Graph Convolutional Tracking

Junyu Gao[1,2,3], Tianzhu Zhang[1,2,4] and Changsheng Xu[1,2,3]
[1] National Lab of Pattern Recognition (NLPR),
Institute of Automation, Chinese Academy of Sciences (CASIA)
[2] University of Chinese Academy of Sciences (UCAS)
[3] Peng Cheng Laboratory, ShenZhen, China
[4] University of Science and Technology of China
{junyu.gao, csxu}@nlpr.ia.ac.cn, tzzhang10@gmail.com

Abstract

Tracking by siamese networks has achieved favorable performance in recent years. However, most of existing siamese methods do not take full advantage of spatial-temporal target appearance modeling under different contextual situations. In fact, the spatial-temporal information can provide diverse features to enhance the target representation, and the context information is important for online adaption of target localization. To comprehensively leverage the spatial-temporal structure of historical target exemplars and get benefit from the context information, in this work, we present a novel Graph Convolutional Tracking (GCT) method for high-performance visual tracking. Specifically, the GCT jointly incorporates two types of Graph Convolutional Networks (GCNs) into a siamese framework for target appearance modeling. Here, we adopt a spatial-temporal GCN to model the structured representation of historical target exemplars. Furthermore, a context GCN is designed to utilize the context of the current frame to learn adaptive features for target localization. Extensive results on 4 challenging benchmarks show that our GCT method performs favorably against state-of-the-art trackers while running around 50 frames per second.

1. Introduction

Visual tracking is a fundamental task in computer vision community, where the target object is localized in a changing video sequence automatically. It has various applications such as intelligent video surveillance, human-computer interaction, robotics, and autonomous driving, to name a few [71, 33, 13, 24, 74, 42, 17]. Despite much progress has been achieved in recent years [38, 3, 45, 9, 66, 36, 20, 82, 22], visual tracking remains difficult due to tremendous challenges such as occlusion, background clutter, illumination variation, scale variation, motion blur, fast motion, and deformation.

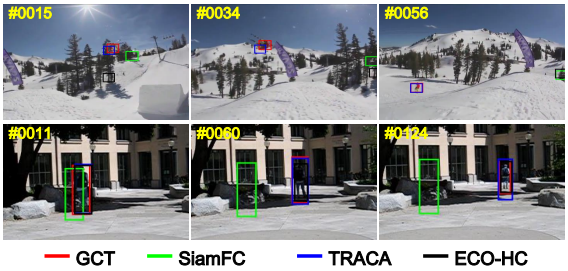


Figure 1. Comparison of our proposed tracker with the popular SiamFC tracker [2] and other two state-of-the-art methods.

Recently, tracking by siamese networks has attracted an increasing attention in the tracking community, which learns a similarity metric between the target object and candidate patches of the current search image in an end-to-end framework [60, 2, 63, 23, 66, 36, 25, 84]. With the powerful deep network and large-scale labeled video frames for offline training, siamese based trackers achieve favorable performance and efficiency. One notable example is the SiamFC tracker [2] which learns a matching function in an embedding space and wins the VOT2017 real-time challenge [33, 32]. However, based on the results in existing tracking benchmarks [70, 71], SiamFC does not achieve a better accuracy than many other types of trackers, such as ECO-HC [10] and TRACA [5]. Figure 1 also shows that SiamFC encounters difficulties when the target object has significant appearance change [25]. To improve the robustness of siamese based methods, various strategies have been proposed such as attention learning [66], dynamic updating [23] and structured modeling [84], which have obtained promising performance.

Despite the above significant progress, most siamese based tracking methods do not take full advantage of spatial-temporal target appearance modeling under different contextual situations : **(1)** Many siamese trackers use the initial target template from the first frame to match candidate patches [60, 2, 36]. However, since visual tracking is a

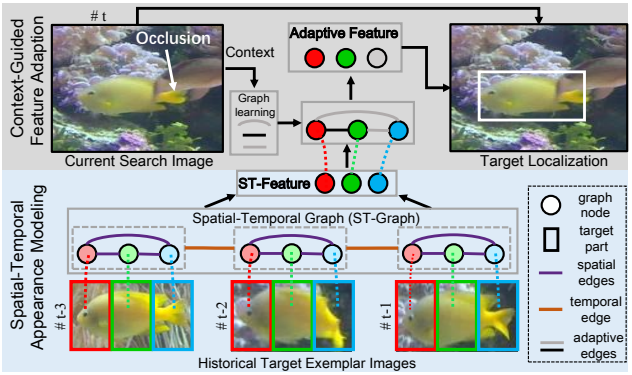


Figure 2. Illustration of our motivation. From bottom to top: (1) In spatial-temporal appearance modeling, the historical target exemplar images are converted to a ST-graph, where each target part corresponds to a graph node (red, green, blue ones in this example). By leveraging this graph, diverse target parts are considered to generate a robust ST-Feature for representing the target object. (2) In context-guided feature adaption, the current search image provides useful foreground/ background information, which helps conduct graph learning for feature adaption. Here, the red and green parts are more important than the blue one because the blue part is occluded in the current search image. With the adaptive feature, robust target localization can be achieved.

dynamic process with changing scenes, there exists a strong spatial-temporal relationship between the target object appearances in consecutive frames. Features from different frames and locations provide diverse information for target appearance modeling [76, 61], such as different parts and viewpoints, motion, deformation, and varied illuminations. In the tracking process, to characterize rotation and translation invariance of target objects, image patches can be modeled as a grid graph [8]. As shown in the bottom of Figure 2, different target parts from historical exemplar images can be organized as a spatial-temporal graph, where the ST-feature can be learned comprehensively for representing the target appearance. **(2)** The surrounding context of the target object has a big impact on tracking performance [48]. However, most existing siamese tracking methods ignore context information of search images for guiding the adaption of target appearance model. Due to lack of the online adaptability, they can hardly capture the variations of target objects, backgrounds or situations in search images well, which may lead to tracking failures [23]. We point out that visual tracking can benefit from the current context information. As shown in the top of Figure 2, with the help of the current context, a new graph is learned as the adaption guidance. Based on the learned graph, the feature used for target localization can be adaptively changed by focusing on the first two parts (green and red ones) and paying less attention to the last part (the blue one) since the part is occluded in the search image. While some methods utilize attention learning [87, 66] or transformation learning [23] for online adaption, they only use the previous target exemplars for updating the target appearance model but ignore the context information from the current search image.

Inspired by the above observation, it is desirable to automatically capture the spatial-temporal patterns of target appearance under the context information of current search image. During the tracking process of siamese based methods, the target exemplar sequence can be organized as a 3D spatial-temporal graph where each target part is considered as a node. Although 3D CNN [62] can be applied for spatial-temporal modeling, it is computationally expensive [52] and cannot handle arbitrary graph structures. Recently, *Graph Convolutional Networks (GCNs)*, which can model the dependencies and propagate messages between different nodes in an arbitrary graph, have received increasing attention and successfully been adopted in various computer vision tasks [35, 46, 67, 72, 56]. Until now, the application of GCNs to visual tracking is yet to be explored.

In this paper, we propose an end-to-end Graph Convolutional Tracking (GCT) method based on a siamese framework, which can jointly consider both the spatial-temporal target appearance structure of historical frames and the context information of the current search image. As shown in Figure 3, for target appearance modeling, we construct a spatial-temporal graph to form a structured representation of the historical target exemplars. A Spatial-Temporal GCN (ST-GCN) is employed to learn a robust target appearance model on this graph and generate a spatial-temporal feature (ST-Feature). Furthermore, to incorporate context information of the current search image for target localization, we combine the ST-Feature and the context feature to produce an adaptive graph. A ConText GCN (CT-GCN) then operates on this graph and generates the adaptive feature for target localization. To make the tracking highly efficient, all the learning processes are performed in offline training. We validate the effectiveness and efficiency of our approach on five popular tracking benchmarks [70, 71, 33, 40, 47].

To summarize, the main contributions of this paper are three-fold:

- An end-to-end graph convolutional tracking framework is explored. To the best of our knowledge, this is the first work to train GCNs in a deep siamese network for visual tracking.
- Both ST- and CT-GCN are designed in a siamese network. The proposed GCT can jointly achieve spatial-temporal target appearance modeling and context-guided adaptive learning for robust target localization.
- Extensive experimental results on five visual tracking benchmarks demonstrate that the proposed GCT algorithm performs favorably against state-of-the-art trackers and runs in real-time.

2. Related Work

Tracking by Siamese Network. One simple yet effective manner of using deep learning for visual tracking is to di-

rectly apply siamese networks as a matching function between target object and candidate patches [60, 2, 26, 63, 23, 66, 87, 36]. The pioneering work, [redacted] [60], learns a matching function in the off-line phase and applies it to find the most similar target candidate in online tracking. Despite SINT achieves promising tracking performance, its speed is only 2 fps because of the candidate sampling process. To improve running speed, Bertinetto *et al*. [2] propose a fully convolutional Siamese framework (SiamFC) to conduct similarity learning in an embedding space, which runs nearly 86 fps with a GPU. Recently, more siamese network based tracking methods have been proposed with real-time high quality. Guo *et al*. [23] propose a dynamic Siamese network (DSiam), which adopts a transformation learning model to adaptively conduct online learning. Wang *et al*. [66] introduce different kinds of attention mechanisms in siamese learning, which mitigates the over-fitting problem and enhances its discriminative capacity. Moreover, other strategies are also adopted to improve the performance of siamese tracking, such as two-fold learning [25], triplet loss optimization [12], region proposal network [36], adversarial learning [68], deep reinforcement learning [30], distractor-aware module [86], and structured modeling [84]. Different from the above methods, we are among the first to utilize graph convolutional operators in siamese networks to comprehensively model the structured cues of a target object, which can jointly consider the spatial-temporal structure and current context information.

Structured Target Appearance Modeling. To handle various challenges in visual tracking scenes, a number of tracking algorithms have been proposed to impose structure information on target appearance modeling. Some trackers explore spatial-temporal modeling in visual tracking [76, 59, 57, 87, 37, 61, 77, 79, 83]. However, these methods are either not end-to-end trainable [76, 37] or only using a holistic target appearance model [57, 87, 61, 21]. For example, although FlowTrack [87] utilizes optical flow to get benefit from inter-frame motion cues, it only adopts the holistic model for target representation and ignores detailed information such as interactions between local target parts. Recently, part-based methods that decompose target object into several parts have been studied actively [8, 78, 22, 39, 7, 8, 80, 20, 82, 85]. For example, a spectral tracking method [8] is proposed to operate on localized surrounding regions of each pixel via graph filters. With the development of deep learning techniques, some part-based methods learn structured information in an end-to-end fashion [84, 15]. Zhang *et al*. [84] utilize conditional random field as a message passing module for learning local structure in a siamese network. However, most existing part-based trackers only consider spatial-structure information of previous frames for locating target object, and can hardly benefit from the long-range temporal information. In this paper, we make full use of both spatial-temporal target structure and context information of search images for target localization in an end-to-end siamese framework.

Graph Neural Networks for Computer Vision. Generalization of neural networks for arbitrarily structured graphs has drawn great attention in recent years. There are two typical ways to develop graph neural networks. On one hand, some methods adopt feed-forward neural networks to every node in a spatial manner [55]. On the other hand, spectral methods provide well-defined localization operators on graphs via convolutions in the Fourier domain [31]. For computer vision tasks, Wang *et al*. [67] propose to represent videos as space-time region graphs which capture similarity relationships and spatial-temporal relationships. To model dynamic skeletons for human action recognition, Yan *et al*. [72] propose a spatial-temporal graph convolutional network with several types of kernels. Shen *et al*. [56] utilize graph convolutional operator to learn probe-gallery relationships for person re-identification. Gao *et al*. utilize graph neural networks to improve the performance of video classification [18] and zero-shot video classification [19].

3. Graph Convolutional Tracking

In this work, we propose a Graph Convolutional Tracker, GCT, which jointly performs spatial-temporal target appearance modeling and context-guided adaptive learning in an end-to-end manner. Figure 3 overviews the pipeline of the proposed tracking algorithm based on a siamese architecture (SiamFC) [2]. The SiamFC learns a similarity function $f(z, x)$ to compare a 127×127 exemplar image z to a 255×255 search image x in a learned convolutional feature embedding space ϕ (we denote $\mathbf{Z} = \phi(z), \mathbf{X} = \phi(x)$):

$$\begin{aligned} f(z, x) &= \phi(z) \star \phi(x) + b \\ &= \mathbf{Z} \star \mathbf{X} + b, \end{aligned} \tag{1}$$

where $\star$ represents cross-correlation between two feature maps, $b \in \mathcal{R}$ denotes a bias for each location. By using Eq. (1), the most similar patch from the search image will be selected as the target object. Despite the favorable efficiency and expansibility of SiamFC, it only uses the first frame as a fixed template in the whole tracking process, which can hardly benefit from the spatial-temporal structure of target appearance under different contextual situations. In fact, features from different frames and locations provide diverse and abundant information for the target appearance modeling [76], such as different parts and viewpoints, motion, deformation, and varied illuminations. For target localization, these features should be adaptively aggregated in spatial-temporal domain guided by the context information of the current search image. To this end, we design a graph convolutional transformation into the siamese architecture to jointly consider target appearance modeling

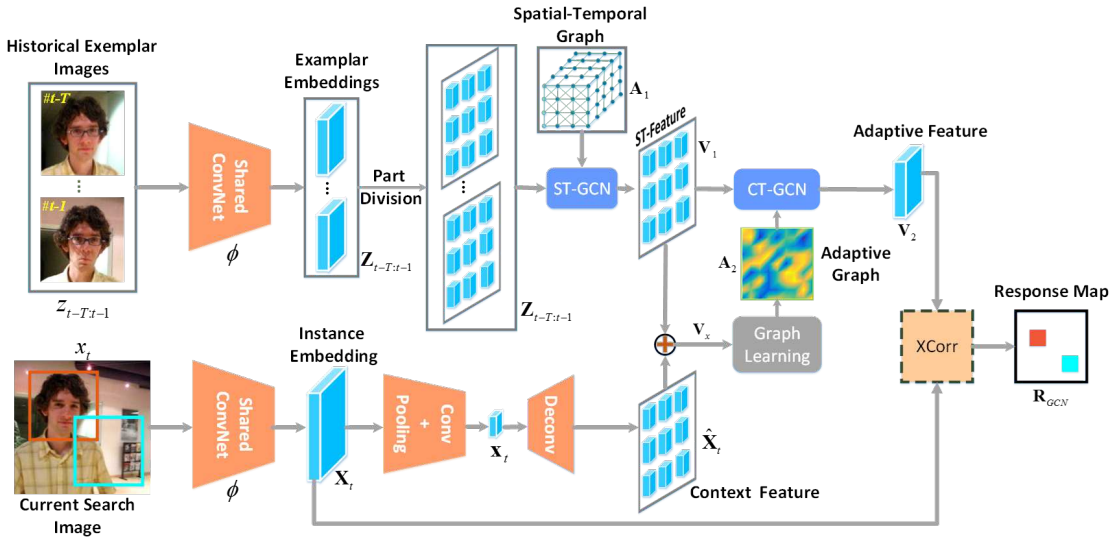


Figure 3. The pipeline of our GCT, which can jointly perform spatial-temporal target appearance modeling and context-guided feature adaption in a siamese framework. Specifically, we use a ST-GCN to model the historical exemplars with a spatial-temporal graph. Then, the generated ST-feature is combined with the current context feature to learn an adaptive graph, which is used by CT-GCN to produce the adaptive feature. This feature is evaluated on the search image embedding via a cross-correlation layer (XCorr) for target localization.

with context information of the current search image:

$$f(z_{t-T:t-1}, x_t) = \psi_{GCN}(\mathbf{Z}_{t-T:t-1}, \mathbf{X}_t) \star \mathbf{X}_t + b, \quad (2)$$

where ψ_{GCN} denotes the proposed graph convolutional transformation. It aims to learn robust spatial-temporal features of the target object in previous frames $t-T:t-1$, guided by the context information of the current search image embedding $\mathbf{X}_t$. T controls the time range for remembering historical information. However, learning ψ_{GCN} is not efficient since it suffers from high computational burden for modeling the message passing between current context information $\mathbf{X}_t$ and each of the historical exemplar embeddings $\mathbf{Z}_{t-T:t-1}$. To reduce the computational cost, we further decompose ψ_{GCN} into two sequential graph convolution modules named Spatial-Temporal GCN (ST-GCN) ψ_1 and ConText GCN (CT-GCN) ψ_2. As a result, the decomposed formulation is:

$$f(z_{t-T:t-1}, x_t) = \psi_2(\psi_1(\mathbf{Z}_{t-T:t-1}), \mathbf{X}_t) \star \mathbf{X}_t + b, \quad (3)$$

where ψ_1 conducts spatial-temporal target appearance modeling for historical exemplars and generates aggregated ST-feature $\mathbf{V}_1 = \psi_1(\mathbf{Z}_{t-T:t-1})$. ψ_2 takes $\mathbf{V}_1$ and the context information of current search image embedding $\mathbf{X}_t$ for learning the adaptive feature $\mathbf{V}_2$, which is then evaluated on the search image embedding $\mathbf{X}_t$ via cross-correlation. In the offline training stage, the loss of an exemplars-instance pair is generally represented as a logistic function [2] :

$$L(z_{t-T:t-1}, x_t, \mathbf{Y}) = \frac{1}{|\nabla|} \sum_{u \in \nabla} \log(1 + \exp(-\mathbf{Y}[u]\mathbf{R}[u])), \quad (4)$$

where ∇ is the set of all the shifting positions on the search image and u denotes a sample of the same size with the target template. $\mathbf{Y}[u] \in \{+1, -1\}$ is the ground-truth label as in [2], and $\mathbf{R}[u] = \mathbf{V}_2[u] \cdot \mathbf{X}_t[u]$ is the response score.

In the following, we first introduce the preliminary of our main building block, GCN [31], which generalizes CNN to graphs. Then we illustrate both ST-GCN and CT-GCN. The details of our tracking method are finally presented.

3.1. Preliminary: Graph Convolutional Networks

Given an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with M nodes $\mathcal{V}$, a set of edges $\mathcal{E}$ between nodes, an adjacency matrix $\mathbf{A} \in \mathrm{R}^{M \times M}$, and a degree matrix $\mathbf{\Lambda}_{ii} = \sum_j \mathbf{A}_{ij}$. We formulate a linear transformation of graph convolution as the multiplication of a graph signal $\mathbf{X} \in \mathrm{R}^{D \times M}$ (the column vector $\mathbf{X}_{i\cdot} \in \mathrm{R}^D$ is the feature representation at the i^{th} node) with a filter $\mathbf{W} \in \mathrm{R}^{D \times C}$:

$$\mathbf{V} = \hat{\mathbf{\Lambda}}^{-\frac{1}{2}} \hat{\mathbf{A}} \hat{\mathbf{\Lambda}}^{-\frac{1}{2}} \mathbf{X}^\top \mathbf{W}, \quad (5)$$

where $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}$, $\mathbf{I}$ is the identity matrix. $\hat{\mathbf{\Lambda}}_{ii} = \sum_j \hat{\mathbf{A}}_{ij}$. In this formulation, the output is a $C \times M$ matrix $\mathbf{V}$. Note that a GCN can be built by stacking multiple graph convolutional layers of the form of Eq. (5), each layer followed by a non-linear operation (such as ReLU). Readers can refer to [31] for more details and an in-depth discussion.

3.2. Target Appearance Modeling via ST-GCN

Spatial-temporal structure of target object is crucial for robust visual tracking. However, most existing siamese network based methods either describe the target appearance from the global view or ignore the historical information in an end-to-end training, resulting in high sensitivity to significant appearance change. In this section, we design a spatial-temporal graph to form a structured representation of the historical exemplar (target object) sequence.

Specifically, the shared ConvNet ϕ in the exemplar branch (the top of Figure 3) takes the historical exemplar images $\{z_i\}_{i=t-1}^{t-T}$ as inputs and produces the corresponding embeddings $\{\mathbf{Z}_i\}_{i=t-1}^{t-T}$. Here, $\mathbf{Z}_i \in \mathrm{R}^{D_1 \times M_z}$, where D_1 and M_z represent the feature dimensionality and the number of parts respectively. Although other automatic part generation methods [84, 67] can be exploited, for simplicity and efficiency, we follow [7, 8] to consider each $D_1 \times 1 \times 1$ grid of the feature map $\mathbf{Z}_i$ as a target part. To perform spatial-temporal modeling of target object, we construct an undirected ST-graph $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1)$ on an exemplar embedding sequence with M_z parts (nodes) and T frames featuring both intra-exemplar and inter-exemplar relationships.

In the graph $\mathcal{G}_1$, the node set $\mathcal{V}_1 = \{v_{ij} | i = t-1, ..., t-T, j = 1, ..., M_z\}$ consists of all the target parts in an exemplar embedding sequence. The edge set $\mathcal{E}_1$ is composed of two types of edges: (1) Spatial edges $\mathcal{E}_1^S$ represents the intra-exemplar connection at each frame: $\mathcal{E}_1^S = \{v_{ij}v_{ik} | 1 \leqslant j, k \leqslant M_z, j \neq k\}$. Note that similar with [8], we adopt a fully-connected graph to depict the spatial relationships since all the target parts may have interactions under various appearance changes. In addition, in our experiment, we find that the fully-connected graph achieves favorable performance while needs less graph convolutional layers than other types of graphs such as k-nearest neighbor graph [8]. (2) Follow [72], we connect the parts with the same location in consecutive frames as the temporal edges $\mathcal{E}_1^T = \{v_{ij}v_{i+1,j}\}$. As a result, the information can be propagated in the temporal domain. With both types of edges, each node is connected to at most $M_z + 1$ nodes among a total of $M_z T$ nodes in $\mathcal{V}_1$, which makes the ST-graph sparse and reduces the computational cost of graph convolution. Based on the ST-graph, we can obtain the corresponding adjacency matrix $\mathbf{A}_1$ and stack multiple graph convolutional layers of Eq. (5) to construct the ST-GCN. The ST-GCN then generates refined feature vectors $\{\hat{\mathbf{Z}}_i\}_{i=t-1}^{t-T}$ for each node of the spatial-temporal graph, $\hat{\mathbf{Z}}_i \in \mathrm{R}^{D_2 \times M_z}$. To reduce the computational burden of the following layers, we then aggregate the features along the temporal axis to produce the compact ST-feature $\mathbf{V}_1 \in \mathrm{R}^{D_2 \times M_z}$:

$$\mathbf{V}_1 = \mathrm{MaxPooling}_T([\hat{\mathbf{Z}}_{t-T}, \hat{\mathbf{Z}}_{t-T+1}, ..., \hat{\mathbf{Z}}_{t-1}]), \quad (6)$$

where the $\mathrm{MaxPooling}_T$ operation is applied with a time range T. $\mathbf{V}_1$ is then taken as input of the CT-GCN.

3.3. Target Feature Adaption via CT-GCN

Our framework not only models the spatial-temporal structure between target exemplars, but also incorporates the context information of current search images to guide the adaptive feature learning. To take full advantage of the context information, we integrate a graph learning model to our framework as shown in Figure 3, which generates an adaptive graph structure for guiding the CT-GCN. As shown in the bottom of Figure 3, taking the current search image x_t as input, the shared ConvNet produces the instance embedding $\mathbf{X}_t \in \mathrm{R}^{D_1 \times M_x}$. To get the global information of the search image, we utilize a convolutional layer followed by a max pooling layer to generate a global feature $\mathbf{x}_t$ with the size of $D_1 \times 1$. Here, the convolutional layer has D_2 filters with a kernel size of 3×3 and stride 1, and the size of the pooling layer is M_x. Taking the global feature $\mathbf{x}_t$ as the current context information, we use a deconvolutional layer to get an enlarged feature $\hat{\mathbf{X}}_t$, which is the same size as the ST-feature $\mathbf{V}_1$. $\hat{\mathbf{X}}_t$ is then fused with $\mathbf{V}_1$ by element-wise addition as follows:

$$\mathbf{V}_x = \mathbf{V}_1 + \hat{\mathbf{X}}_t, \quad (7)$$

where $\mathbf{V}_x$ considers both the spatial-temporal feature of target object and the context information of current frame. To perform graph learning for robust feature adaption, we use $\mathbf{V}_x$ to generate an adaptive graph $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2)$ with the adjacency matrix $\mathbf{A}_2$ defined as:

$$\mathbf{A}_2^{ji} = \frac{\exp(g(\mathbf{V}_{x,i})^\top h(\mathbf{V}_{x,j}))}{\sum_{i=1}^{M_z} \exp(g(\mathbf{V}_{x,i})^\top h(\mathbf{V}_{x,j}))}, \quad (8)$$

where $\mathbf{V}_{x,i}$ is the i^{th} column vector of $\mathbf{V}_x$, $g(\cdot)$ and $h(\cdot)$ are two 1×1 convolutional layers with D_1 filters.

With the learned graph, we are able to construct the CT-GCN via Eq. (5), which takes the ST-feature as input and produces the adaptive feature $\mathbf{V}_2 \in \mathrm{R}^{D_1 \times M_z}$ for target localization in tracking process.

3.4. The Proposed Tracking Algorithm

Network Structure. As shown in Figure 3, we use the modified AlexNet [34] pre-trained on ImageNet [54] as the shared ConvNet. The weights of the first three conv layers are fixed and only the last two conv layers are fine-tuned. We also add an additional 3×3 conv layer to reduce the output channel dimensionality to $D_1 = 256$. The part numbers of the exemplar and search image embedding are $M_z = 6 \times 6 = 36$ and $M_x = 22 \times 22 = 484$, respectively. For the ST-GCN, we adopt 2 graph convolutional layers with the output channel dimensionality of 512 and 256 (D_2). The CT-GCN also has 2 graph convolutional layers with 384 and 256 channel numbers. Following [69], we apply the LeakyReLU as the activation function for both ST-GCN and CT-GCN.

Offline Training. We use videos from the video object detection dataset of the ImageNet Large Scale Visual Recognition Challenge (ILSVRC2015) [54] as training data. The dataset contains almost 4500 videos with a total of more than one million annotated frames. In each video snippet of an object, we collect each training sample of $T + 1$ frames within the nearest 100 frames. We use the former T frames as exemplar images and take the last one as the search image. We adopt the ADAM optimizer with learning rate of

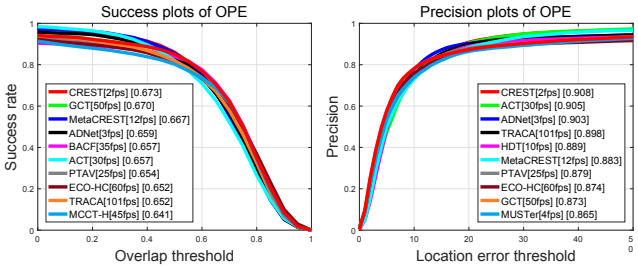
0.005 and set weight decay to $5e-5$. The model is trained for 50 epochs with a batch size of 24.

Tracking Inference. For the tracker initialization, we duplicate the first frame T times as the exemplar images. We set $T = 10$ in our experiments. In the tracking process, we use an interval $\tau = 7$ to update the exemplar images, which enables our method to effectively remember a long range of historical information. Specifically, for every τ frames, the first exemplar image is removed and the new exemplar is added. We use a ratio of 0.4 to smooth the new exemplar with the initial exemplar. The target center can be determined by locating the maximum value in the response map $\mathbf{R}_G$ generated by the cross-correlation layer, as shown in Figure 3. Since different layers in a deep network characterize the target from different perspectives [45, 25], we further use the features from the 5-th conv layer of the shared ConvNet to generate the other response map R_S. The final response map is calculated by balancing $\mathbf{R}_G$ and $\mathbf{R}_S$ with a coefficient γ: $\mathbf{R} = \gamma\mathbf{R}_G + (1-\gamma)\mathbf{R}_S$. γ is set to 0.7. Follow [2], a cosine window is further added to the response map to penalize large displacement.

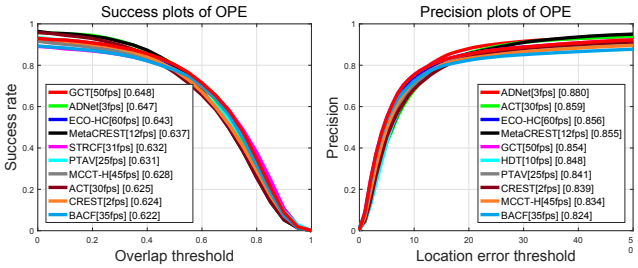
Scale Estimation. To handle scale variations, we follow [2] to search on three scales of the current search image with scale factors of $1.0375^{\{-1,0,1\}}$. We update the scale by linear interpolation with a factor of 0.59 to provide damping. To further speed up the tracker, we only use the response map $\mathbf{R}_S$ to estimate the scales, which also shows favorable performance in the experiments.

Discussion. The proposed GCT consists of both ST-GCN and CT-GCN, which can jointly perform spatial-temporal target appearance modeling and feature adaption with context information in an end-to-end framework. For the ST-GCN, we design a fixed spatial-temporal graph in consideration of two factors. (1) Since the spatial-temporal graph is large with M_zT nodes, fixing the adjacency matrix $\mathbf{A}_1$ is more computationally efficient than fine-tuning it [18]. Note that another graph-based tracking method [8] also adopts a fixed graph for appearance modeling. (2) Although the temporal edges may not connect the same target part in consecutive frames, the message can still be passed between any related parts because the spatial edges in each frame are fully-connected. In addition, ST-GCN has multiple layers which can enlarge the receptive field of each node. For the CT-GCN, we use the search image to provide rich context information such as target object and the surrounding background for guiding the feature adaption[1]. After the end-to-end offline training with large-scale training videos, in the online tracking process, the graph $\mathcal{G}_2$ can be automatically and adaptively produced with different ST-features and context features. The effectiveness of both types of GCN is verified in our experiments.

[1]In siamese learning, the exemplar images also include background information surrounding the target object.



(a) Results for OTB-2013 benchmark [70]



(b) Results for OTB-2015 benchmark [71]

Figure 4. Quantitative results on OTB datasets. Our GCT method performs favorably against the state-of-the-art trackers.

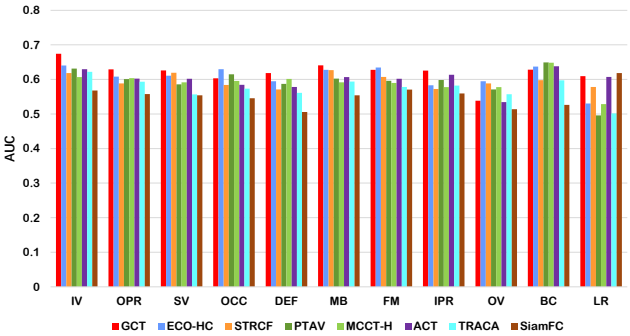


Figure 5. 11 attributes comparison of 7 real-time trackers on OTB-2015 in term of AUC. The proposed GCT method performs favorably against the state-of-the-arts.

4. Experimental Results

We conduct extensive experiments [2] on 4 challenging datasets including the OTB-2013 Object Tracking Benchmark [70] with 50 sequences, its updated version OTB-2015 [71] with 100 sequences, VOT2017 benchmark [33] with 60 videos, and UAV123 benchmark [47] with 123 aerial tracking videos. Our tracker is implemented on TensorFlow. The hardware environment includes an Intel E5-2687 3.0GHz CPU, 256GB RAM and a NVidia 1080Ti GPU.

4.1. Experiments on OTB

Evaluation Protocol. Following the protocol used in the recently published methods [84, 73, 81, 57], we report the results in one-pass evaluation (OPE) [70]. The evaluation is based on two metrics: success plot and precision plot. (1) The success plot illustrates the ratios of successful frames

[2]`http://nlpr-web.ia.ac.cn/mmc/homepage/jygao/gct_cvpr2019.html` (project page of our GCT)

Table 1. Comparison with 6 state-of-the-art trackers on the OTB-2013 and OTB-2015, based on AUC score. Our method provides comparable results against the state-of-the-art trackers.

Method	MDNet [49]	SANet [15]	ECO [9]	CCOT [11]	DSLT [43]	VITAL [58]	GCT (Ours)
OTB-2013	70.8	68.6	70.9	67.2	68.3	**71.0**	67.0
OTB-2015	67.8	**69.2**	69.1	67.1	66.0	68.2	64.8
Speed(FPS)	2.6	1.0	6.0	0.6	5.7	1.5	**49.8**

over the range of thresholds $[0, 1]$, where Area-under-the-curve (AUC) is included in the legend. (2) The precision plot shows the average distance precision along with a range of thresholds, and the average Distance precision (DP) score at 20 pixels for each tracker is reported.

Baseline Methods. We evaluate our GCT method with 29 trackers in the OTB benchmark [70, 71] and other state-of-the-art tracking methods that presented at top conferences and journals, including MetaCREST (ECCV 2018) [50], ACT (ECCV 2018) [4], MCCT-H (CVPR 2018) [64], TRACA (CVPR 2018) [5], STRCF (CVPR 2018) [37], CREST (ICCV 2017) [57], PTAV (ICCV 2017) [14], BACF (ICCV 2017) [16], ECO-HC (CVPR 2017) [9], ACFN (CVPR 2017) [6], ADNet (CVPR 2017) [73], CSR-DCF (CVPR 2017) [44], Staple_CA (CVPR 2017) [48], CFNet (CVPR 2017) [63], SINT (CVPR 2016, only for OTB-2013) [60], Staple (CVPR 2016) [1], HDT (CVPR 2016) [51], SiamFC (ECCVW 2016) [2], SRDCF (ICCV 2015) [10], MUSTer (CVPR 2015) [29], CNN-SVM (ICML 2015) [28], RPT (CVPR 2015) [39], KCF (T-PAMI 2015) [27] and MEEM (ECCV 2014) [75].

Quantitative Evaluation. Figure 4 illustrates the success and precision plots of the overall performance among compared trackers. To make it clear, we only plot the top 10 ranked methods. The proposed GCT approach performs favorably with AUC of $(67.0\%, 64.8\%)$ and DP of $(87.3\%, 85.4\%)$ on the OTB-2013 and OTB-2015, respectively. SINT [60], CFNet [63], and SiamFC [2] are three state-of-the-art siamese based trackers, which provide the results with an AUC score of 63.5%, 61.0%, and 60.7% on OTB-2013, respectively. Compared to them, our method gets an absolute gain of 3.5%, 6.0%, and 6.3%. Another siamese based tracker, DaSiamRPN [86], has the AUC score of 65.9% on OTB-2015, which is slightly better than our method (64.8%). However, DaSiamRPN uses other large-scale datasets for model training, such as COCO Detection dataset [41] and Youtube-BB [53]. This strategy can also be used to further boost the performance of our method. Overall, compared with the state-of-the-arts, the proposed GCT achieves better or comparable results. Note that the DP score of our method is not very significant, which may because of the low resolution of the response map (17×17) and its interpolation process in target localization. This can be improved by training a siamese network with high-resolution response map like [65]. We also compare GCT to the currently topmost non-realtime trackers including MDNet (CVPR2016) [49], SANet (CVPRW2017) [15], ECO (CVPR2017) [9], CCOT (ECCV2016) [11], DSLT (ECCV2018) [43], and VITAL (CVPR2018) [58] . In Table 1, the AUC scores of the algorithms on both benchmarks are presented along with the run-time speed. Our method has comparable performance and achieves a significant speed improvement. Moreover, MDNet, SANet, and VITAL train and test deep models for tracking using videos from the same ALOV/OTB/VOT domain, which is forbidden in VOT challenges due to the overfitting problem [2].

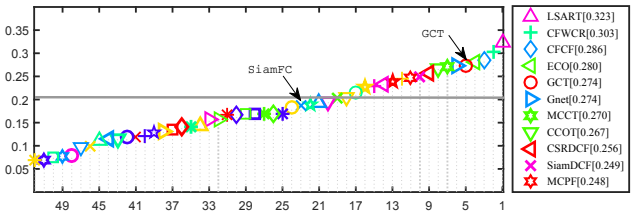


Figure 6. Comparison of EAO scores on VOT2017 challenge. The gray horizontal line denotes the VOT2017 state-of-the-art bound.

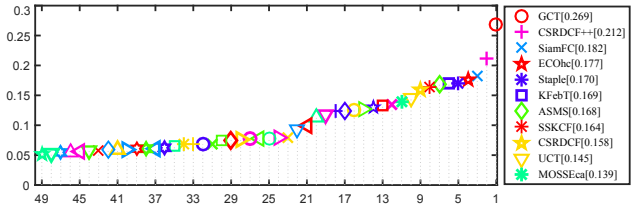


Figure 7. The EAO scores for the real-time experiment on VOT2017 challenge. GCT performs the best.

Attribute-based Evaluation. We further analyze the performance of our GCT tracker under different attributes on OTB-2015 benchmark. Figure 5 shows the comparison of GCT and another seven state-of-the-art real-time trackers. Specifically, our method achieves the best under 6 out of 11 attributes. For the rest five, GCT performs favorably.

4.2. Experiments on VOT2017

We compare our GCT with the state-of-the-art methods on VOT 2017 benchmark [33, 32]. The performance is evaluated by Expected Average Overlap (EAO), which reflects both robustness and accuracy. Figure 6 reports the results of ours against other 51 trackers with respect to the EAO score. As presented in the VOT2017 report [33], trackers whose EAO values exceed 0.203 will be considered as state-of-the-art methods. Our proposed GCT ranks the fifth with the EAO score of 0.274. Figure 7 shows the EAO scores in the real-time experiment of VOT2017. Our tracker achieves the best performance with the EAO score of 0.269 and outperforms other real-time methods by a large margin.

4.3. Experiments on UAV123

Finally, we evaluate the proposed GCT on the recently proposed aerial video dataset, UAV123 [47], which has 123 UAV tracking sequences with more than 110K frames. GCT is compared with all 14 trackers reported in [47] and other real-time state-of-the-art methods including MCCT-H [64], STRCF [37], ECO-HC [9], and Staple [1]. Figure 8 again

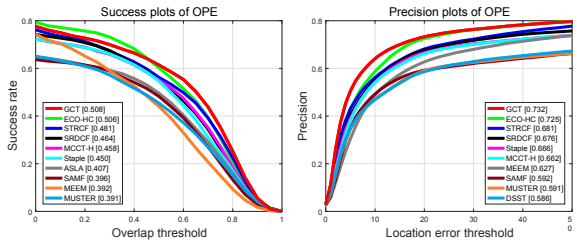


Figure 8. Quantitative results on the UAV123 benchmark [40]. Our proposed GCT method performs favorably.

Table 2. Analysis of our approach on the OTB-2013 and OTB-2015. The impact of progressively integrating one component at the time, from left to right, is displayed.

	SiamFC	$\Longrightarrow$S-GCN	$\Longrightarrow$ST-GCN	$\Longrightarrow$CT-GCN
OTB-2013(%)	60.7	62.5	64.9	67.0
OTB-2015(%)	57.7	60.2	63.5	64.8
FPS(OTB-2015)	76.1	66.7	58.6	49.8

shows that our proposed GCT performs favorably.

4.4. Further Remarks

Component Contribution. To verify the contributions of each component in our framework, we implement and evaluate four variants of our approach on OTB-2013 and OTB-2015 benchmarks. In Table 2, the impact of progressively adding one component, from left to right, is presented. For simplicity, we take the results on OTB-2015 for illustration here. The first is the baseline *SiamFC*[3], which removes the following GCN modules and only uses the response map $\mathbf{R}_S$ for target localization. We then add a spatial GCN (*S-GCN*) on SiamFC and use the fused response map in tracking process. Specifically, S-GCN removes the temporal edges in ST-GCN and sets $T = 1$. S-GCN outperforms SiamFC by an absolute gain of 2.5%, which shows the part-based spatial modeling is useful in visual tracking. Additionally incorporating our proposed *ST-GCN* elevates us to an AUC score of 63.5%, leading to a relative gain of 5.5% compared to S-GCN. The significant result clearly shows the effectiveness of our spatial-temporal appearance modeling. Finally, we add *CT-GCN* to our framework, which obtains a relative gain of 2.0% compared to ST-GCN. Table 2 also shows the impact on the tracker speed achieved by our components. Overall, the proposed GCT with both ST-GCN and CT-GCN achieves the best tracking performance and a favorable run-time speed.

Detailed Analysis of the ST-GCN. To quantitatively analyze different depths of the ST-GCN, we design another two variants, *ST-1L* and *ST-4L*. ST-1L has 1 graph convolutional layers with output channel number of 256. The 4-layer model ST-4L has channel numbers as $512 \rightarrow 1024 \rightarrow 512 \rightarrow 256$. In the left of Figure 9, we do not find much gain by adding more layers above our 2-layer ST-GCN model. To make our tracker efficient, we set the number of layers in ST-GCN to 2. We also explore other graph structures for ST-GCN. As shown in Figure 9, we design two baselines *ST-knn-2L* and *ST-knn-4L*, which adopt an 8-nearest-neighbor graph used in [8] to represent spatial edges. Although ST-knn-4L can achieve similar performance compared to our method, it is less efficient since it needs more graph convolutional layers.

[3]Since this baseline is implement by ourselves, the results are slightly different from the initial SiamFC tracker [2]

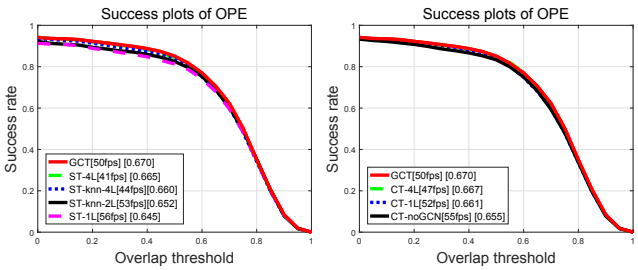


Figure 9. Ablation study of both ST-GCN and CT-GCN on OTB-2013 benchmark.

Detailed Analysis of the CT-GCN. We also evaluate the effect of different numbers of layers in CT-GCN and design similar baselines with them in ST-GCN. Figure 9 (b) shows *CT-1L* gets inferior results while *CT-4L* has lower running speed. In addition, to verify the effectiveness of the CT-GCN, we develop a baseline method, *CT-noGCN*, which removes the graph convolutional layers. CT-noGCN only uses the scores produced by Eq. (8) to generate the adaptive feature via linear combination. We can find that our proposed GCT outperforms it by a relative gain of 2.3%. In fact, GCT can further conduct message passing between related parts based on the learned graph, which is better than the linear combination with the generated scores.

5. Conclusions

In this paper, we propose a graph convolutional tracking framework, which can jointly achieve spatial-temporal target appearance modeling and context-aware adaptive learning for robust target localization in a unified framework. We show that by carefully designing the spatial-temporal GCN and the context GCN, the proposed GCT achieves state-of-the-art results in both accuracy and speed. The encouraging performance is demonstrated in extensive experiments of four challenging benchmarks. In the future, we intend to explore other types of graph neural networks for visual tracking, such as graph embedding and graph attention model. We will also apply our method in other computer vision tasks, *e.g.* multi-object tracking and person re-id.

Acknowledgements

This work is supported in part by the National Natural Science Foundation of China under Grants 61432019, 61572498, 61532009, 61728210, 61721004, 61751211, 61572296, 61720106006 and U1705262, and the Key Research Program of Frontier Sciences, CAS, Grant NO. QYZDJ-SSW-JSC039, the Beijing Natural Science Foundation 4172062, and Youth Innovation Promotion Association CAS 2018166.

References

[1] Luca Bertinetto, Jack Valmadre, Stuart Golodetz, Ondrej Miksik, and Philip H. S. Torr. Staple: Complementary learners for real-time tracking. In *CVPR*, 2016.

[2] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *ECCV Workshops*, 2016.

[3] Adel Bibi, Matthias Mueller, and Bernard Ghanem. Target response adaptation for correlation filter tracking. In *ECCV*, 2016.

[4] Boyu Chen, Dong Wang, Peixia Li, Shuang Wang, and Huchuan Lu. Real-time actor-critictracking. In *ECCV*, 2018.

[5] Jongwon Choi, Hyung Jin Chang, Tobias Fischer, Sangdoo Yun, Kyuewang Lee, Jiyeoup Jeong, Yiannis Demiris, and Jin Young Choi. Context-aware deep feature compression for high-speed visual tracking. In *ECCV*, 2018.

[6] Jongwon Choi, Hyung Jin Chang, Sangdoo Yun, Tobias Fischer, Yiannis Demiris, and Young Choi Jin. Attentional correlation filter network for adaptive visual tracking. In *CVPR*, 2017.

[7] Zhen Cui, Shengtao Xiao, Jiashi Feng, and Shuicheng Yan. Recurrently target-attending tracking. In *CVPR*, 2016.

[8] Zhen Cui, Jian Yang, et al. Spectral filter tracking. *arXiv preprint arXiv:1707.05553*, 2017.

[9] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Eco: Efficient convolution operators for tracking. In *CVPR*, 2017.

[10] Martin Danelljan, Gustav Hager, Fahad Shahbaz Khan, and Michael Felsberg. Learning spatially regularized correlation filters for visual tracking. In *ICCV*, 2015.

[11] Martin Danelljan, Andreas Robinson, Fahad Shahbaz Khan, and Michael Felsberg. Beyond correlation filters: learning continuous convolution operators for visual tracking. In *ECCV*, 2016.

[12] Xingping Dong and Jianbing Shen. Triplet loss in siamese network for object tracking. In *ECCV*, 2018.

[13] Lingyu Duan, Yihang Lou, Shiqi Wang, Wen Gao, and Yong Rui. Ai oriented large-scale video management for smart city: Technologies, standards and beyond. *IEEE MultiMedia*, 2018.

[14] Heng Fan and Haibin Ling. Parallel tracking and verifying: A framework for real-time and high accuracy visual tracking. In *ICCV*, 2017.

[15] Heng Fan and Haibin Ling. Sanet: Structure-aware network for visual tracking. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017.

[16] Hamed Kiani Galoogahi, Ashton Fagg, and Simon Lucey. Learning background-aware correlation filters for visual tracking. In *ICCV*, 2017.

[17] Junyu Gao, Tianzhu Zhang, and Changsheng Xu. A unified personalized video recommendation via dynamic recurrent neural networks. In *ACM MM*, 2017.

[18] Junyu Gao, Tianzhu Zhang, and Changsheng Xu. Watch, think and attend: End-to-end video classification via dynamic knowledge evolution modeling. In *ACM MM*, 2018.

[19] Junyu Gao, Tianzhu Zhang, and Changsheng Xu. I know the relationships: Zero-shot action recognition via two-stream graph convolutional networks and knowledge graphs. In *AAAI*, 2019.

[20] Junyu Gao, Tianzhu Zhang, and Changsheng Xu. Smart: Joint sampling and regression for visual tracking. *IEEE Transactions on Image Processing*, PP(99):1–1, 2019.

[21] Junyu Gao, Tianzhu Zhang, Xiaoshan Yang, and Changsheng Xu. Deep relative tracking. *IEEE Transactions on Image Processing*, 26(4):1845–1858, 2017.

[22] Junyu Gao, Tianzhu Zhang, Xiaoshan Yang, and Changsheng Xu. P2t: Part-to-target tracking via deep regression learning. *IEEE Transactions on Image Processing*, 27(6):3074–3086, 2018.

[23] Qing Guo, Wei Feng, Ce Zhou, Rui Huang, Liang Wan, and Song Wang. Learning dynamic siamese network for visual object tracking. In *ICCV*, 2017.

[24] Junwei Han, Xiang Ji, Xintao Hu, Dajiang Zhu, Kaiming Li, Xi Jiang, Guangbin Cui, Lei Guo, and Tianming Liu. Representing and retrieving video shots in human-centric brain imaging space. *IEEE Transactions on Image Processing*, 22(7):2723–2736, 2013.

[25] Anfeng He, Chong Luo, Xinmei Tian, and Wenjun Zeng. A twofold siamese network for real-time object tracking. In *CVPR*, 2018.

[26] David Held, Sebastian Thrun, and Silvio Savarese. Learning to track at 100 fps with deep regression networks. In *ECCV*, 2016.

[27] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):583–596, 2015.

[28] Seunghoon Hong, Tackgeun You, Suha Kwak, and Bohyung Han. Online tracking by learning discriminative saliency map with convolutional neural network. In *ICML*, 2015.

[29] Zhibin Hong, Zhe Chen, Chaohui Wang, Xue Mei, Danil Prokhorov, and Dacheng Tao. Multi-store tracker (muster): A cognitive psychology inspired approach to object tracking. In *CVPR*, 2015.

[30] Chen Huang, Simon Lucey, and Deva Ramanan. Learning policies for adaptive tracking with deep feature cascades. In *ICCV*, 2017.

[31] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2016.

[32] Matej Kristan, Jiri Matas, Aleš Leonardis, Tomáš Vojíř, Roman Pflugfelder, Gustavo Fernandez, Georg Nebehay, Fatih Porikli, and Luka Čehovin. A novel performance evaluation methodology for single-target trackers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016.

[33] Matej Kristan, Roman Pflugfelder, Ales Leonardis, Jiri Matas, Fatih Porikli, Luka Cehovin, Georg Nebehay, Gustavo Fernandez, Tomas Vojir, Adam Gatt, et al. The visual object tracking vot2017 challenge results. In *ICCV*, 2017.

[34] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[35] Chung-Wei Lee, Wei Fang, Chih-Kuan Yeh, and Yu-Chiang Frank Wang. Multi-label zero-shot learning with structured knowledge graphs. In *CVPR*, 2018.

[36] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *CVPR*, 2018.

[37] Feng Li, Cheng Tian, Wangmeng Zuo, Lei Zhang, and Ming-Hsuan Yang. Learning spatial-temporal regularized correlation filters for visual tracking. In *CVPR*, 2018.

[38] Yang Li and Jianke Zhu. A scale adaptive kernel correlation filter tracker with feature integration. In *ECCV*, 2014.

[39] Yang Li, Jianke Zhu, and Steven CH Hoi. Reliable patch trackers: Robust visual tracking by exploiting reliable patches. In *CVPR*, 2015.

[40] Pengpeng Liang, Erik Blasch, and Haibin Ling. Encoding color information for visual tracking: algorithms and benchmark. *IEEE Transactions on Image Processing*, 24(12):5630–5644, 2015.

[41] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollar, and Larry Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[42] Si Liu, Zhen Wei, Yao Sun, Xinyu Ou, Junyu Lin, Bin Liu, and Ming-Hsuan Yang. Composing semantic collage for image retargeting. *IEEE Transactions on Image Processing*, 27(10):5032–5043, 2018.

[43] Xiankai Lu, Chao Ma, Bingbing Ni, Xiaokang Yang, Ian Reid, and Ming-Hsuan Yang. Deep regression tracking with shrinkage loss. In *ECCV*, 2018.

[44] Alan Lukei, Tom Voj, Luka ehovin, Ji Matas, and Matej Kristan. Discriminative correlation filter with channel and spatial reliability. In *CVPR*, 2017.

[45] Chao Ma, Jia-Bin Huang, Xiaokang Yang, and Ming-Hsuan Yang. Hierarchical convolutional features for visual tracking. In *ICCV*, 2015.

[46] Kenneth Marino, Ruslan Salakhutdinov, and Abhinav Gupta. The more you know: Using knowledge graphs for image classification. In *CVPR*, 2017.

[47] Matthias Mueller, Neil Smith, and Bernard Ghanem. A benchmark and simulator for uav tracking. In *ECCV*, 2016.

[48] Matthias Mueller, Neil Smith, and Bernard Ghanem. Context-aware correlation filter tracking. In *CVPR*, 2017.

[49] Hyeonseob Nam and Bohyung Han. Learning multi-domain convolutional neural networks for visual tracking. In *CVPR*, 2016.

[50] Eunbyung Park and Alexander C Berg. Meta-tracker: Fast and robust online adaptation for visual object trackers. In *ECCV*, 2018.

[51] Yuankai Qi, Shengping Zhang, Lei Qin, Hongxun Yao, Qingming Huang, and Jongwoo Lim Ming-Hsuan Yang. Hedged deep tracking. In *CVPR*, 2016.

[52] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *ICCV*, 2017.

[53] Esteban Real, Jonathon Shlens, Stefano Mazzocchi, Xin Pan, and Vincent Vanhoucke. Youtube-boundingboxes: A large high-precision human-annotated data set for object detection in video. In *CVPR*, 2017.

[54] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[55] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009.

[56] Yantao Shen, Hongsheng Li, Shuai Yi, Dapeng Chen, and Xiaogang Wang. Person re-identification with deep similarity-guided graph neural network. In *ECCV*, 2018.

[57] Yibing Song, Chao Ma, Lijun Gong, Jiawei Zhang, Rynson Lau, and Ming-Hsuan Yang. Crest: Convolutional residual learning for visual tracking. In *ICCV*, 2017.

[58] Yibing Song, Chao Ma, Xiaohe Wu, Lijun Gong, Linchao Bao, Wangmeng Zuo, Chunhua Shen, Rynson Lau, and Ming-Hsuan Yang. Vital: Visual tracking via adversarial learning. In *ECCV*, 2018.

[59] Chong Sun, Dong Wang, and Huchuan Lu. Occlusion-aware fragment-based tracking with spatial-temporal consistency. *IEEE Transactions on Image Processing*, 25(8):3814–3825, 2016.

[60] Ran Tao, Efstratios Gavves, and Arnold W M Smeulders. Siamese instance search for tracking. In *CVPR*, 2016.

[61] Zhu Teng, Junliang Xing, Qiang Wang, Congyan Lang, Songhe Feng, Yi Jin, et al. Robust object tracking based on temporal and spatial deep networks. In *ICCV*, 2017.

[62] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015.

[63] Jack Valmadre, Luca Bertinetto, Joo F Henriques, Andrea Vedaldi, and Philip H. S Torr. End-to-end representation learning for correlation filter based tracking. In *CVPR*, 2017.

[64] Ning Wang, Wengang Zhou, Qi Tian, Richang Hong, Meng Wang, and Houqiang Li. Multi-cue correlation filters for robust visual tracking. In *CVPR*, 2018.

[65] Qiang Wang, Jin Gao, Junliang Xing, Mengdan Zhang, and Weiming Hu. Dcfnet: Discriminant correlation filters network for visual tracking. *arXiv preprint arXiv:1704.04057*, 2017.

[66] Qiang Wang, Zhu Teng, Junliang Xing, Jin Gao, Weiming Hu, and Stephen Maybank. Learning attentions: residual attentional siamese network for high performance online visual tracking. In *CVPR*, 2018.

[67] Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. In *ECCV*, 2018.

[68] Xiao Wang, Chenglong Li, Bin Luo, and Jin Tang. Sint++: Robust visual tracking via adversarial positive instance generation. In *CVPR*, 2018.

[69] Xiaolong Wang, Yufei Ye, and Abhinav Gupta. Zero-shot recognition via semantic embeddings and knowledge graphs. In *CVPR*, 2018.

[70] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Online object tracking: A benchmark. In *CVPR*, 2013.

[71] Yi Wu, Jongwoo Lim, and Ming Hsuan Yang. Object tracking benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37:1834–1848, 2015.

[72] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. *arXiv preprint arXiv:1801.07455*, 2018.

[73] Sangdoo Yun, Jongwon Choi, Youngjoon Yoo, Kimin Yun, and Young Choi Jin. Action-decision networks for visual tracking with deep reinforcement learning. In *CVPR*, 2017.

[74] Dingwen Zhang, Deyu Meng, and Junwei Han. Co-saliency detection via a self-paced multiple-instance learning framework. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(5):865–878, 2017.

[75] Jianming Zhang, Shugao Ma, and Stan Sclaroff. MEEM: robust tracking via multiple experts using entropy minimization. In *ECCV*, 2014.

[76] Kaihua Zhang, Lei Zhang, Qingshan Liu, David Zhang, and Ming-Hsuan Yang. Fast visual tracking via dense spatio-temporal context learning. In *ECCV*, 2014.

[77] Tianzhu Zhang, Bernard Ghanem, Si Liu, and Narendra Ahuja. Robust visual tracking via multi-task sparse learning. In *CVPR*, 2012.

[78] Tianzhu Zhang, Bernard Ghanem, Si Liu, and Narendra Ahuja. Robust visual tracking via structured multi-task sparse learning. *International Journal of Computer Vision*, 101(2):367–383, 2013.

[79] Tianzhu Zhang, Si Liu, Narendra Ahuja, Ming-Hsuan Yang, and Bernard Ghanem. Robust Visual Tracking via Consistent Low-Rank Sparse Learning. *International Journal of Computer Vision*, 111(2):171–190, 2015.

[80] Tianzhu Zhang, Si Liu, Changsheng Xu, Bin Liu, and Ming-Hsuan Yang. Correlation particle filter for visual tracking. *IEEE Transactions on Image Processing*, 27(6):2676–2687, 2018.

[81] Tianzhu Zhang, Changsheng Xu, and Ming-Hsuan Yang. Multi-task correlation particle filter for robust object tracking. In *CVPR*, 2017.

[82] Tianzhu Zhang, Changsheng Xu, and Ming-Hsuan Yang. Learning multi-task correlation particle filters for visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2):365–378, 2019.

[83] Tianzhu Zhang, Changsheng Xu, and Ming-Hsuan Yang. Robust structural sparse tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2):473–486, 2019.

[84] Yunhua Zhang, Lijun Wang, Jinqing Qi, Dong Wang, Mengyang Feng, and Huchuan Lu. Structured siamese network for real-time visual tracking. In *ECCV*, 2018.

[85] Yuhui Zheng, Le Sun, Shunfeng Wang, Jianwei Zhang, and Jifeng Ning. Spatially regularized structural support vector machine for robust visual tracking. *IEEE transactions on neural networks and learning systems*, (99):1–11, 2018.

[86] Zheng Zhu, Qiang Wang, Bo Li, Wei Wu, Junjie Yan, and Weiming Hu. Distractor-aware siamese networks for visual object tracking. In *ECCV*, 2018.

[87] Zheng Zhu, Wei Wu, Wei Zou, and Junjie Yan. End-to-end flow correlation tracking with spatial-temporal attention. In *CVPR*, 2018.