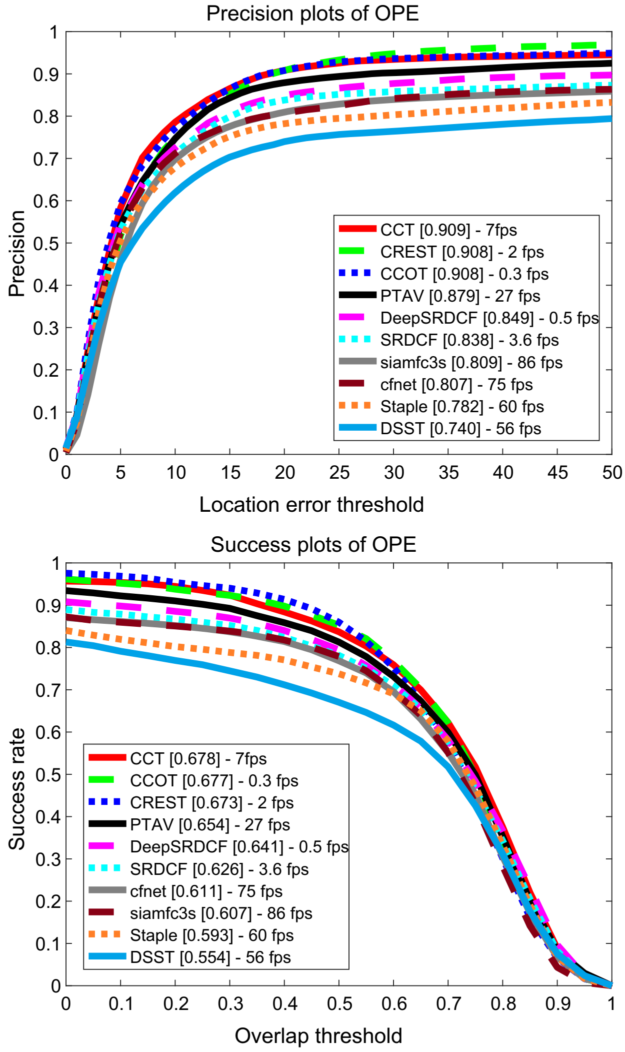

Fig. 2 Precision plots and success plots for all the trackers in comparison with OTB2013

over threefold speedup against CREST (2 fps) and over 20-fold speed up against CCOT (0.3 fps).

5.2.2 Attribute-based evaluation

We further analyze the performance of CCT under different attributes in OTB2013. All the videos in OTB2013 are annotated with 11 different attributes, namely illumination variation, scale variation, occlusion, deformation, motion blur, fast motion, in-plane rotation, out-of-plane rotation, out of view, background clutter and low resolution. Figure 3 shows the comparison of CCT with other tracking algorithms on these eleven attributes. On all the eleven attributes, CCT achieves competitive performances which demonstrates its robustness in challenging tracking scenarios.

5.2.3 Qualitative evaluation

To intuitively demonstrate the superiority of CCT, we further present some screenshots of the tracking results on benchmark sequences from OTB2013. Figure 4 shows screenshots from 7 challenging videos in the OTB2013 dataset. Here, we compare CCT against SRDCF [23], DSST [16] and cfnet [22] with respect to five challenging attributes (e.g., fast motion, deformation, partial occlusion, short-term full occlusion and background clutter). The videos (from top to bottom) are *Skiing*, *Bolt*, *Jogging1*, *Lemming*, *Ironman* and *Soccer*.

In the *Skiing* sequence, the target undergoes large displacement between adjacent frames due to fast motion. DSST, SRDCF and cfnet drift to the background in the beginning of the sequence. On contrast, our CCT tracker persistently tracks the target with the coarse tracker which holds a large search area.

In the *Bolt* sequence, the target undergoes shape deformation. DSST and SRDCF lose track of the target due to the employed HOG feature which is sensitive to geometric deformation. Our CCT tracks the target well with the coarse tracker which employs high-level convolutional features. These features encode semantics and are robust to target deformation.

In the *Jogging1* and *Lemming* sequences, the targets undergo partial occlusion and short-term full occlusion, respectively. Only the CCT tracker manages to track the target until the end of both sequences. It is worth noting that DSST fails to track the target in both sequences, while our CCT tracker manages to track the target from beginning to end. This demonstrates the effectiveness of the coarse tracker in re-detecting tracking failures and resuming tracking of the target.

In the *Ironman* and *Soccer* sequences, our CCT tracker demonstrates strong robustness against background clutters,

Table 1 Quantitative comparison of DPR, OSR and average frame rate (FPS) of all trackers with OTB2013

	CCT	CCOT	CREST	PTAV	DeepSRDCF	SRDCF	cfnet	siamfc3s	Staple	DSST
DPR (%)	*90.9*	**90.8**	**90.8**	87.9	84.9	83.8	80.7	80.9	78.2	74.0
OSR (%)	*67.8*	**67.7**	67.3	65.4	64.1	62.6	61.1	60.7	59.3	55.4
Avg. FPS	7	0.3	2	27	0.5	3.6	**75**	*86*	60	56

The *best* and **second best** values are highlighted in color

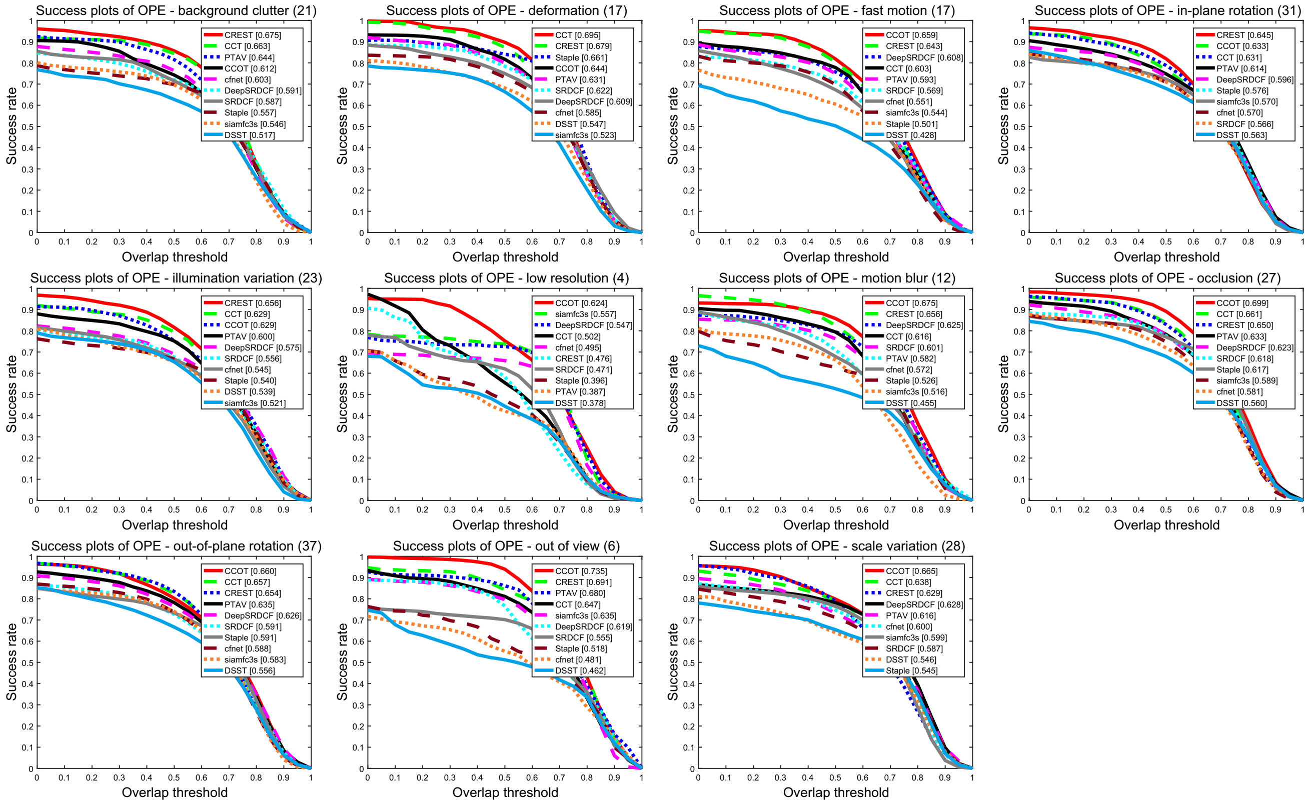

Fig. 3 *Success ratio* plots on 11 attributes of the OTB2013 dataset. Trackers are ranked by their AUC scores

which can also be attributed to the semantic convolutional features in the coarse tracker.

Overall, it is easy to see that CCT performs better than the compared trackers in the presence of fast motion (*Skiing*), deformation(*Bolt*) , partial occlusion (*Jogging1*), short-term full occlusion (*Lemming*) and background clutter (*Ironman* and *Soccer*). However, like most trackers, CCT tends to drift to the background in the presence of long-term full occlusion as shown in Fig. 5. In future work, we tend to solve this problem by equipping CCT with a re-detection module to achieve long-term tracking.

5.3 Experiment on TC128

Experimental results on the TC128 dataset containing 128 videos are shown in Fig. 6. Our CCT is compared with all the default trackers in TC128. Among the 18 compared trackers, our CCT obtains the best distance precision rate (DPR) of 70.62% and the best overlap success rate (OSR) of 52.24%. By comparison, CCT achieves significant performance improvement, demonstrating the advantages of our coarse-to-fine tracking framework.

5.4 Detailed analysis of CCT

Different coarse tracker $\mathcal{C}$. As shown in Fig. 2, CREST achieves a tracking speed of only 2 fps. Our CCT improves the tracking frame rates to 7 fps, but is still far from real-time tracking. We attribute the low computational efficiency of CCT to the SGD training in CREST. Therefore, for computationally restricted applications, we can reduce the updating frequency of CREST or replace CREST with Siamfc3s in $\mathcal{C}$ to achieve real-time tracking.

Different refined tracker $\mathcal{R}$. In the refined tracker $\mathcal{R}$, DSST can be replaced with its fast version, fDSST [7], to further improve the tracking speed. Alternatively, DSST can be replaced with Staple to further improve the localization accuracy and robustness to target deformation.

Feature sharing. In our approach, the coarse tracker $\mathcal{C}$ employs high-level semantic convolutional features extracted from the deep layers of a convolutional neural network for target representation. However, the ready-made shallow convolutional features extracted from the earlier layers are wasted. It is worth noting that the extraction of shallow convolutional features also consumes a lot of computing resources. In the future work, we will explore the potential

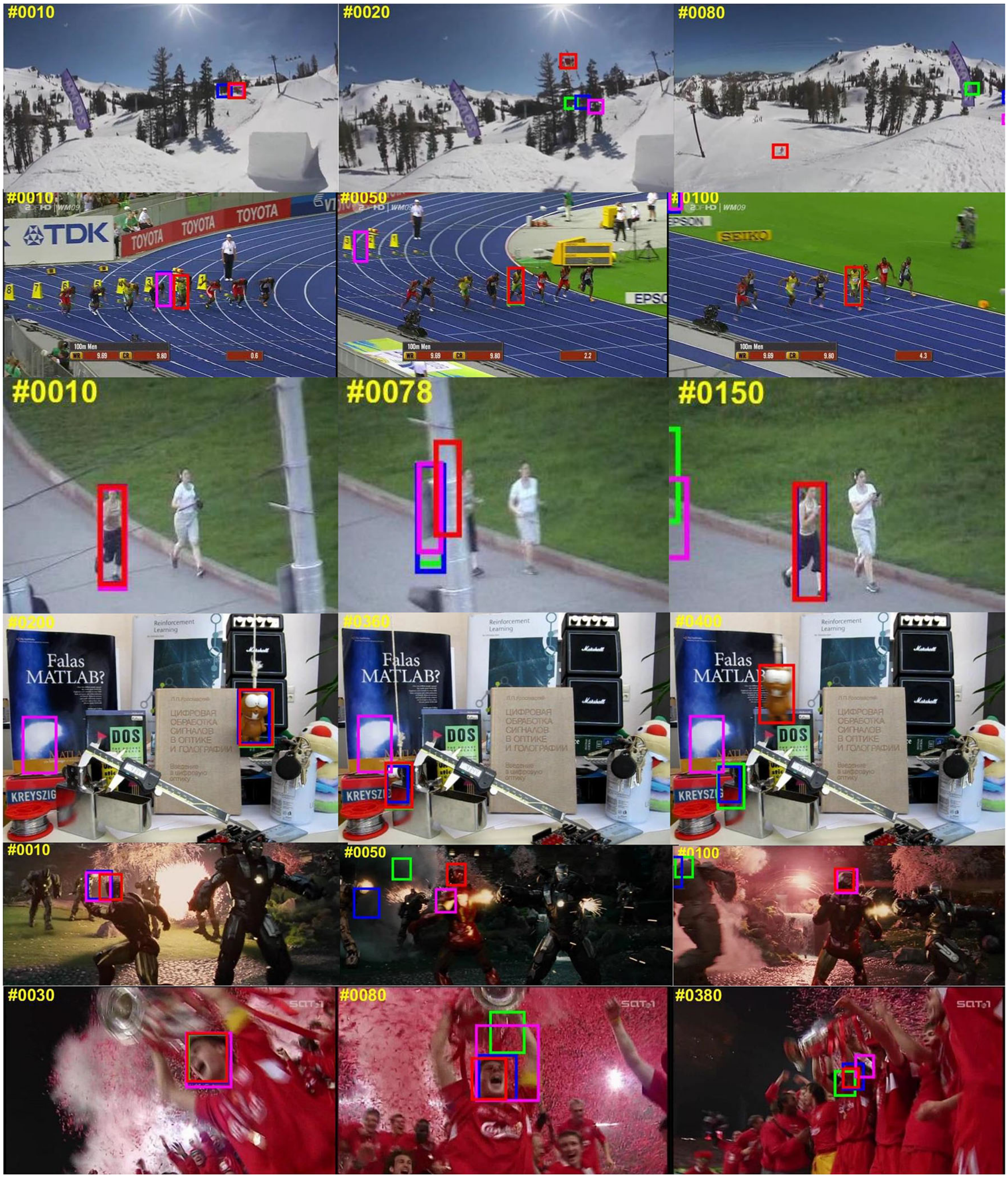

Fig. 4 Tracking screenshots of CCT, DSST, SRDCF and cfnet. The videos (from top to bottom) are *Skiing*, *Bolt*, *Jogging1*, *Lemming*, *Ironman* and *Soccer* from the OTB2013 dataset

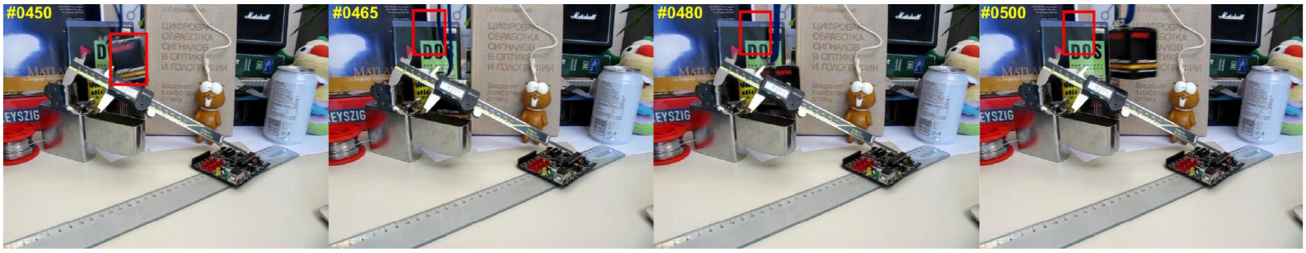

Fig. 5 Tracking failure of our CCT tracker in the presence of long-term full occlusion

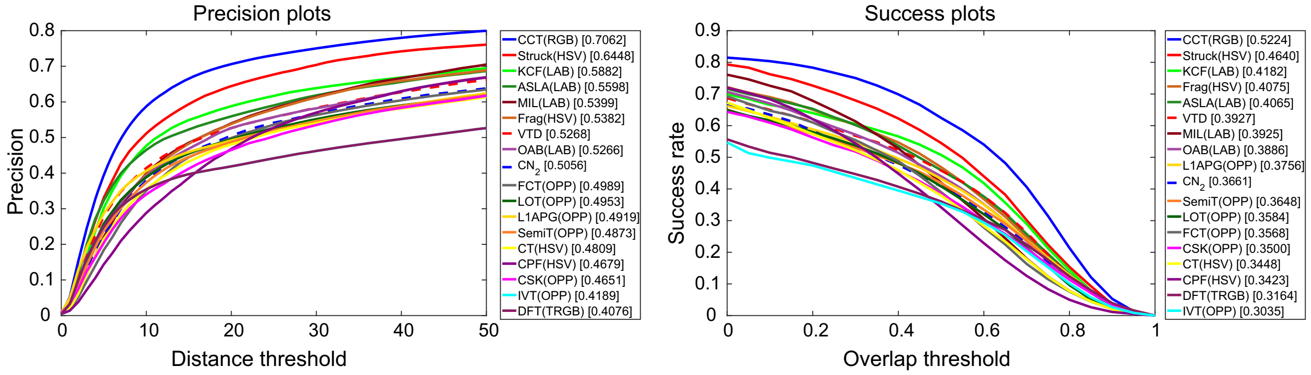

Fig. 6 Precision plots and success plots for all the trackers in comparison with TC128

of replacing handcraft features with the ready-made shallow convolutional features in the refined tracker $\mathcal{R}$.

6 Conclusion

In this paper, we propose a new coarse-to-fine tracking framework for cascade correlation tracking (CCT), which decomposes visual tracking into two subtasks, coarse-grained tracking and fine-grained tracking. We show that the coarse tracker and refined tracker can cooperate in a coarse-to-fine manner to achieve accurate and fast tracking. With these two collaborative modules, CCT achieves encouraging results on the OTB2013 and TC128 benchmarks while maintaining low model complexity. Since CCT is a very flexible framework with great rooms for improvement and generalization, we expect this work to stimulate the designing of more efficient tracking algorithms in the future.

Acknowledgements This work is supported by the Natural Science Foundation of China (NSFC) (Nos. 61701506, 61671456).

References

1. Danelljan, M., Robinson, A., Khan, F.S., Felsberg, M.: Beyond correlation filters: Learning continuous convolution operators for visual tracking. In: Proceedings , Part V, Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, pp. 472–488, (2016)
2. Ma, C., Huang, JB., Yang, X., Yang, MH.: Hierarchical convolutional features for visual tracking. In: IEEE International Conference on Computer Vision (ICCV), (Dec 2015), pp. 3074–3082
3. Lee, K.-H., Hwang, J.-N.: On-road pedestrian tracking across multiple driving recorders. IEEE Trans. Multimed. **17**(9), 1429–1438 (2015)
4. Guan, T., Wang, C.: Registration based on scene recognition and natural features tracking techniques for wide-area augmented reality systems. IEEE Trans. Multimed. **11**(8), 1393–1406 (2009)
5. Guile, W., Kang, W.: Vision-based fingertip tracking utilizing curvature points clustering and hash model representation. IEEE Trans. Multimed. **19**(8), 1730–1741 (2017)
6. Danelljan, M., Khan, FS., Felsberg, M., Weijer, J.v.d. : Adaptive color attributes for real-time visual tracking. In: IEEE Conference on Computer Vision and Pattern Recognition, (June 2014), pp. 1090–1097
7. Danelljan, M., Hager, G., Khan, FS., Felsberg M.: Discriminative scale space tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence, (99), pp.1–1 (2016)
8. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: High-speed tracking with kernelized correlation filters. IEEE Trans. Pattern Anal. Mach. Intell. **37**(3), 583–596 (2015)
9. Ma, C., Yang, X., Zhang, Chongyang., Yang MH.: Long-term correlation tracking. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (June 2015), pp. 5388–5396
10. Bibi, Adel., Mueller, Matthias., Ghanem, Bernard.: Target response adaptation for correlation filter tracking. In Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI, pp. 419–433, (2016)
11. Bertinetto, L., Valmadre, J., Golodetz, S., Miksik, O., Torr, P.H.S.: Staple: Complementary learners for real-time tracking. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (June 2016), pp. 1401–1409
12. Mueller, Matthias., Smith, Neil., Ghanem, Bernard.: Context-aware correlation filter tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2017)
13. Danelljan, M., Hger, G., Khan, F.S., Felsberg, M.: Convolutional features for correlation filter based visual tracking. In: IEEE International Conference on Computer Vision Workshop (ICCVW), (Dec 2015), pp. 621–629
14. Kristan, Matej., Leonardis, Aleš., Matas, Jiri., Felsberg, Michael., Pflugfelder, Roman., Čehovin, Luka., Vojir, Tomas., Häger, Gustav., Lukežič, Alan., Fernandez, Gustavo: The visual object tracking vot2016 challenge results. Springer, (Oct 2016)
15. Song, Yibing., Ma, Chao., Gong, Lijun., Zhang, Jiawei., Lau,Rynson W.H., Yang, Ming-Hsuan .: CREST: convolutional residual learning for visual tracking. In: IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29 (2017), pp. 2574–2583
16. Danelljan, Martin., Häger, Gustav., Khan, Fahad Shahbaz., Felsberg, Michael.: Accurate scale estimation for robust visual tracking. In: British Machine Vision Conference, BMVC 2014, Nottingham, UK, September 1-5, (2014)
17. Wu, Y., Lim, J., Yang. MH.: Online object tracking: A benchmark. In: *IEEE Conference on Computer Vision and Pattern Recognition*, (June 2013), pp. 2411–2418

18. Liang, P., Blasch, E., Ling, H.: Encoding color information for visual tracking: algorithms and benchmark. IEEE Trans. Image Process. **24**(12), 5630–5644 (2015)
19. Wu, Y., Lim, J., Yang, M.H.: Object tracking benchmark. IEEE Trans. Pattern Anal. Mach. Intell. **37**(9), 1834–1848 (2015)
20. Bolme, DS., Beveridge, JR., Draper, BA., Lui, YM.: Visual object tracking using adaptive correlation filters. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, (June 2010), pp. 2544–2550
21. Henriques, João F., Caseiro, Rui., Martins, Pedro., Batista. Jorge P.: Exploiting the circulant structure of tracking-by-detection with kernels. In Computer Vision - ECCV 2012 - 12th European Conference on Computer Vision, Florence, Italy, October 7–13, 2012, Proceedings, Part IV, pp. 702–715, (2012)
22. Valmadre, Jack., Bertinetto, Luca., Henriques, João F., Vedaldi, Andrea., Torr, Philip HS.: End-to-end representation learning for correlation filter based tracking. CoRR, abs/1704.06036, (2017)
23. Danelljan, M., Hger, G., Khan, FS., Felsberg, M.: Learning spatially regularized correlation filters for visual tracking. In: IEEE International Conference on Computer Vision (ICCV), (Dec 2015), pp. 4310–4318
24. Galoogahi, Hamed Kiani., Fagg, Ashton., Lucey, Simon.: Learning background-aware correlation filters for visual tracking. CoRR, abs/1703.04590, (2017)
25. Kalal, Z., Mikolajczyk, K., Matas, J.: Tracking-learning-detection. IEEE Trans. Pattern Anal. Mach. Intell. **34**(7), 1409–1422 (2012)
26. Simonyan, Karen., Zisserman, Andrew.: Very deep convolutional networks for large-scale image recognition. ICLR, (2014)
27. Vedaldi, Andrea., Lenc, Karel.: Matconvnet: Convolutional neural networks for matlab. In: Proceedings of the 23rd ACM international conference on Multimedia, pp. 689–692 (2015)
28. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 1 (June 2005), pp. 886–893
29. Bertinetto, Luca., Valmadre, Jack., Henriques, João F., Vedaldi, Andrea ., Torr, Philip H.S. : Fully-convolutional siamese networks for object tracking. In Computer Vision - ECCV 2016 Workshops - Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part II, pp. 850–865, (2016)
30. Fan, Heng., Ling, Haibin.: Parallel tracking and verifying: a framework for real-time and high accuracy visual tracking. In IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29 (2017), pp. 5487–5495

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

Dongdong Li received his BS degree from Wuhan University in 2012 and his MS degree from National University of Defense Technology in 2014. He has been pursuing his doctors degree in information and communication engineering since 2015. His main research interests include visual tracking, camera calibration and deep learning. He serves as a reviewer for Optical Engineering, Optics and Lasers in Engineering and IEEE Trans Multimedia.

Gongjian Wen received his BS, MS and PhD degrees from National University of Defense Technology in 1994, 1997 and 2000, respectively. He completed two-year postdoctoral assignment at Wuhan University. Currently, he is the head of the fourth Department of Science and Technology on Automatic Target Recognition Laboratory and is mainly interested in image understanding and photogrammetry and remote sensing.

Yangliu Kuai received his BS and MS degrees from National University of Defense Technology in 2013 and 2015, respectively. She has been pursuing his doctor's degree in information and communication engineering since 2016. Her main research interests include visual tracking and deep learning. She serves as a reviewer for Journal of Visual Communication and Image Representation.

Fatih Porikli is an IEEE Fellow and a Professor with the Research School of Engineering, Australian National University, Canberra, Australia. He is also acting as the Leader of the Computer Vision Group at NICTA, Australia. He received his Ph.D. degree from NYU. Previously, he served as a Distinguished Research Scientist at Mitsubishi Electric Research Laboratories, Cambridge, USA. He has contributed broadly to object detection, motion estimation, tracking, image-based representations, and video analytics. He is the coeditor of two books on Video Analytics for Business Intelligence and Handbook on Background Modeling and Foreground Detection for Video Surveillance. He is an Associate Editor of five journals. His publications won four Best Paper Awards and he has received the R&D 100 Award in the Scientist of the Year category in 2006. He served as the General and Program Chair of numerous IEEE conferences in the past. He has 66 granted patents.