

# Spectral Filter Tracking

Zhen Cui<sup>\*†</sup>, *Member, IEEE*, Youyi Cai<sup>†</sup>, Wenming Zheng, *Member, IEEE*, Jian Yang, *Member, IEEE*,

**Abstract**—Visual object tracking is a challenging computer vision task with numerous real-world applications. Here we propose a simple but efficient Spectral Filter Tracking (SFT) method. To characterize rotational and translation invariance of tracking targets, the candidate image region is modeled as a pixelwise grid graph. Instead of the conventional graph matching, we convert the tracking into a plain least square regression problem to estimate the best center coordinate of the target. But different from the holistic regression of correlation filter based methods, SFT can operate on localized surrounding regions of each pixel (i.e., vertex) by using spectral graph filters, which thus is more robust to resist local variations and cluttered background. To bypass the eigenvalue decomposition problem of the graph Laplacian matrix  $\mathcal{L}$ , we parameterize spectral graph filters as the polynomial of  $\mathcal{L}$  by spectral graph theory, in which  $\mathcal{L}^k$  exactly encodes a  $k$ -hop local neighborhood of each vertex. Finally, the filter parameters (i.e., polynomial coefficients) as well as feature projecting functions are jointly integrated into the regression model.

SFT can simply boil down to only a few line codes, but surprisingly it beats the correlation filter based model with the same feature input, and achieves the current best performance on the dataset [36] under the same feature extraction strategy (i.e., the existing VGG-Net model [32]). The code will be fully released in our website soon.

## I. INTRODUCTION

Visual object tracking is a fundamental task in computer vision, due to its wide applications to video surveillance, traffic monitoring, and augmented reality, etc. Despite significant advance that has been achieved in visual tracking over the past few decades, this task remains very challenging because of unpredictable appearance variations including partial occlusion, geometric deformation, illumination change, background clutter, fast motion, etc.

The typical visual tracking starts with an initial bounding box of an object at the first frame, and then sequentially predicts the locations of the target in the next frames. To attain robust tracking, numbers of tracking methods have sprung up. Among the existing tracking works, those part-based methods [1], [6], [26] have drawn increasing attention due to their robustness to local occlusion or appearance variations. They usually partition the object target (or the candidate region) into

several parts and extract some useful cues from these parts. In the part-based methods, the topology structures (e.g., tree or graph) [22], [5] are often used to characterize the relationship of parts, and then some voting or matching strategies [1], [38] are taken to find those reliable parts. In principle the part-based model is robust to resist partial occlusions and local appearance variations, but in practice it is difficult for accurate part partition, even though several methods [1], [21] have been developed. More recently the tracking-by-segmentation methods [29], [34], [18], [35] attempt to accurately annotate foreground and background regions based on the superpixel techniques. But the segmentation is usually time-consuming, and its results heavily influence on the tracking performance.

In contrast, the holistic tracking methods are more popular, especially the recent correlation filter (CF) based model. Due to high-efficiency and excellent robustness, the correlation filter (CF) based model has aroused wide attention [4], [9], [12], [23], [27], [17] in the field of visual tracking. CF based methods attempt to learn a group of discriminative correlation filters that can produce correlation peaks for the tracking targets while suppressing their responses on background regions. To speed up the tracker, the original convolutional filters are transformed into frequency domain, and then learnt by scanning the candidate regions on a circular sliding window. As a holistic model, the CF based methods identically treat the whole candidate region, so those cluttered background might affect the trackers and degrade the tracking performance. To address this problem, some regularization methods of correlation filters [11], [8] are proposed to spatially suppress background regions. But the holistic CF based methods is flexible enough not to resist local appearance variations like those part-based methods.

In this paper, we propose a simple but efficient Spectral Filter Tracking (SFT) method. To model rotational and translation invariance of the tracking targets, we construct a pixel-level grid graph for a candidate image region, which thus avoids any operations of part partition or superpixel segmentation. Moreover, the vertexes of the graph enclose the center of the tracking target, so we only need discover the best matching vertex from this graph. But instead of the conventional graph matching, we convert it into a plain least square regression problem to estimate the best center coordinate of the target.

But different from the holistic regression of CF based methods, we regress the tracking model on multiple localized regions for each pixel (i.e., vertex). To extract the local regions associated with each vertex, we use those spectral filters of graph. To solve those spectral filters, we need to decompose the graph Laplacian matrix. To bypass the eigenvalue decomposition problem, the spectral filters are parameterized as the polynomial of graph Laplacian matrix, in which the

Zhen Cui is with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China.  
E-mail: zhen.cui@njjust.edu.cn,

Youyi Cai and Wenming Zheng are with the Key Laboratory of Child Development and Learning Science of Ministry of Education, Research Center for Learning Science, Southeast University, Nanjing, Jiangsu 210096, China.  
E-mail: yy\_cai@seu.edu.cn, wenming\_zheng@seu.edu.cn.,

Jian Yang is with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China.  
E-mail: csjyang@njjust.edu.cn. Asterisk indicates corresponding author.

$k$ -th term of Laplacian matrix exactly defines a  $k$ -localized spatial region. Consequently spectral filtering on graph is approximately equal to the operating on graph Laplacian matrix. By using the terms of the Laplacian polynomial as the filter bases, we jointly learn the parameters (*i.e.*, polynomial coefficients) as well as feature projection by feeding responses of filter bases into the regression model.

Finally, the proposed tracker SFT can simply boil down to only a few line codes, but surprisingly the experimental results on the dataset [36] show that, the SFT beats the CF based model under the condition of the same input feature, surpasses the recent CF methods [10], [28] on the localization accuracy, and achieves the current best performance under the same feature extraction strategy (*i.e.*, the existing VGG-Net model [32]).

## II. RELATED WORK

Video object tracking has been extensively studied over the past decades. They usually fall into two categories: generative model [3], [25], [39], [30] and discriminative model [2], [15], [17]. Generative methods search for the best matching regions of the tracked target. Discriminative methods learn a classification model to distinguish the target from the backgrounds. Below we briefly introduce the main related work, including the correlation filter based methods and the part based methods.

Recently the correlation filter based discriminative model has aroused wide attention in the field of visual tracking. After Minimum Output Sum of Squared Error (MOSSE) [4] filter was proposed, numerous correlation filter methods start flowing in the field of computer vision [7]. Henriques *et al.* [16] used the kernel trick, and Danelljan *et al.* [12] represented the inputs with color attributes. To handle the target scale problem, SAMF [23], DSST [9] and an improved KCF [17] were proposed subsequently and achieved state-of-the-art performance. With more related methods developed [27], [26], [24], correlation filter based trackers have demonstrated their robustness. Especially, by employing high-level convolutional features as the inputs, correlation filter based methods [10], [28] achieved the current best performance on the public visual tracking dataset [36]. As a holistic filtering model, correlation filter based methods easily absorb those clutter information of background region while making full use of the background information. To reduce the influence of clutter background, some regularization methods [11], [8] were proposed to suppress the response of background region by weighting correlation filters. Although the regularization strategy has demonstrated the initial success on suppressing the background, it is still lack of an intrinsic revision for the holistic model. Different from these correlation filter based methods, we directly perform locally filtering on pixel-level graph structure.

In contrast to correlation filter based methods, the part-based methods [1], [30], [6], [26] seems be more silent recently. They usually partition the object target into several parts, and then attempt to discover some useful cues from the reliable parts. Adam *et al.* [1] partitioned the object into several fragments, and then employed the voting strategy to decide the target

position. Jia *et al.* [19] selected the closest candidate patches of the next frame by using  $l_1$  sparsity. Kwon *et al.* [21] modeled local patches in topology structure in order to find reliable parts. Zhang *et al.* [38] performed part matching among multiple frames. Liu *et al.* [26] learned one response function on each part, and integrated all response maps to decide the final tracking confidence. Besides, the topology structure is used to model the relationship of the parts, *e.g.*, tree structure [22] or graph structure on superpixels [5]. In principle the part-based model is a robust solution to resist partial occlusion and local appearance variations. But in practice it is difficult to accurately define the partition of local parts, even though several strategies [1], [21] have been developed. To address this problem, the tracking-by-segmentation methods [29], [34], [18], [35] attempt to accurately annotate foreground and background regions by using the superpixel segmentation technique. But the tracking performance heavily depends on the segmentation results, and the superpixel segmentation is rather time-consuming. Different from these part-based methods, our proposed method performs local spectral filtering on the pixel-grid graph structure and then convert the target localization as a simple regression problem.

## III. SPECTRAL FILTER TRACKER

### A. Overview

An overview on the SFT flowchart is shown in Fig. 1. Given a video frame, we first determine a small candidate region around the bounding box localized from the previous frame, considering the motions of targets in continuous video frames are usually subtle. To enhance the discriminability, we can represent the candidate region with hand-crafted descriptors (*e.g.*, HOG [13]) or convolutional features [32]. Thereafter we can obtain multi-channel features, where each spatial pixel position is associated with a multi-channel feature vector. To reduce the effect of local appearance variations in the tracking, we model the candidate region as a pixelwise grid graph (Section III-B), which has rotation-invariant and shifting-invariant property. In the graph, one spatial pixel is regarded as one vertex of the graph, and the edges connect those spatial adjacent vertexes. This problem becomes the conventional graph matching. But generally the solution of graph matching is rather complex, which might involve the integer programming.

To bypass graph matching, we use spectral graph theory to analyze graph structure. Instead of the holistic filtering in CF based trackers, we perform local filtering on the graph structure (Section III-C). By using spectral graph filters, we can derive out the responses on localized graph regions for each vertex. But it involves eigenvalue decomposition of graph Laplacian matrix. To avoid this operation, we parameterize spectral graph filters as a polynomial of Laplacian matrix. Each entry of the polynomial actually plays the role of localized filtering on graph. It means that the polynomial terms enclose different scale spectral graph filters. By using the polynomial terms as the basic filters, we can obtain the corresponding multi-scale features for each vertex, which well-model local information of graph. For each vertex, by concatenating its multi-scale responses to form the final representation, finally we feed the final representation into the

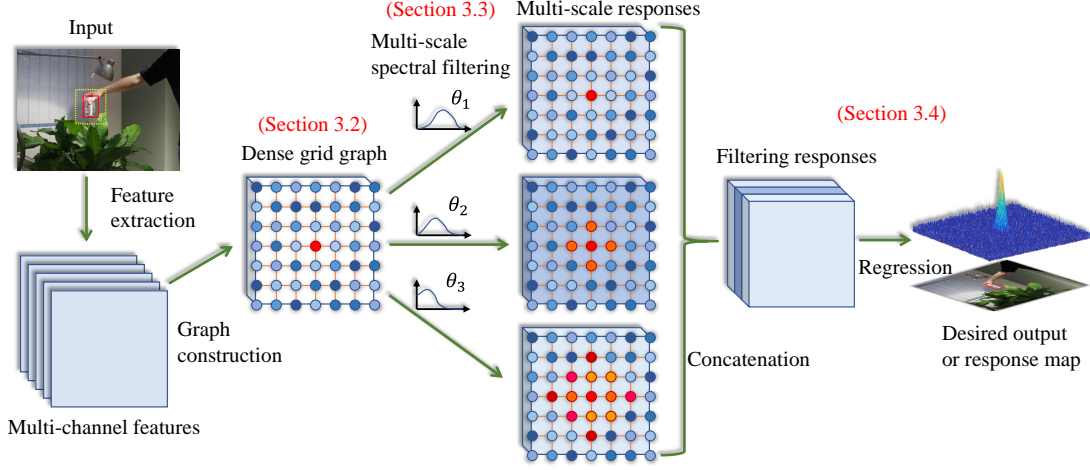


Fig. 1: The flowchart of spectral filter tracking. More details can be found in Section III-A.

regression model to jointly solving those filter parameters (*i.e.*, polynomial coefficients) and feature projecting functions (Section III-D).

### B. The Representation of Graph

We model the pixelwise spatial grid structure as an undirected weighted graph. The weighted graph  $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{W}\}$  consists of a set of vertices  $\mathcal{V}$  ( $|\mathcal{V}| = N$ ) and a set of edges  $\mathcal{E}$ . The adjacency matrix  $\mathbf{W}$  assigns positive values to those connected edges. Besides, each vertex is associated with a signal, *i.e.*, here a multi-channel feature vector extracted from its coordinate position. Formally, the feature extraction function  $f: \mathcal{V} \rightarrow \mathbb{R}^d$  defines the signals of vertexes, where  $d$  is the feature dimension.

In spectral graph theory, a crucial operator is the graph Laplacian operator  $\mathcal{L}$ . The operator is defined as  $\mathcal{L} = \mathbf{D} - \mathbf{W}$ , where  $\mathbf{D} \in \mathbb{R}^{N \times N}$  is the diagonal degree matrix with  $D_{ii} = \sum_j W_{ij}$ . An popular option is the normalized graph Laplacian, where each weight  $W_{ij}$  is multiplied by a factor  $\frac{1}{\sqrt{D_{ii}D_{jj}}}$ , *i.e.*,

$$\mathcal{L}^{norm} = \mathbf{D}^{-\frac{1}{2}} \mathcal{L} \mathbf{D}^{-\frac{1}{2}} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}} \mathbf{W} \mathbf{D}^{-\frac{1}{2}}, \quad (1)$$

where  $\mathbf{I}$  is the identity matrix. Unless otherwise specified, the Laplacian matrix used below is the normalized version.

As a real symmetric matrix, the graph Laplacian  $\mathcal{L}$  has a complete set of orthonormal eigenvectors. The eigenvectors  $\{\mathbf{u}_l\}$  satisfy  $\mathcal{L}\mathbf{u}_l = \lambda_l \mathbf{u}_l$  for  $l = 1, 2, \dots, N$ , where  $\{\lambda_l\}$  are nonnegative real eigenvalues. We assume all eigenvalues are ordered as  $0 = \lambda_1 < \lambda_2 \leq \lambda_3 \leq \dots \leq \lambda_N = \lambda_{max}$ . In matrix expression, the Laplacian matrix is decomposed into  $\mathcal{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$ , where  $\mathbf{\Lambda} = \text{diag}([\lambda_1, \lambda_2, \dots, \lambda_N])$ . Analogous to the classic Fourier transform, the graph Fourier transform of a signal  $\mathbf{x}$  in spatial domain can be defined as  $\hat{\mathbf{x}} = \mathbf{U}^\top \mathbf{x}$  [31], where  $\hat{\mathbf{x}}$  is the produced frequency signal. The corresponding inverse Fourier transform is  $\mathbf{x} = \mathbf{U}\hat{\mathbf{x}}$ .

### C. The Construction of Local Spectral Filters

Suppose  $g(\cdot)$  is a filter function of the graph  $\mathcal{L}$ , we can define the frequency filtering on the input signal  $\mathbf{x}$  as  $\hat{z}(\lambda_l) = \hat{x}(\lambda_l)\hat{g}(\lambda_l)$ , or the inverse graph Fourier transform,

$$z(i) = \sum_{l=1}^N \hat{x}(\lambda_l) \hat{g}(\lambda_l) \hat{u}_l(i), \quad (2)$$

where  $\hat{z}(\lambda_l), \hat{x}(\lambda_l), \hat{g}(\lambda_l)$  are the Fourier coefficients corresponding to the spectrum  $\lambda_l$ . By using matrix notation, the signal  $\mathbf{x}$  is filtered as

$$\mathbf{z} = \hat{g}(\mathcal{L})\mathbf{x} = \mathbf{U} \begin{bmatrix} \hat{g}(\lambda_1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \hat{g}(\lambda_N) \end{bmatrix} \mathbf{U}^\top \mathbf{x}. \quad (3)$$

Given the input  $\mathbf{x}$  and the output  $\mathbf{z}$ , we need to solve the filter function  $g(\cdot)$  in Eqn. (3), which requires eigenvalue decomposition. To reduce computation cost, a low order polynomial may be used to approximate  $\hat{g}(\cdot)$  in the frequency domain. Here we use the Chebyshev expansion of  $K$  order [14], which is defined by the recurrent relation  $T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$  with  $T_0 = 1$  and  $T_1 = x$ . In the appropriate Sobolev space, the set of Chebyshev polynomials form an orthonormal basis, so any one function in the space  $x \in [-1, 1]$  may be expressed via the expansion:  $f(x) = \sum_{k=0}^{\infty} a_k T_k(x)$ .

To make the eigenvalues  $\{\lambda_l\}$  of the Laplacian matrix  $\mathcal{L}$  fall in  $[-1, 1]$ , we may scale and shift them as  $\tilde{\lambda}_l = \frac{2}{\lambda_{max}} \lambda_l - 1$ , and then employ the Chebyshev polynomials on  $\{\tilde{\lambda}_l\}$ . If we consider a linear combination of the polynomial components, the  $K$ -order filter can be written as,

$$\hat{g}(\lambda_l) = \sum_{k=0}^{K-1} \theta_k T_k(\tilde{\lambda}_l), \quad (4)$$

where  $\theta \in \mathbb{R}^K$  is a parameter vector of the polynomial coefficients, and  $K$  is the order of the polynomial. By putting

Eqn. (4) into Eqn. (3), we can have

$$\mathbf{z} = \mathbf{U} \begin{bmatrix} \sum_{k=0}^{K-1} \theta_k T_k(\tilde{\lambda}_l) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sum_{k=0}^{K-1} \theta_k T_k(\tilde{\lambda}_l) \end{bmatrix} \mathbf{U}^\top \mathbf{x}$$

$$= \sum_{k=0}^{K-1} \theta_k \mathbf{U} \begin{bmatrix} T_k(\tilde{\lambda}_l) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & T_k(\tilde{\lambda}_l) \end{bmatrix} \mathbf{U}^\top \mathbf{x} \quad (5)$$

$$= \sum_{k=0}^{K-1} \theta_k T_k(\mathbf{U} \begin{bmatrix} \tilde{\lambda}_l & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \tilde{\lambda}_l \end{bmatrix} \mathbf{U}^\top) \mathbf{x} \quad (6)$$

$$= \sum_{k=0}^{K-1} \theta_k T_k\left(\frac{2}{\lambda_{max}} \mathcal{L} - \mathbf{I}\right) \mathbf{x}. \quad (7)$$

From Eqn. (6) to Eqn. (7), we use the spectral decomposition of Laplacian matrix,  $\mathcal{L} = \mathbf{U} \text{diag}([\lambda_1, \dots, \lambda_N]) \mathbf{U}^\top$ . From Eqn. (5) to Eqn. (6), we utilize the basic calculation on the filter function, *i.e.*,

$$\mathcal{L}^k = \mathbf{U} \text{diag}([\lambda_1^k, \dots, \lambda_N^k]) \mathbf{U}^\top$$

$$= (\mathbf{U} \text{diag}([\lambda_1, \dots, \lambda_N]) \mathbf{U}^\top)^k. \quad (8)$$

According to graph theory,  $\mathcal{L}^k$  encodes a  $k$ -hop local neighborhood of each vertex. Consequently, the  $K$ -order polynomial in Eqn. (7) is a exactly  $K$ -localized filter function on the Laplacian graph. To obtain the local filtering responses on graph, thus we only need operate the Laplacian matrix  $\mathcal{L}$ . That means, each entry of the polynomial can be regarded as the filter bases, and  $\theta$  is the parameters to be solved.

#### D. The Prediction of the Tracking Target

In the visual tracking, we need predict the centers of the tracking target. Similar to those CF based methods, we regress a peak map  $\mathbf{y} \in \mathbb{R}^{N \times 1}$  from the multi-channel features  $\mathbf{X} \in \mathbb{R}^{N \times d}$ , where  $N$  is the number of pixels (*i.e.*, vertexes) within the candidate region, each row of  $\mathbf{X}$  corresponds to a signal of one vertex. Now we denote  $\tilde{\mathcal{L}} = \frac{2}{\lambda_{max}} \mathcal{L} - \mathbf{I}$ . Then the filter bases defined in the polynomial of Eqn. (7) become  $\{T_0(\tilde{\mathcal{L}}), T_1(\tilde{\mathcal{L}}), \dots, T_{K-1}(\tilde{\mathcal{L}})\}$ , where the  $k$ -th filter basis only relates to the  $k$ -hop neighbor vertexes. Given a signal  $\mathbf{x}$  and the filter parameter  $\theta = [\theta_0, \theta_2, \dots, \theta_{K-1}]$ , we can obtain the local filtering response  $\mathbf{z}$  if employing a linear combination of  $K$  filter bases. We use the  $k$  filter bases to filtering the graph, and then combine the learning of filter parameters and feature projecting function into a least square regression model,

$$\arg \min_{\mathbf{w}} \|\mathcal{F}(\mathbf{X}) \mathbf{w} - \mathbf{y}\|^2 + \gamma \|\mathbf{w}\|^2, \quad (9)$$

where  $\gamma$  is the balance parameter, and  $\mathcal{F}(\mathbf{X})$  concatenates the responses of  $K$  filter bases in feature dimensionality,

$$\mathcal{F}(\mathbf{X}) = [T_0(\tilde{\mathcal{L}}) \mathbf{X}, T_1(\tilde{\mathcal{L}}) \mathbf{X}, \dots, T_{K-1}(\tilde{\mathcal{L}}) \mathbf{X}]. \quad (10)$$

Thus the tracking model can be easily solved as

$$\mathbf{w} = (\mathcal{F}(\mathbf{X})^\top \mathcal{F}(\mathbf{X}))^{-1} \mathcal{F}(\mathbf{X})^\top \mathbf{y}. \quad (11)$$

---

#### Algorithm 1 Spectral Filter Tracking Algorithm

---

**Input:** A video sequence with initial target position at the first frame; The regression map  $\mathbf{y}$  (Gaussian-shape).

**Output:** The coordinates of the tracking target.

- 1: Initialize the Laplacian matrix  $\mathcal{L}$ .
  - 2: **repeat**
  - 3:   **if**  $t > 1$  **then**
  - 4:     Extract feature  $\mathbf{X}$  from the candidate region.
  - 5:     Compute  $K$ -order responses  $\mathcal{F}(\mathbf{X})$  in Eqn. (10).
  - 6:     Compute detection score  $\tilde{\mathbf{y}} = \mathcal{F}(\mathbf{X}) \mathbf{w}$  Eqn. (11).
  - 7:     Find the target center with maximum score.
  - 8:   **end if**
  - 9:   Extract feature  $\mathbf{X}$  at the current target location.
  - 10:   Compute  $K$ -order responses  $\mathcal{F}(\mathbf{X})$  in Eqn. (10).
  - 11:   Derive the regression model  $\mathbf{w}_t$  in Eqn. (11).
  - 12:   **if**  $(t = 1)$  **then**  $\mathbf{w} = \mathbf{w}_t$ ;
  - 13:   **else**  $\mathbf{w} = (1 - \alpha) * \mathbf{w} + \alpha * \mathbf{w}_t$ .
  - 14: **until** All frames are traversed.
- 

#### E. The Algorithm

We summarize the whole tracking algorithm in Alg. 1. There are two crucial steps, including the computation of the filtering responses and the regressor. As the spectral filter bases can be pre-computed before detecting targets, the computation cost mainly spends on the matrix inverse operation in Eqn. (11). The computation complexity is about  $O(d^3 K^3)$ , where  $d$  is the feature dimension and  $K$  is the filter order. To speed up SFT, we can project features into low-dimension space by using Principal Components Analysis (PCA) or random projecting (Section IV-B). Besides, the order  $K$  may be downscaled by designing the skipping neighborhood (Section IV-A). For other strategies of accelerating matrix inverse calculation, we leave them as the future work.

### IV. IMPLEMENTAL DETAILS

In this section, we introduce more details of SFT, including how to construct graph, how to reduce feature dimensions, and how to process the scale problem, etc.

#### A. Graph Construction

As shown in Fig. 2, we define the neighborhood based on spatial layout. As analyzed in Section III-C, the filter basis  $T_k(\tilde{\mathcal{L}})$  is an exactly  $k$ -localized filter, where the neighborhood is propagated in  $\mathcal{L}^k$ . Thus we only need to define the spatial nearest neighbors of each reference point as shown in the first two cases of Fig. 2, and employ the  $k$  order filters to evolve neighborhood. Considering the high-degree similarity of image textures in adjacent pixels, we may skip several pixels to connect edges as shown in the last two cases of Fig. 2. Thus, when filtering on the same size region, the skipping mode need less filters (*i.e.*, a smaller  $K$ ). Consequently, the tracker speeds up if the smaller  $K$  is employed, because the complexity of matrix inverse is related to  $K$ .

After choosing neighbourhood, we may assign Gaussian weights or  $\{0, 1\}$  weights to those connected edges. To simplify the weighting step, here we use the  $\{0, 1\}$  weighting

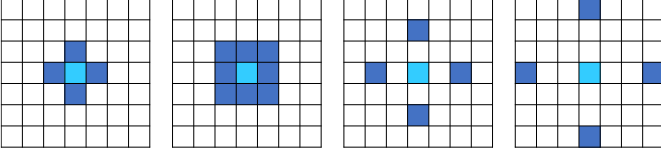


Fig. 2: Some exemplars of adjacency relationship. As image textures of adjacent pixels are high-degree similar, we can skip several pixels to connect edges as shown in the right two cases. The advantage is that the filter order  $K$  can be decreased in proportion for the same size filtering region.

strategy, *i.e.*, the adjacency matrix  $\mathbf{W}$  of the weighted graph  $\mathcal{G}$  is defined as

$$W_{ij} = \begin{cases} 1, & \text{if } e \in \mathcal{E} \text{ connects vertices } i \text{ and } j, \\ 0, & \text{otherwise.} \end{cases} \quad (12)$$

### B. Multi-channel Features

The deep VGG-Net [32] is used to extract high-level features. We crop an image patch with 2.4 times the size of target bounding box and then resize it to  $224 \times 224$  pixels for the VGG-Net with 19 layers. Similar to the literature [28], we use the outputs from six convolutional layers ( $\{10, 11, 12, 14, 15, 16\}$ -th layers) as six types of feature maps. All feature maps are resize to  $57 \times 57$  pixel size, which is about quadruple to the smallest size ( $14 \times 14$ ) of feature maps. The odd size can uniquely define the target center. As the number of feature maps in each layer is 512, SFT will spend high computation cost on the matrix inverse as analyzed in Section III-E, if all feature maps are concatenated together ( $512 \times 6 = 3072$  dimensions). To speed up the tracking, two strategies are taken for the convolutional features: (i) We learn six trackers respectively corresponding to six layer features, and average the six tracking target centers as the final target center. (ii) We employ PCA to project each layer features into 100 dimensions.

### C. Other Details

For the scale estimation, we employ the same strategy to [9]. The filters at multiple resolutions are used to estimated scale changes in the target size. We extract the samples with sizes in scaled factors  $a^r$  ( $r \in \{\lfloor \frac{1-S}{2} \rfloor, \dots, \lfloor \frac{S-1}{2} \rfloor\}$ ) at the previous target location. The scales  $a^r$  is relative to the current target scale, and  $S$  is the number of scales and  $a$  is the scale increment factor. In our experiment, we follow the same parameter settings to [9], where  $S = 33, a = 1.02$ .

For the filter order  $K$ , we make the largest filter basis (*i.e.*,  $T_{K-1}(\tilde{\mathcal{L}})$ ) cover the whole target region. Suppose the target size is  $h \times w$ , the filter order  $K$  is set to  $\max(h, w)$  if choosing the nearest spatial neighborhood (*e.g.*, the first two cases in Fig. 2). For the skipping modes in the last two cases of Fig. 2,  $K$  is assigned to  $\lceil \frac{\max(h, w)}{s} \rceil$ , where  $s$  is the skipping step.

The default neighbor relationship uses the third case in Fig. 2. The balance parameter  $\gamma$  is set to 1.

## V. EXPERIMENT

### A. Dataset and Setting

In order to verify our proposed tracker, we conduct extensive experiments on OTB-2015 dataset [36]. The dataset consists of 100 video sequences with 11 different attributes including illumination changes, scale variation, motion blur, fast motion, etc. Two widely used evaluation criteria, *i.e.*, precision plot and success plot, are used in the following experiments.

Precision plot measures center location error (CLE) which computes the difference between prediction positions and ground truth. It shows how many percentage of frames whose center location error is within a previously given threshold. Here the threshold score is set to 20 pixels. Success plot denotes bounding box overlap ratio which is based on area under the curve (AUC). The overlap ratio is defined as  $S = |r_t \cap r_0| / |r_t \cup r_0|$ , where  $\cup$  and  $\cap$  are the union and intersection operators,  $r_0$  is the predicted bounding box and  $r_t$  is the ground truth. More details can be found in [36].

### B. Selection of Adjacent Vertices

As discussed in Section IV-A, we only need to connect those nearest neighbors, as spectral graph filters can propagate the neighbor relationship to distant vertexes. Here we test four cases of Fig. 2, whose results are reported in Tab. I. From this table, we can observe that, (i) more neighbors (Case 2) slightly degrade the performance, which may be attribute to feature confusion after averaging features of all neighbors during compute  $\mathcal{L}^k \mathbf{X}$ ; (ii) the skipping mode with one pixel interval (Case 3) reaches the best performance. The skipping strategy can be regarded as the downsampling for feature maps. Thus the increase of skipping step (Case 4) degrades the performance because some useful information can not be encoded in the filtering process. Thus, we use the third case as the default setting in the following experiments.

TABLE I: The performance of different neighborhood strategies as shown in Fig. 2. Case 1~4 are respectively correspond to the sequential subfigures. **Note that here we don't perform scale estimation.**

Case 1		Case 2		Case 3		Case 4	
CLE	AUC	CLE	AUC	CLE	AUC	CLE	AUC
0.855	0.572	0.847	0.565	0.866	0.576	0.862	0.574

### C. Comparisons with CF Based Trackers

To fairly compare the CF based model, the same features of VGG-Net are feed into the CF based model, which is our standard baseline, called VGG\_CF. Besides, we compare the classic CF based methods, CSK [16] and KCF [17]. Fig. 3 shows the results under the precision plots of One-Pass Evaluation(OPE) and the success plots based on area under curve(AUC). The performances of the three CF based methods are quite different. As the CSK trackers only use raw feature and CSK use HOG feature while the robust deep CNN feature is employed for VGG\_CF tracker, which makes it outperform the other two CF based methods obviously. Compared to the baseline VGG\_CF, our proposed SFT achieves a gain



of 3.4% in CLE. Meanwhile we obtains an AUC scores of 57.6% which also outperforms VGG\_CF tracker. The reason may be two folds: (i) local filtering on spatial regions, (ii) rotation-invariance and shifting-invariance for graph structure. To implement an intrinsic comparison, here we don't process the scale.

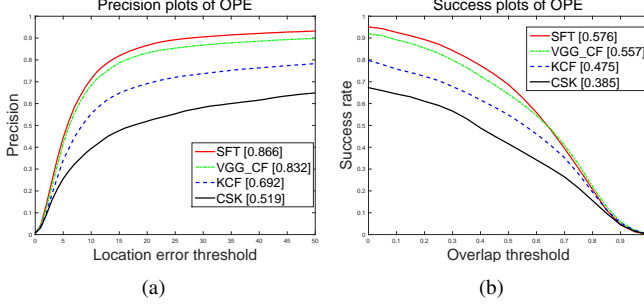


Fig. 3: The precision and success plot of comparisons with CF based methods. **Note that here we don't perform scale estimation for an intrinsic comparison.**

#### D. Comparisons with State-of-the-art

We compare our proposed SFT with nine state-of-the-art trackers: DeepSRDCF [10], MEEM [37], HDT [28], CSK [16], KCF [17], DSST [9], SCM [39], STRUCK [15], TLD [20]. The DeepSRDCF and HDT are two recent representative deep learning based trackers. Others employ HOG feature mostly. Only the top-10 trackers are reported in the experiments.

**Quantitative Evaluation.** Fig. 4 plots the precision curves and success curves among all trackers. The top-10 trackers ranked by CLE and AUC scores are shown with different colors. From these figures, we have three observations: (i) In the precision plot, SFT outperforms all state-of-the-art trackers, which demonstrates its effectiveness. (ii) In success plot, SFT achieves a comparable result with the current best method DeepSRDCF. Actually here we only use the parameters used in [9] without any tuning for scale estimation. (iii) In fine localization, SFT is slightly inferior to DeepSRDCF, but SFT is more robust to those large appearance variations. The main reason is that, the spectral filtering averages all features of neighbors when all edges are assigned to equal weights, thus in the filtering some subtle textures may be lost while most invariant information is preserved.

**Attribute-based evaluation.** For comprehensive analysis of our proposed SFT, we provide each attribute plot in Fig. 5. As observed from these figures, SFT achieves more excellent performance compared to state-of-the-art in almost all cases. Particularly, SFT is rather effective in handling low resolution, background cluster and illumination variation challenges. In the case of low resolution the CLE score of our tracker is 99.8% which surpasses DeepSRDCF by 11.1%. However, our tracker seems lost target easily in the case of fast motion and motion blur, which might be attribute to the small search window of SFT or boundary effect.

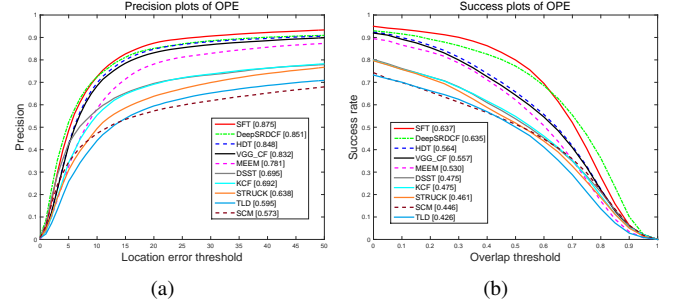


Fig. 4: The precision and success plots of quantitative comparison for the 100 sequences on OTB-2015 dataset. The center location error (CLE) and area under curve (AUC) scores of the top 10 trackers are reported. Best viewed with Zooming up.

**Qualitative evaluation.** Fig. 6 shows some visual results of the top ranked trackers (including our proposed SFT, DeepSRDCF, KCF, MEEM, DSST, DLT [33]) on the eight most challenging image sequences, Ironman, Matrix, MotorRolling, Skiing, Tiger1, Box, Human3, Human6. As shown in Fig 6, the prediction position and bounding box of our proposed method are more precise than others trackers in various scenes.

## VI. CONCLUSION

In this paper, we propose a simple but efficient Spectral Filter Tracking (SFT) method. In SFT, the candidate image region is model as a pixel-level grid graph. To estimate the best-matching vertex, we borrow spectral graph filters to encode the local graph structure. Considering the computation cost of eigenvalue decomposition on the Laplacian matrix, we approximate spectral filtering as the polynomial of a series of filter bases. For the filter bases, we employ the Chebyshev expansion terms, where each term encodes a localized filtering region of graph. Thus all filter bases span a multi-scale filtering space. Finally the filtering parameters and feature projecting function are jointly reduced into a simple regression model. The proposed SFT simply boils down to only a few line codes, but the experimental results on the dataset [36] demonstrate that the SFT is more effective and achieves state-of-the-art performance. In future, we will consider to speed up the tracker.

## REFERENCES

- [1] A. Adam, E. Rivlin, and I. Shimshoni. Robust fragments-based tracking using the integral histogram. In *IEEE Computer Society Conference on Computer vision and pattern recognition*, volume 1, pages 798–805, 2006.
- [2] S. Avidan. Support vector tracking. *IEEE transactions on pattern analysis and machine intelligence*, 26(8):1064–1072, 2004.
- [3] B. Babenko, M.-H. Yang, and S. Belongie. Visual tracking with online multiple instance learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 983–990, 2009.
- [4] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui. Visual object tracking using adaptive correlation filters. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2544–2550, 2010.
- [5] Z. Cai, L. Wen, J. Yang, Z. Lei, and S. Z. Li. Structured visual tracking with dynamic graph. In *Asian Conference on Computer Vision*. 2013.

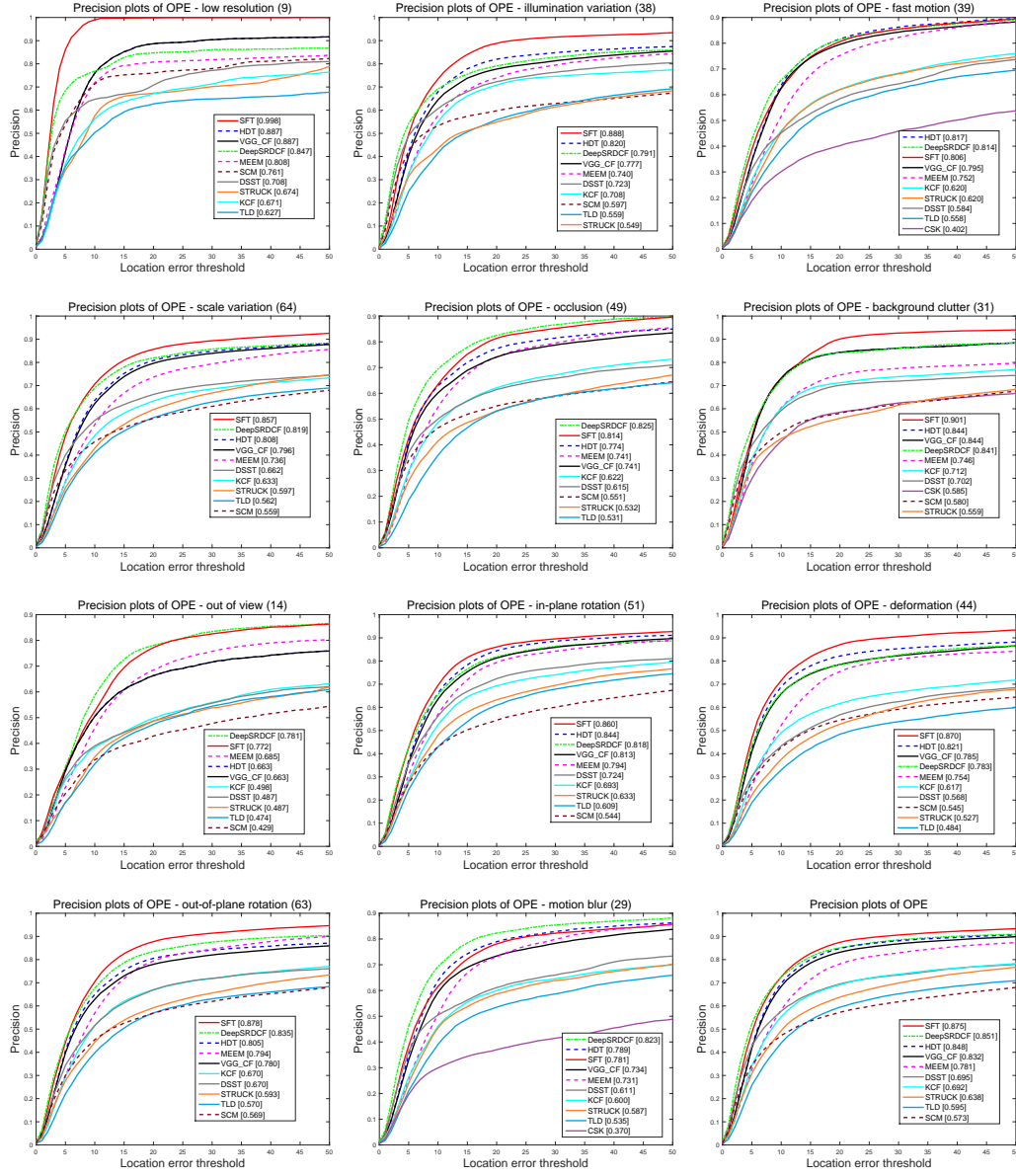


Fig. 5: The precisions of plots of 11 attributes. Our tracker achieves is superior to other methods in most cases except motion blur and fast motion. The reasons of failures might be the small search window ( $2.4\times$ ) and boundary effects for our method. Best viewed with Zooming up.

- [6] L. Cehovin, M. Kristan, and A. Leonardis. Robust visual tracking using an adaptive coupled-layer visual model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(4):941–953, 2013.
- [7] Z. Chen, Z. Hong, and D. Tao. An experimental survey on correlation filter-based tracking. *arXiv preprint arXiv:1509.05520*, 2015.
- [8] Z. Cui, S. Xiao, J. Feng, and S. Yan. Recurrently target-attending tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1449–1458, 2016.
- [9] M. Danelljan, G. Häger, F. Khan, and M. Felsberg. Accurate scale estimation for robust visual tracking. In *British Machine Vision Conference, Nottingham, September 1-5, 2014*. BMVA Press, 2014.
- [10] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg. Convolutional features for correlation filter based visual tracking. In *IEEE International Conference on Computer Vision Workshops*, pages 58–66, 2015.
- [11] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg. Learning spatially regularized correlation filters for visual tracking. In *IEEE International Conference on Computer Vision*, pages 4310–4318, 2015.
- [12] M. Danelljan, F. Shahbaz Khan, M. Felsberg, and J. Van de Weijer. Adaptive color attributes for real-time visual tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1090–1097, 2014.
- [13] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2010.
- [14] D. K. Hammond, P. Vandergheynst, and R. Gribonval. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30(2):129–150, 2011.
- [15] S. Hare, S. Golodetz, A. Saffari, V. Vineet, M.-M. Cheng, S. L. Hicks, and P. H. Torr. Struck: Structured output tracking with kernels. *IEEE transactions on pattern analysis and machine intelligence*, 38(10):2096–2109, 2016.
- [16] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. Exploiting the circulant structure of tracking-by-detection with kernels. In *European conference on computer vision*, pages 702–715. Springer, 2012.
- [17] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern*



Fig. 6: Qualitative results of SFT on some most challenging sequences (Ironman, Matrix, MotorRolling, Skiing, Tiger1, Box, Human3, Human6 respectively from left to right and top to bottom). Best viewed with Zooming up.

- Analysis and Machine Intelligence*, 37(3):583–596, 2015.
- [18] Z. Hong, C. Wang, X. Mei, D. Prokhorov, and D. Tao. Tracking using multilevel quantizations. In *European Conference on Computer Vision*, pages 155–171. Springer, 2014.
- [19] X. Jia, H. Lu, and M.-H. Yang. Visual tracking via adaptive structural local sparse appearance model. In *IEEE International Computer Vision and Pattern Recognition*, 2012.
- [20] Z. Kalal, J. Matas, and K. Mikolajczyk. Pn learning: Bootstrapping binary classifiers by structural constraints. In *IEEE International Computer Vision and Pattern Recognition*, 2010.
- [21] J. Kwon and K. M. Lee. Tracking of a non-rigid object via patch-based dynamic appearance modeling and adaptive basin hopping monte carlo sampling. In *IEEE International Computer Vision and Pattern Recognition*, 2009.
- [22] J. Kwon and K. M. Lee. Highly nonrigid object tracking via patch-based dynamic appearance modeling. *IEEE transactions on pattern analysis and machine intelligence*, 35(10):2427–2441, 2013.
- [23] Y. Li and J. Zhu. A scale adaptive kernel correlation filter tracker with feature integration. In *European Conference on Computer Vision/ECCV Workshops*, 2014.
- [24] Y. Li, J. Zhu, and S. C. Hoi. Reliable patch trackers: Robust visual tracking by exploiting reliable patches. In *IEEE International Computer Vision and Pattern Recognition*, 2015.
- [25] B. Liu, J. Huang, L. Yang, and C. Kulikowsk. Robust tracking using local sparse appearance model and k-selection. In *Computer Vision and Pattern Recognition*, 2011.
- [26] T. Liu, G. Wang, and Q. Yang. Real-time part-based visual tracking via adaptive correlation filters. *Intelligence*, page 2345390, 2015.
- [27] C. Ma, X. Yang, C. Zhang, and M.-H. Yang. Long-term correlation tracking. In *IEEE International Computer Vision and Pattern Recognition*, 2015.
- [28] Y. Qi, S. Zhang, L. Qin, H. Yao, Q. Huang, J. Lim, and M.-H. Yang. Hedged deep tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4303–4311, 2016.
- [29] X. Ren and J. Malik. Tracking as repeated figure/ground segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.
- [30] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang. Incremental learning for robust visual tracking. *International Journal of Computer Vision*, 77(1-3):125–141, 2008.
- [31] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine*, 30(3):83–98, 2013.
- [32] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [33] N. Wang and D.-Y. Yeung. Learning a deep compact image representation for visual tracking. In *Advances in neural information processing systems*, pages 809–817, 2013.
- [34] S. Wang, H. Lu, F. Yang, and M.-H. Yang. Supapixel tracking. In *International Conference on Computer Vision*, pages 1323–1330, 2011.
- [35] L. Wen, D. Du, Z. Lei, S. Z. Li, and M.-H. Yang. Jots: Joint online tracking and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2226–2234, 2015.
- [36] Y. Wu, J. Lim, and M.-H. Yang. Object tracking benchmark. 2015.
- [37] J. Zhang, S. Ma, and S. Sclaroff. Meem: robust tracking via multiple experts using entropy minimization. In *European Conference on Computer Vision*, pages 188–203. Springer, 2014.
- [38] T. Zhang, K. Jia, C. Xu, Y. Ma, and N. Ahuja. Partial occlusion handling for visual tracking via robust part matching. In *IEEE International Computer Vision and Pattern Recognition*, 2014.
- [39] W. Zhong, H. Lu, and M.-H. Yang. Robust object tracking via sparsity-based collaborative model. In *Computer Vision and Pattern Recognition*, 2012.