# Iterative Graph Seeking for Object Tracking

Dawei Du, Longyin Wen, *Member, IEEE*, Honggang Qi, *Member, IEEE*, Qingming Huang, *Fellow, IEEE*, Qi Tian, *Fellow, IEEE*, and Siwei Lyu, *Senior Member, IEEE*

***Abstract*— To effectively solve the challenges in object tracking, such as large deformation and severe occlusion, many existing methods use graph-based models to capture target part relations, and adopt a sequential scheme of target part selection, part matching, and state estimation. However, such methods have two major drawbacks: 1) inaccurate part selection leads to performance deterioration of part matching and state estimation and 2) there are insufficient effective global constraints for local part selection and matching. In this paper, we propose a new object tracking method based on iterative graph seeking, which integrate target part selection, part matching, and state estimation using a unified energy minimization framework. Our method also incorporates structural information in local parts variations using the global constraint. We devise an alternative iteration scheme to minimize the energy function for searching the most plausible target geometric graph. Experimental results on several challenging benchmarks (i.e., VOT2015, OTB2013, and OTB2015) demonstrate improved performance and robustness in comparison with existing algorithms.**

***Index Terms*— Object tracking, iterative graph seeking, alternative iteration scheme, energy minimization.**

Manuscript received April 4, 2017; revised September 16, 2017 and November 23, 2017; accepted December 9, 2017. Date of publication December 20, 2017; date of current version January 12, 2018. This work was supported in part by the National Natural Science Foundation of China under Grant 61620106009, Grant 61332016, Grant U1636214, Grant 61650202, Grant 61472388, Grant 61771341, and Grant 61429201, in part by the Key Research Program of Frontier Sciences, CAS: QYZDJ-SSW-SYS013, in part by ARO under Grant W911NF-15-1-0290, in part by Faculty Research Gift Awards through the NEC Laboratories of America and Blippar, and in part by the U.S. National Science Foundation National Robotics Initiative under Grant IIS-1537023. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Jianfei Cai. *(Corresponding authors: Honggang Qi; Qingming Huang.)*

D. Du and H. Qi are with the School of Computer and Control Engineering, University of Chinese Academy of Sciences, Beijing 100190, China, and also with the Key Laboratory of Big Data Mining and Knowledge Management, University of Chinese Academy of Sciences, Beijing 101408, China (e-mail: dawei.du@vipl.ict.ac.cn; hgqi@jdl.ac.cn).

L. Wen is with GE Global Research, Niskayuna, NY 12309 USA (e-mail:longyin.wen@ge.com).

Q. Huang is with the School of Computer and Control Engineering, University of Chinese Academy of Sciences, Beijing 100190, China, also with the Key Laboratory of Big Data Mining and Knowledge Management, University of Chinese Academy of Sciences, Beijing 101408, China, and also with the Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China (e-mail: qmhuang@ucas.ac.cn).

Q. Tian is with the Department of Computer Science, The University of Texas at San Antonio, San Antonio, TX78249-1604 USA (e-mail: qi.tian@utsa.edu).

S. Lyu is with the Computer Science Department, University at Albany, State University of New York, Albany, NY12222 USA (e-mail: slyu@albany.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2017.2785626

## I. Introduction

OBJECT tracking is one of the fundamental problems in computer vision with a range of applications, *e.g.*, wide area surveillance, autonomous driving, unmanned aerial vehicle and human-computer interaction. Although much work has been done for this task, it is still challenging due to the difficulty in handling geometric deformations, in-plain and out-of-plain rotations, partial occlusions and cluttered backgrounds.

Many existing methods, *e.g.*, [7], [14], [16], [37], [45], focus on the global appearance variations of the object. They may be less effective in modeling local appearance variations when handling large deformations and occlusions. To this end, some recent methods have attempted to use local models to describe object appearance, using pixels [12], [31], keypoints [29], [47], rectangular parts [5], [42] and superpixels [11], [35].

Pixel based methods focus on modeling the object at the pixel-level, but tend to have weak discriminability in cluttered background. The performance of keypoint based methods depends heavily on detection accuracy of keypoints, which makes them less effective for videos with extensive textureless regions with fewer keypoints. Rectangular parts usually include background pixels into the appearance model for objects with non-rectangular shape. Superpixel based methods use a group of mid-level visual cues to represent target parts. Therefore, local appearance variations are exploited by superpixel representation more effectively. As illustrated in Fig. 2(a), superpixel based methods generally consist of three sequential steps: target part selection, target part matching, and target state estimation. Target part selection first selects candidate target parts from the background. Then, a local appearance model is used to associate parts between consecutive frames by clustering methods, called target part matching. Finally, target state (*i.e.*, center pixel location and size of the target) is estimated based on majority voting of matching results.

However, their performances are still not satisfactory in complex scenarios for two reasons. First, part selection is separated from part matching and state estimation. Due to imprecise appearance model, inaccurate part selection will negatively affect part matching, even resulting in tracking failure. Second, they do not take global constraint into consideration, making them sensitive to local noise in severe occlusion or cluttered background. The relations of these three components are summarized as follows:

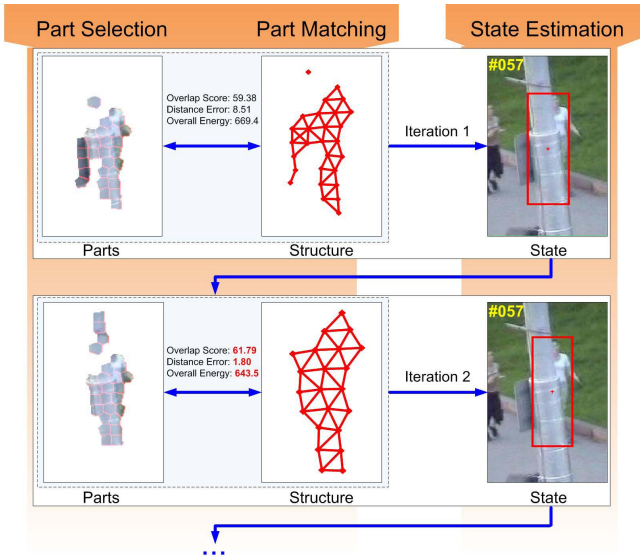



Fig. 1. The proposed method. Along with the decrease of overall energy in (2) in the iterative optimization, both two measures (*i.e.*, overlap score and distance error) are improved. The shadow and bidirectional arrows mean part selection and part matching are considered simultaneously.

1) Target parts are matched between two consecutive frames to provide complementary cues for part selection in the current frame, and vice versa.
2) Both part selection and matching capture local appearance variations to facilitate global target state estimation.
3) The estimated target state in turn provides a global constraint to improve the accuracy of part selection and matching.

To address these issues, in this paper, we formulate object tracking as an iterative graph seeking problem. To be specific, as shown in Fig. 2(b), we seek the target geometric graph by assembling target part selection, target part matching, and target state estimation into a unified energy minimization framework. The target is represented by a geometric graph, in which the nodes correspond to the target parts (superpixels) and the edges represent the interactions between them. In this process, we search the most plausible geometric graph iteratively in the current frame with three constraints of different aspects, namely local part selection and matching, and global target state estimation, which is called *Iterative Graph Seeking* (IGS). Experimental results on challenging benchmark datasets demonstrate that the IGS tracker effectively overcomes the aforementioned limitations of existing superpixel based methods and considerably improves tracking performance.

The contributions of this paper are summarized as follows:

- Object tracking task is formulated as an iterative graph seeking problem by assembling target part selection, part matching, and state estimation into a unified energy objective to make them provide complementary information for robust tracking performance.
- An iterative algorithm is proposed to solve the energy minimization problem with multiple variables effectively.
- Experiments on several challenging benchmark datasets (*i.e.*, VOT2015, OTB2013 and OTB2015) show improved performance of our tracker against the existing methods.

The remainder of this paper is organized as follows: related works on object tracking are reviewed in Section II. The proposed approach is presented in Section III. The optimization of our method is described in Section IV. Experimental evaluations on three benchmark datasets are reported in Section V. Concluding remarks are given in Section VI.

## II. Related Works

In this section, we briefly discuss works that are closely related to our work. From the point of the view about how trackers representing the object, they can be classified into two categories: global-based and local-based.

### A. Global-Based Methods

Several existing object tracking methods focus on designing robust appearance model to describe the appearance variations of the object in bounding box [14], [20], [37], [45], [48]. In [37], object appearance is represented by both temporal and spatial context information. Gao *et al.* [14] formulate the probability of object appearance as exponentially related to the confidence of a classifier output using Gaussian process regression. To address the model drift problem, a multi-expert restoration scheme is proposed in [45], where the best expert is selected from a tracker and its historical snapshots to correct undesirable model updates based on a minimum entropy criterion. Hua *et al.* [20] develop a tracking-by-detection framework based on a set of candidate object locations, and select the best candidate as the tracking result based on detection confidence score and an objectness measure computed with edges. Similarly, Zhu *et al.* [48] introduce informative contours to score proposals based on the number of contours for object tracking.

Recently, correlation filter based methods [2], [6], [7], [22], [26], [33] are widely leveraged for this task because of good performance and high speed to model object appearance. Danelljan *et al.* [7] propose to learn discriminative correlation filters based on a scale pyramid representation to track the object. In order to resolve the problem of the fixed template size in [7], Li and Zhu [22] design an effective scale adaptive scheme, and then integrate HOG and color name features to further boost the overall tracking performance. Tang and Feng [33] derive a multi-kernel correlation filter based tracker to fully exploit the invariance-discriminative power spectrums of various features. For long-term tracking, a kernel ridge regression method is developed in [26] to encode the appearance of target and its context based on correlation filters. Besides, local color statistics are combined in correlation filters using a ridge regression framework to cope with variation in shape [2]. To modulate the distribution of attention according to various feature and kernel types of the target, Choi *et al.* [6] propose a novel attentional feature-based correlation filter evolved with multiple trained elementary trackers. These methods focus on appearance variations of global bounding box but with little consideration about local target structure information. Nevertheless, the target structure is an important cue to deal with deformation and occlusion challenges.

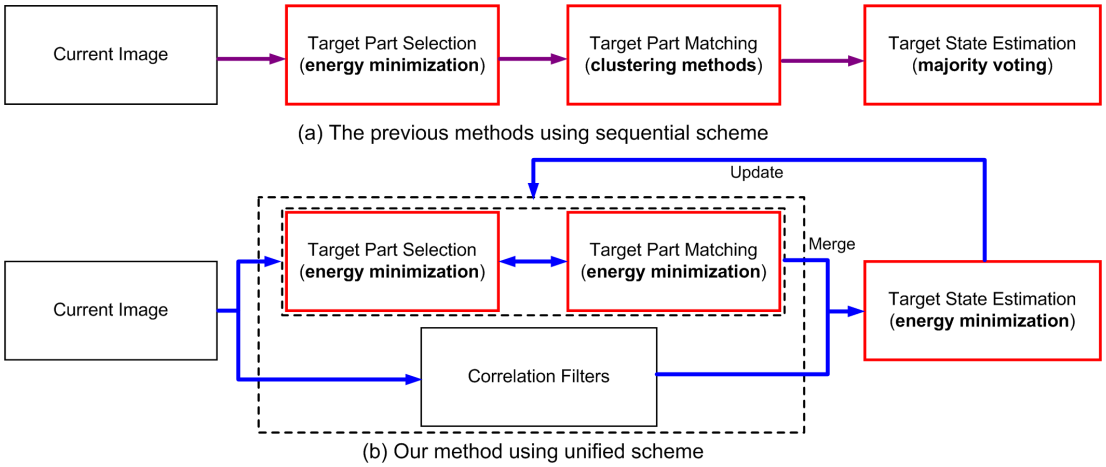



Fig. 2. The flowchart of superpixel based methods including target part selection, part matching and state estimation (red rectangle). (a) The previous methods [4], [10], [35] correspond to a sequential order of three steps (violet line). (b) Our method combines them into a unified framework (blue line). The blue bidirectional arrows indicate the interactions between the target part selection module and the target part matching module, and the one-way arrows represent the sequential relations of these modules.

### *B. Local-Based Methods*

Different from global-based methods, local-based methods are developed to capture local appearance and structure changes of the object. In [29], a keypoint based tracking method is developed to use the clustering of correspondences to distinguish between inlier and outlier keypoints. Zhao *et al.* [47] develop a keypoint tracker based on spatio-temporal multi-task structured output optimization driven by discriminative metric learning. Yu *et al.* [43] select mid-level keypoints by max-pooling over the local descriptor responses from a set of filters to locate the object. However, sparse sampling of keypoints may not locate the tracking object accurately, and may even loss the object in textureless scenes.

For dense sampling, some researchers employ a few rectangular parts to encode target structure [5], [24], [27], [42]. Yao *et al.* [42] train an online latent structured SVM to predict the location of parts, where both a global bounding box and a small number of part boxes are generated to approximate target state. Cehovin *et al.* [5] develop a coupled-layer model that combines the object's global and local appearance to handle significant appearance variations. Maresca and Petrosino [27] combine a set of local tracker responses on different size of rectangular parts to track the object. In [24], a few group of correlation filters are applied to address non-rigid deformations of target parts. However, rectangular parts may include redundant background information to learn appearance model, leading to inferior tracking performance.

To account for non-rectangular shape of the object, recent works usually take advantages of segmentation techniques [12], [19], [31] to find the object contour in pixel-level. In [12], Duffner and Garcia propose a generalized Hough transform based detector with pixel-based descriptors and a probabilistic foreground segmentation method for tracking in a combined way. In [19], a hierarchical appearance representation model is proposed for object tracking. Specifically, shared information is extracted across multi-level quantization of an image, *i.e.*, pixels, super-pixels and bounding boxes, by a probabilistic graphical model. In [31], an appearance-based tracking approach is developed by using color histograms to distinguish the object from its surrounding region.

On the other hand, superpixel representation is also used to model the object appearance [30], [41], [44]. Yang *et al.* [41] present a discriminative appearance model to distinguish the target and the background by confidence map at superpixel level. Yuan *et al.* [44] combine depth cue and mid-level features captured by superpixels to provide discriminability for the target and background separation. In [30], a probabilistic model of the target variations over time is proposed based on the Earth Mover's Distance minimization problem.

To consider the relation between superpixels, another line of superpixel based methods is to match candidate foreground superpixels of the target in two consecutive frames to estimate target state [4], [10], [35]. Wang and Yagi [35] propose a tracking method based on many-to-many image superpixel matching, and then estimate target state by searching for the maximum probability on the confidence map of superpixel displacement. In [4], the tracking problem is formulated as graph matching between the geometric structure graph of the target and that of the candidate target proposals graph. Du *et al.* [10] develop a dense subgraph searching based method, where the target is represented by a hypergraph constructed from segmented foreground superpixels in several consecutive frames. With all the advantages the superpixel representation offers, its disadvantages are also obvious: most of them only use local information, which is insufficiently discriminative when object appearance changes drastically.

## III. Methodology

### *A. Problem Formulation*

The object tracking problem is formulated as the estimation of target state $\mathcal{B}_t = \{o_t, s_t\}$. Here $o_t$ and $s_t$ are the center pixel location and size of the target at frame $t$ in searching region $\mathcal{R}_t$ centered at $o_{t-1}$ (*i.e.*, the center pixel location in the previous frame).

We first apply the SLIC algorithm [1] to over-segment $\mathcal{R}_t$ into a group of superpixels $\mathcal{P}_t = \{p_{i,t}\}_{i=1}^{m}$. For target part selection, we aim to distinguish target parts from background using the appearance model $\mathcal{M}_t$. The appearance model is based on an online SVM classifier $\mathcal{C}_t$ and a HSV histogram $\mathcal{H}_t$, *i.e.*, $\mathcal{M}_t = \{\mathcal{C}_t, \mathcal{H}_t\}$. To encode the structure information of the target, we construct a geometric graph $\mathcal{G}_t = \{\mathcal{X}_t, \mathcal{E}_t\}$, where $\mathcal{X}_t$ is the node set corresponding to target parts, and $\mathcal{E}_t$ is the edge set representing the interactions between them. For each superpixel $p \in \mathcal{P}_t$, we assign a label $\ell_p \in \mathbf{b} \cup \mathbf{f}$, where $\mathbf{b} = \{f_0\}$ and $\mathbf{f} = \{f_1, \cdots, f_n\} \cup \{f_1^+, \cdots, f_{n'}^+\}$ are the label of background and foreground, respectively. Specifically, $\{f_1, \cdots, f_n\}$ is the label set of target parts in $\mathcal{G}_t$, and $\{f_1^+, \cdots, f_{n'}^+\}$ is the label set of newly generated target parts. In other words, three types of labels are assigned to each superpixel, namely background, labels of target parts in $\mathcal{G}$ and newly added target parts. We denote the labeling of all superpixels as $\mathcal{L}_t = \{\ell_p\}_{p \in \mathcal{P}_t}$. Last, we use the correlation filters $\mathcal{F}_t$ to incorporate a global constraint to estimate target state $\mathcal{B}_t$ in the current frame.

The overall tracking is then formulated as optimizing the following energy function given the current appearance model $\mathcal{M}_t$, correlation filters $\mathcal{F}_t$, geometric graph $\mathcal{G}_t$, labeling $\mathcal{L}_t$, and target state $\mathcal{B}_t$:

$$(\mathcal{B}_t^*, \mathcal{L}_t^*, \mathcal{M}_t^*, \mathcal{G}_t^*, \mathcal{F}_t^*) = \underset{\mathcal{B},\mathcal{L},\mathcal{M},\mathcal{G},\mathcal{F}}{\operatorname{argmin}} E_t^{\text{all}}, \tag{1}$$

where the overall energy $E_t^{\text{all}}$ consists of three components:

$$E_t^{\text{all}} = \underbrace{E_t^{\text{PS}}(\mathcal{L}, \mathcal{M})}_{\text{part selection}} + \underbrace{E_t^{\text{PM}}(\mathcal{L}, \mathcal{G})}_{\text{part matching}} + \underbrace{E_t^{\text{SE}}(\mathcal{B}, \mathcal{L}, \mathcal{F})}_{\text{state estimation}}. \tag{2}$$

$E_t^{\text{PS}}(\mathcal{L}, \mathcal{M})$, $E_t^{\text{PM}}(\mathcal{L}, \mathcal{G})$ and $E_t^{\text{SE}}(\mathcal{B}, \mathcal{L}, \mathcal{F})$ are the energy functions of part selection, part matching and state estimation, respectively. For clarity of presentation, we drop the frame index $t$ whenever it does not cause confusion in the following sections, and the subscript $t$ in (2) indicates that these energy functions depend on the values of $\mathcal{B}_t, \mathcal{L}_t, \mathcal{M}_t, \mathcal{G}_t, \mathcal{F}_t$. We describe the initialization of models (*i.e.*, $\mathcal{M}_t, \mathcal{G}_t, \mathcal{F}_t$) in Section III-B, followed by the description of the three energy functions in detail, Section III-C∼III-E.

### B. Initialization

Based on the first annotated state $\mathcal{B} = \{o, s\}$, we generate searching region $\mathcal{R}$ centered at $o$ with the size of $\xi$ times of target size $s$. We regard superpixels inside the target bounding box as positive training samples, and superpixels inside the searching region but outside the bounding box as negative ones. Thus, an SVM classifier $\mathcal{C}$ and a HSV histogram $\mathcal{H}$ are initialized by these samples. Meanwhile, correlation filters $\mathcal{F}$ are initialized by calculating features maps on $\mathcal{R}$ as in [7].

Besides, the geometric graph $\mathcal{G} = \{\mathcal{X}, \mathcal{E}\}$ is constructed from the current target parts. Specifically, each node $x \in \mathcal{X}$ is a target part such that $\{x|\ell_x \neq f_0\}$, and each edge $e \in \mathcal{E}$ is determined by geometric neighboring parts, *i.e.*, $\{e|\forall x_i, x_j \in e, ||z(x_i) - z(x_j)||_2 \leq 2d\}$. Here $z(\cdot)$ denotes the geometric center of all pixels of that part, and $d = \sqrt{W \cdot H/\rho}$ is the average diameter of $\rho$ superpixels in the searching region with width $W$ and height $H$. $2d$ is the distance threshold for neighboring parts. Target parts are then selected with (3) by the graph-cut algorithm [3]. To represent local appearances, each target part is represented by its appearance vector obtained from concatenating LBP and HSV features and relative geometric position with respect to target center.

### C. Energy of Part Selection

The energy function of part selection consists of data cost and smoothness cost, defined as

$$E^{\text{PS}}(\mathcal{L}, \mathcal{M}) = \sum_{p \in \mathcal{P}} E^{\text{da}}(\ell_p, \mathcal{M}) + \lambda^{\text{b}} \sum_{p,q \in \mathcal{N}} E^{\text{sm}}(\ell_p, \ell_q), \tag{3}$$

where $\mathcal{N}$ is the set of pairs of interacting superpixels. The data cost $E^{\text{da}}$ measures the probability of superpixel belonging to the target or background, and the smoothness cost $E^{\text{sm}}$ encourages consistency of labels. Scaling factor $\lambda^{\text{b}}$ balances the contribution of the two terms.

The data cost is defined as

$$E^{\text{da}}(\ell_p, \mathcal{M}) = \begin{cases} \frac{1}{2}(-\min(\mathcal{C}(p), \delta) + \min(\mathcal{H}(p, \ell_p), \delta)) & \ell_p \in \mathbf{f}, \\ \frac{1}{2}(\min(\mathcal{C}(p), \delta) + \min(\mathcal{H}(p, \ell_p), \delta)) & \ell_p \in \mathbf{b}, \end{cases} \tag{4}$$

where $\ell_p \in \mathbf{f}$ denotes $\ell_p \in \{f_1, \cdots, f_n\} \cup \{f_1^+, \cdots, f_{n'}^+\}$, and $\ell_p \in \mathbf{b}$ is equivalent to $\ell_p \in \{f_0\}$. $\mathcal{C}(p)$ is the SVM[1] classification score of superpixel $p$ belonging to the target, and $\mathcal{H}(p, \ell_p)$ is the average likelihood of pixels in superpixel $p$ from HSV histogram. Here we use the above two terms to support each other for more robustness. Specifically, the average likelihood of pixels focuses on describing the appearance of the pixel of the superpixels of the target, while the SVM classifier aims to distinguish the superpixels of the target from the superpixels of the background. Besides, to prevent infinite value in the data cost, we set $|\mathcal{C}(p)| \leq \delta$, $|\mathcal{H}(p, \ell_p)| \leq \delta$, with $\delta = 10\,000$ in our implementation.

The smoothness cost is defined as

$$E^{\text{sm}}(\ell_p, \ell_q) = \mathbb{I}(\ell_p \neq \ell_q) \cdot \exp(-||c(p) - c(q)||_2), \tag{5}$$

where $\mathbb{I}(\cdot) = 1$ if its argument is true, and 0 otherwise. $c(p)$ corresponds to mean RGB values of pixels in superpixel $p$.

### D. Energy of Part Matching

The energy function of part matching is the summation of appearance cost and geometric cost, calculated as

$$E^{\text{PM}}(\mathcal{L}, \mathcal{G}) = \sum_{p \in \mathcal{P}} E^{\text{ap}}(\ell_p, \mathcal{G}) + \lambda^{\text{b}} \sum_{p,q \in \mathcal{N}} E^{\text{ge}}(\ell_p, \ell_q, \mathcal{G}). \tag{6}$$

We assume that for high frame-rate sequences target parts move smoothly and target structure does not change drastically in two consecutive frames. In other words, their geometric centers and appearances change slowly in a very short time interval. We measure this using the distance threshold $\varepsilon$.

[1]We use the linear SVM model in the Liblinear toolbox, which can be found in the website: http://www.csie.ntu.edu.tw/~cjlin/liblinear/.

The appearance cost is defined as

$$E^{\mathrm{ap}}(\ell_p, \mathcal{G}) = \begin{cases} \chi^2(\zeta(p), \zeta(x)) & \text{if } ||z(p) - z(x)||_2 \le \varepsilon, \\ \varrho \cdot \chi^2(\zeta(p), \zeta(x)) & \text{otherwise,} \end{cases} \quad (7)$$

where $\chi^2(\cdot,\cdot)$ is the chi-squared distance between appearance vectors $\zeta(p)$ and $\zeta(x)$. $\varrho$ is a scaling factor used to make the superpixel to match with its neighboring parts in $\mathcal{G}$, *i.e.*, $\varrho > 1$ (we set $\varrho = 20$ in this work).

On the other hand, the geometric relations between neighboring parts in $\mathcal{G}$ and foreground parts in $\mathcal{R}$ are compatible because of relative invariance of target structure. We define the geometric cost as

$$E^{\mathrm{ge}}(\ell_p, \ell_q, \mathcal{G}) = \mathbb{I}(\ell_p = f_{x_i}, \ell_q = f_{x_j}) \cdot \exp(-\frac{1}{d}||(z(x_i) - z(x_j)) - (z(p) - z(q))||^2), \quad (8)$$

where $e_1 = \{x_i, x_j\}$ and $e_2 = \{p, q\}$ are edges describing geometric relations respectively. They satisfy the constraint of distance threshold for neighboring parts, *i.e.*, $\|z(p) - z(x_i)\|_2 \le 2d$ and $\|z(q) - z(x_j)\|_2 \le 2d$.

### E. Energy of State Estimation

The energy function of state estimation reflects our request that target state occupies foreground/background pixels as many/few as possible, and has a relative large response score by correlation filters. It is defined as

$$E^{\mathrm{SE}}(\mathcal{B}, \mathcal{L}, \mathcal{F}) = -\Big[(1 - \lambda^{\mathrm{c}}) \cdot \mathcal{F}(\mathcal{B}, \mathcal{R}) + \lambda^{\mathrm{c}} \sum_{p \in \mathcal{B}} \big(\mathbb{P}(\ell_p \in \mathbf{f}) - \mathbb{P}(\ell_p \in \mathbf{b})\big)\Big], \quad (9)$$

where $\mathbf{f} = \{f_1, \cdots, f_n\} \cup \{f_1^+, \cdots, f_{n'}^+\}$ is the label of target parts and newly generated target parts, respectively. $n$ and $n^+$ denote the number of labels from target parts and newly generated target parts. $\mathbf{b} = \{f_0\}$ is the label of background. Thus $\mathbb{P}(\ell_p \in \mathbf{f})$ and $\mathbb{P}(\ell_p \in \mathbf{b})$ count the percentage of pixels of foreground and background superpixel $p$ in $\mathcal{B}$, respectively. The part labels capture the local appearance of target parts, and we use correlation filters to exploit the global appearance of the target. Specifically, $\mathcal{F}(\mathcal{B}, \mathcal{R})$ returns the response score of center $o$ of any rectangular region $\mathcal{B} \subset \mathcal{R}$ as the potential bounding box of the target, scaling factor $\lambda^{\mathrm{c}}$ balances the two terms such that $0 \le \lambda^{\mathrm{c}} \le 1$. Putting simply, (9) indicates that larger response score, larger number of foreground pixels, and smaller number of background pixels correspond to smaller overall energy.

## IV. Optimization

The energy minimization problem in (2) involves several variables, *i.e.*, $\mathcal{B}$, $\mathcal{L}$, $\mathcal{M}$, $\mathcal{G}$ and $\mathcal{F}$, which is hard to optimize. We optimize them by iterating three steps until convergence is reached. In the first step, labels of each superpixel $\mathcal{L}$, are obtained with the graph-cut algorithm [3] while keeping other variables fixed. In the second step, target state $\mathcal{B}$ is estimated based on the labels of superpixels with the remaining variables fixed. In the last step, appearance model $\mathcal{M}$, geometric graph $\mathcal{G}$ and correlation filters $\mathcal{F}$ are refined with the remain variables fixed. These three steps are carried out alternatively to generate a series of labelings and target states until a minimum of the overall energy in (2) is reached. We initialize $\mathcal{M}, \mathcal{G}, \mathcal{F}$ based on the first annotated state, as described in Section III-B. The overall optimization process is summarized in Algorithm 1.

**Algorithm 1** Optimization Algorithm for (2)

**Input:** A sequence with T frames and the first annotated state $\mathcal{B}_1$
**Output:** target state $\mathcal{B}_t^*$ at frame $t$ ($t \ge 2$)

```
initialize models M_1, G_1, F_1 based on annotated state B_1 in Section III-B
for frame t ∈ [2, T] do
    while overall energy E_t^all in (2) is decreasing do
        minimize (10) using fixed models M_t, G_t, F_t and state B_t to infer L_t in Section IV-A
        minimize E^SE(B, L, F) using fixed models F_t and L_t to estimate state B_t in Section IV-B
        minimize (11) using fixed labeling L_t and state B_t to update models M_t, G_t, F_t in Section IV-C
    end while
end for
```

### A. Inferring $\mathcal{L}$

Fixing $\mathcal{M}, \mathcal{G}, \mathcal{F}$ and $\mathcal{B}$, the first term in (9) is constant. Therefore the optimization of $\mathcal{L}$ is formulated as

$$\operatorname*{argmin}_{\mathcal{L}} \big\{E^{\mathrm{PS}}(\mathcal{L}, \mathcal{M}) + E^{\mathrm{PM}}(\mathcal{L}, \mathcal{G}) + \lambda^{\mathrm{c}} \sum_{p \in \mathcal{B}} (\mathbb{P}(\ell_p \in \mathbf{b}) - \mathbb{P}(\ell_p \in \mathbf{f}))\big\}. \quad (10)$$

As discussed in Section III-A, the part selection module aims to distinguish the target parts from the background parts, while the part matching module aims to get the correspondences between the target parts in current frames and previous frames. Both of them can be formulated as the label assignment problem. That is, we assign a label to each part to indicate which of the following categories it belongs to, *i.e.*, an existed target part in the previous frames, a newly generated target part, or a background part. In our framework, we integrate these two tasks in a unified objective (10) and complete them simultaneously. Thus, we only need to run the graph-cut algorithm [3] once to infer $\mathcal{L}$.

### B. Estimating $\mathcal{B}$

Given $\mathcal{L}$, the energy functions of part selection and matching in (2) are constants. The estimation of target state in the next frame reduces to find $\operatorname{argmin}_{\mathcal{B}} E^{\mathrm{SE}}(\mathcal{B}, \mathcal{L}, \mathcal{F})$. We optimize the target state using a dense sampling strategy. Similar to [7], we sample candidate states that vary in target center/scale. In practice, we first search target center $o_t$ based on the previous scale $s_{t-1}$ and subsequently target scale $s_t$. With fixed $\mathcal{L}$, we select the state corresponding to $\operatorname{argmin}_{\mathcal{B}} E^{\mathrm{SE}}(\mathcal{B}, \mathcal{L}, \mathcal{F})$.

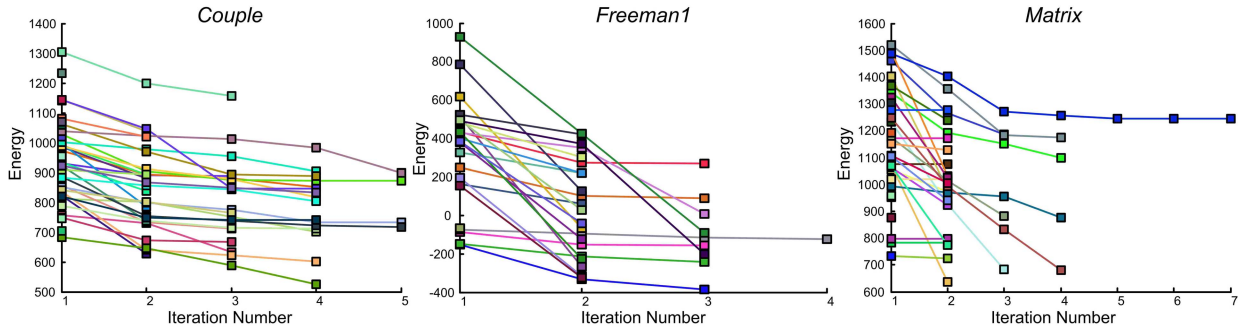



Fig. 3. Three examples in OTB2015 of the energy minimization process, *i.e.*, *Couple*, *Freeman1* and *Matrix*. Different frames are described by different colors. For clarity, the process of energy minimization in a few frames is illustrated.

### C. Learning $\mathcal{M}, \mathcal{G}, \mathcal{F}$

Fixing $\mathcal{L}$ and $\mathcal{B}$, the smooth energy $E^{\text{sm}}$ in (5) and the last two terms in (9) are constants. We can learn the models (*i.e.*, $\mathcal{M}, \mathcal{G}, \mathcal{F}$) by optimizing

$$\underset{\mathcal{M},\mathcal{G},\mathcal{F}}{\text{argmin}} \Big\{ \sum_{p\in\mathcal{P}} E^{\text{da}}(\ell_p, \mathcal{M}) + E^{\text{PM}}(\mathcal{L}, \mathcal{G}) - (1-\lambda^{\text{c}}) \cdot \mathcal{F}(\mathcal{B}, \mathcal{R}) \Big\}. \tag{11}$$

Since the three terms in (11) are independent, we optimize them separately. In particular, we update the learned models only if the overall energy in (11) is decreasing.

*1) Learning $\mathcal{M}$:* Based on the training samples generated by $\mathcal{L}$, we first learn a new appearance model $\widehat{\mathcal{M}}$, *i.e.*, SVM classifier $\widehat{\mathcal{C}}$ and HSV histogram $\widehat{\mathcal{H}}$. Then, we replace $\mathcal{M}$ with $\widehat{\mathcal{M}}$ if $\sum_{p\in\mathcal{P}} E^{\text{da}}(\ell_p, \widehat{\mathcal{M}}) < \sum_{p\in\mathcal{P}} E^{\text{da}}(\ell_p, \mathcal{M})$.

*2) Learning $\mathcal{G}$:* Along with the change of appearance and structure of the target, some existed target parts will be updated or occluded, and new target parts will be added. Based on $\mathcal{L}$, we update $\mathcal{G} = \{\mathcal{X}, \mathcal{E}\}$ by a simple heuristic strategy. To restrain background parts and recover missing target parts, we use four steps: *accept*, *merge*, *add* and *remove*:

- *Accept.* If target part $x$ is matched with superpixel $p$, we update the corresponding part based on $p$ if they have adjacent geometric centers, *i.e.*, $z(x) \leftarrow z(p), \zeta(x) \leftarrow \zeta(p)$, if $||z(p) - z(x)||_2 < d$.
- *Merge.* If label $f_x$ is assigned to $k$ number of superpixels $\{p_i\}_{i=1}^k$, we merge these superpixels as the new state of part $x$, *i.e.*, $z(x) \leftarrow \frac{1}{k} \cdot \sum_i z(p_i), \zeta(x) \leftarrow \frac{1}{k} \cdot \sum_i \zeta(p_i)$, such that $||z(p) - \frac{1}{k} \cdot \sum_i z(p_i)||_2 < d$.
- *Add.* For a non-matched superpixel $p$ in sparse region of target state, we add $p$ as a new target part if the distance between $p$ and other parts is larger than the threshold, *i.e.*, $\mathcal{X} \leftarrow \mathcal{X} \cup p$, if $\forall x \in \mathcal{X}, ||z(p) - z(x)||_2 > d$.
- *Remove.* If target part $x$ has not been matched with any superpixel for more than $N = 5$ frames, this part will be removed from the node set, *i.e.*, $\mathcal{X} \leftarrow \mathcal{X}/x$.

Along with the decrease of $E^{\text{PM}}(\mathcal{L}, \mathcal{G})$, we update the geometric graph $\mathcal{G}$ iteratively. By doing so, the newly generated target parts are used to search the most plausible target geometric graph $\mathcal{G}$ in the current frame at different iterations.

*3) Learning $\mathcal{F}$:* Centered at the current target center $o_t$, we generate a searching region with $\xi$ times of the current target size $s_t$ to calculate feature maps. Correlation filters $\mathcal{F}$ are updated based on the feature maps using an incremental update manner [7] if response score $\mathcal{F}(\mathcal{B}, \mathcal{R})$ in (11) increases.

In the above iterative optimization, we can obtain more accurate appearance model and geometric graph model (*i.e.*, we can remove some noisy parts in the iteration). If we do not carry out the iterative optimization, the error in the target geometric graph will accumulate in subsequent frames, resulting in target failure. For example, as shown in Fig. 9, if we do not optimize the graph iteratively, the tracker fail to track the target in sequence *Football1* after 10 frames.

### D. Convergence Analysis

The convergence of energy minimization in (2) can be guaranteed. First, the overall energy $E^{\text{all}}$ has a finite lower bound, *i.e.*, $E^{\text{all}} \geq -\sum_{p\in\mathcal{P}} \delta - 1$, which is proved as follows.

- In (4), we set average likelihood $\mathcal{H}(p, \ell_p)$ and SVM classification score $\mathcal{C}(p)$ less than a large constant $\delta$. Thus we have $E^{\text{da}}(\ell_p, \mathcal{M}) \geq \min(-|-\mathcal{C}(p) + \mathcal{H}(p, \ell_p)|/2, -|\mathcal{C}(p) + \mathcal{H}(p, \ell_p)|/2) \geq -|\mathcal{C}(p)|/2 - |\mathcal{H}(p, \ell_p)|/2 \geq -\delta$. Furthermore, we have $E^{\text{sm}}(\ell_p, \ell_q) \geq 0$ based on (5). In this way, we can get $E^{\text{PS}}(\mathcal{L}, \mathcal{M}) \geq -\sum_{p\in\mathcal{P}} \delta$.
- In (3), we can conclude $E^{\text{PM}}(\mathcal{L}, \mathcal{G}) \geq 0$, because the chi-squared distance $\chi^2(\zeta(p), \zeta(x)) \geq 0$ in (7), and geometric cost $E^{\text{ge}}(\ell_p, \ell_q) \geq 0$ based on (8).
- In (9), the response score $\mathcal{F}(\mathcal{B}, \mathcal{R})$, as well as the percentages $\sum_{p\in\mathcal{B}}(\mathbb{P}(\ell_p \in \mathbf{f})$ and $\sum_{p\in\mathcal{B}}(\mathbb{P}(\ell_p \in \mathbf{b}))$ range from 0 to 1. Therefore, we have $E^{\text{SE}}(\mathcal{B}, \mathcal{F}, \mathcal{L}) \geq -[(1-\lambda^{\text{c}}) \cdot 1 + \lambda^{\text{c}} \cdot (1-0)] = -1$.

Besides, the three steps in the optimization process update the variables $\mathcal{B}$, $\mathcal{L}$, $\mathcal{M}$, $\mathcal{G}$ and $\mathcal{F}$ based on lower energy. That is, at iteration $\tau$, we have $E^{\text{all}}(\tau) \leq E^{\text{all}}(\tau - 1)$. Meanwhile, the iteration will be terminated automatically if the overall energy is no longer reduced. It usually takes about $1 \sim 10$ iterations to reach the convergence, and some convergence examples are shown in Fig. 3.

## V. Experiments

We evaluate our method and compare it with state-of-the-art methods on three widely used benchmarks for object tracking, namely VOT2015 [21], OTB2013 [39] and OTB2015 [40].

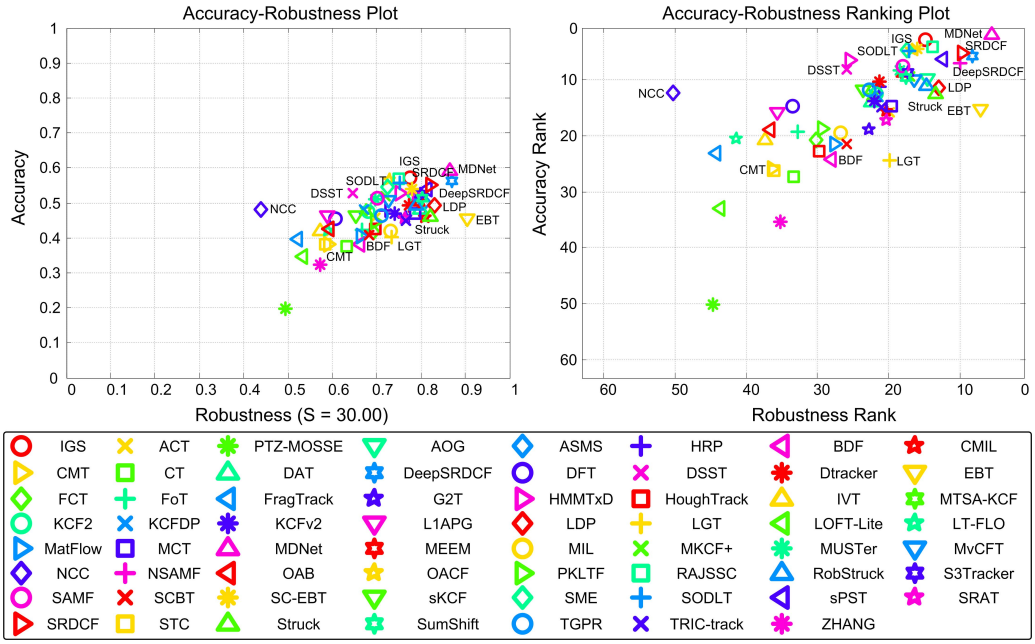



Fig. 4. The accuracy-robustness and its ranking plots for the VOT2015 benchmark [21]. One tracker close to the top-right boundary of the two plots achieves good performance. For clarity, a few compared trackers are marked in the plots. Please see [21] for the tracker references. This figure is best viewed in color.

The experimental results of other state-of-the-art trackers are provided from the authors or reproduced from the available source codes.

### *A. Implementation Details*

The IGS tracker is implemented using MATLAB[2]. The average running time for all testing sequences is 1.0 Frame-Per-Second (FPS) on a single thread for a machine with a 3.4 GHz Intel i7 processor and 8 GB memory.

We change one parameter and keep other ones fixed to set the optimal parameters empirically, and then all the parameters are fixed in the evaluated benchmarks. Some important parameters are discussed in Section V-E. The maximum number of iterations in optimization is set as $\tau_{\max} = 10$. The search ratio of the target region is set as $\xi = 1.5$, and the searching region is normalized to a fixed bounding box of size $150 \times 150$. The superpixel size in the region $m$ is set to 125. To keep the balance of the three components in (2), we set the scaling factor $\lambda^{\mathrm{b}} = 0.25$ in (3). The distance threshold for the target geometric graph in (7) is set as $\varepsilon = 3d$, where $d$ is the average diameter of superpixels in the searching region. For correlation filters, we use HOG [13] and color name [34] based feature maps, and the remaining parameters are set to be the same as in [7]. Scaling factor $\lambda^{\mathrm{l}}$ is set to 0.3 in (9).

### *B. Results on VOT2015*

The VOT2015 benchmark [21] has 60 sequences with various objects in challenging backgrounds, which are annotated by several visual attributes including illumination change, object size change and clutter. For evaluation, we employ two performance measures: *accuracy* and *robustness*. The accuracy measures how well the bounding box predicted by the tracker overlaps with the ground truth bounding box. The robustness measures how many times the overlap measure becomes zero (failures) in tracking.

In Fig. 4, we present the performance of IGS and compared it with all 62 trackers submitted to the VOT2015 Challenge. They are roughly categorized into four groups: Bounding Box (BB) based methods (*e.g.*, TGPR [14], MEEM [45], sPST [20], MUSTer [18], and EBT [48]), Correlation Filter (CF) based methods (*e.g.*, SAMF [22], MKCF+ [33], SRDCF [9], LDP [24], DSST [7], KCF2 [16] and OACF [2]), local based methods (*e.g.*, LGT [5], BDF [27], DAT [31] and CMT [29]), and Deep learning based methods (*e.g.*, DeepSRDCF [8], SODLT [36] and MDNet [28]).

From the Accuracy-Robustness plot, IGS tracks with a decent average overlap 0.57 during successful tracking periods, and the probability of IGS still tracking the target after $S = 30$ frames is approximately 0.77. Besides, IGS is close to the top-right boundary in the Accuracy-Robustness ranking plot, which suggests that it has good tradeoff between robustness and accuracy. Notably, MDNet (marked as pink triangle in the top-right corner of Accuracy-Robustness ranking plot) consistently produces the highest overlap and lowest failures. This is attributed to the CNN model trained with a large amount of additional tracking sequences.

*Quantitative Evaluation:* For comprehensive evaluation, Table I shows the quantitative results of all submitted methods over the VOT2015 benchmark[3]. From Table I, our tracker achieves the second best accuracy rank of 3.02, and the second best overlap of 0.57. In the following, the comparison with other tracking methods is discussed in details.

[2]We make the source codes of our tracker and comparison results available on our website: https://sites.google.com/site/daviddo0323/.

[3]All the results are calculated by the official evaluation kit in the website https://github.com/votchallenge/vot-toolkit.

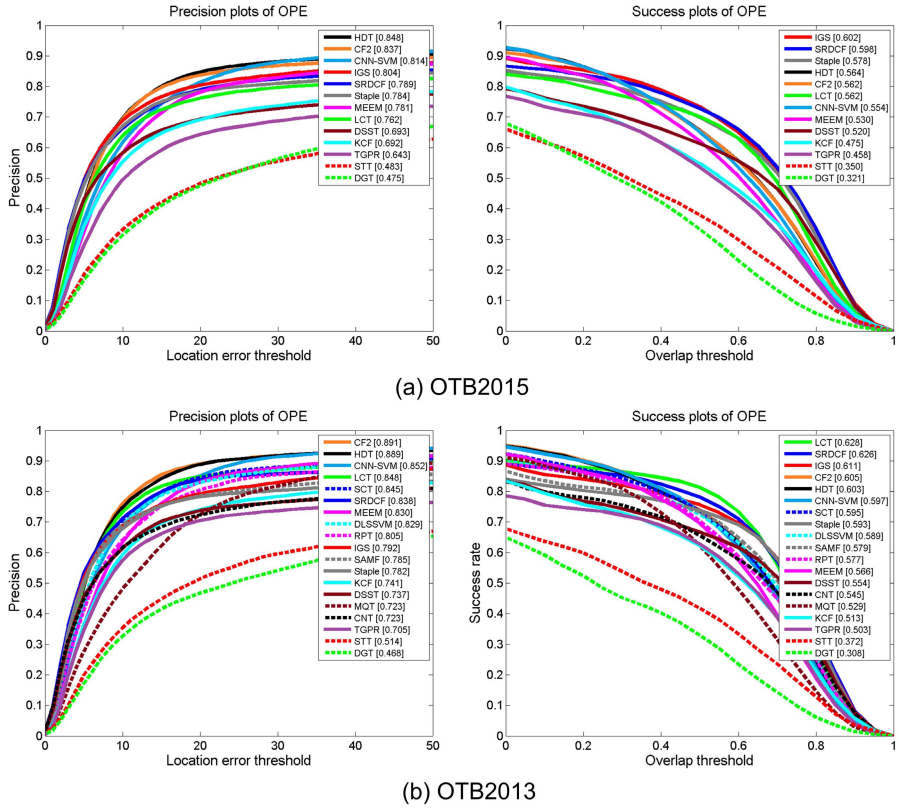



Fig. 5. The precision and success plots on the OTB2015/OTB2013 benchmark using OPE.

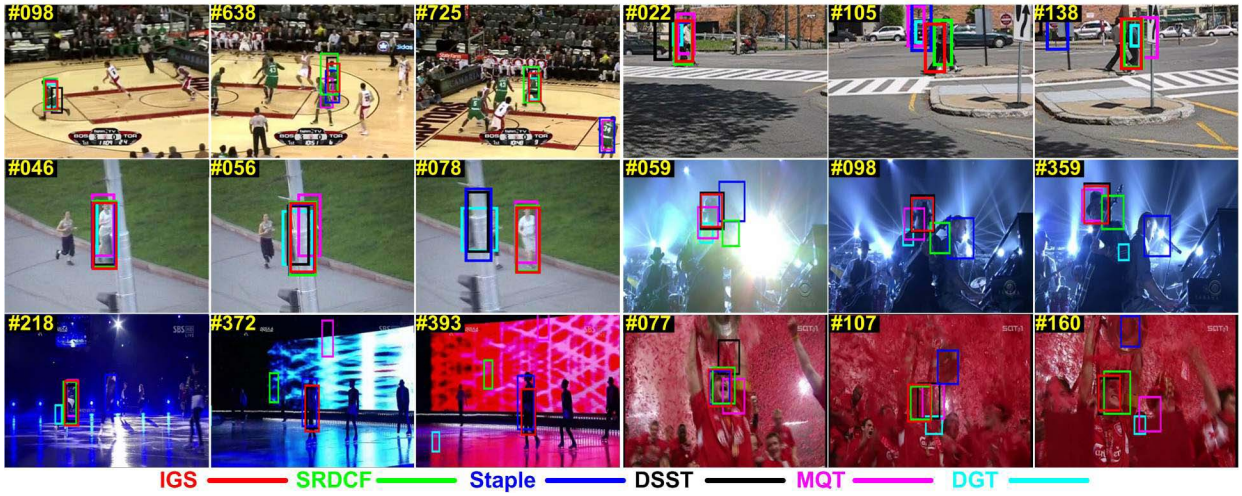



Fig. 6. Tracking results of the proposed IGS method, SRDCF [9], Staple [2], DSST [7], MQT [19] and DGT [4] on *Basketball*, *Couple*, *Jogging-2*, *Shaking*, *Skating1* and *Soccer*, denoted in different colors and lines.

First of all, MDNet and DeepSRDCF are the two top-performing trackers, which employ CNN features. Among the non-deep learning based methods, our method performs comparable with the state-of-the-art ones. Specifically, IGS shows a considerable improvement of overlap scores over several bounding box based methods (*e.g.*, Struck [15], TGPR [14], MEEM [45], and MUSTer [18]). When local appearance variation information is taken into account by IGS, the influence of object deformation in the tracking sequences is well reduced, as shown by the higher overlap score.

Furthermore, compared with CF based methods, IGS performs on par with SRDCF and against the remain ones (*e.g.*, SAMF [22], MKCF+ [33], LDP [24], DSST [7], KCF2 [16] and OACF [2]) in terms of accuracy and robustness ranks. Specifically, our method outperforms the baseline DSST method by 4% higher mean overlap score and 1.19 lower

TABLE I

RESULTS OF THE SUBMITTED TRACKERS IN THE VOT2015 CHALLENGE. "ACC." AND "ROB." ARE ABBREVIATIONS FOR "ACCURACY" AND "ROBUSTNESS", RESPECTIVELY. "↑" AND "↓" MEAN THAT LARGER OVERLAP AND SMALLER FAILURES AND RANK MEASURES MEAN BETTER PERFORMANCE. THE BEST, SECOND AND THIRD BEST RESULTS ARE SHOWN IN RED, GREEN AND BLUE FONTS, RESPECTIVELY

| Trackers | Category | Acc. Rank↓ | Rob. Rank↓ | Failures↓ | Overlap↑ |
|---|---|---|---|---|---|
| **IGS** | CF | 3.02 | 14.87 | 1.53 | 0.57 |
| **RAJSSC** | CF | 4.22 | 13.80 | 1.75 | 0.57 |
| **OACF** [2] | CF | 4.80 | 16.67 | 1.88 | 0.57 |
| **SRDCF** [9] | CF | 5.32 | 9.72 | 1.18 | 0.55 |
| **SME** | CF | 4.83 | 17.38 | 2.10 | 0.54 |
| **NSAMF** | CF | 7.18 | 10.05 | 1.45 | 0.53 |
| **DSST** [7] | CF | 8.25 | 25.95 | 2.72 | 0.53 |
| **MKCF+** [34] | CF | 9.47 | 17.03 | 1.97 | 0.52 |
| **MvCFT** | CF | 10.05 | 16.42 | 1.93 | 0.51 |
| **SAMF** [23] | CF | 7.78 | 18.05 | 2.08 | 0.51 |
| **LDP** [25] | CF | 11.58 | 13.05 | 1.30 | 0.49 |
| **KCF2** [17] | CF | 12.70 | 21.75 | 2.43 | 0.48 |
| **KCFv2** | CF | 13.83 | 22.03 | 2.03 | 0.47 |
| **sKCF** | CF | 11.88 | 23.63 | 2.75 | 0.46 |
| **ACT** | CF | 15.83 | 19.83 | 2.22 | 0.46 |
| **STC** | CF | 26.33 | 36.08 | 3.73 | 0.38 |
| **LOFT-Lite** | CF | 33.00 | 43.70 | 5.37 | 0.35 |
| **PTZ-MOSSE** | CF | 50.03 | 44.67 | 6.10 | 0.20 |
| **MDNet** [29] | deep | 2.05 | 5.55 | 0.77 | 0.59 |
| **DeepSRDCF** [8] | deep | 5.92 | 8.35 | 1.00 | 0.56 |
| **SODLT** [37] | deep | 4.95 | 17.18 | 1.81 | 0.56 |
| **sPST** [21] | BB | 6.48 | 12.48 | 1.42 | 0.54 |
| **SC-EBT** | BB | 4.60 | 16.08 | 1.72 | 0.54 |
| **HMMTxD** | BB | 6.65 | 25.52 | 2.25 | 0.53 |
| **MUSTer** [19] | BB | 8.45 | 18.43 | 2.20 | 0.51 |
| **MEEM** [46] | BB | 9.02 | 18.10 | 1.78 | 0.50 |
| **ASMS** | BB | 11.65 | 22.07 | 1.78 | 0.50 |
| **RobStruck** | BB | 11.25 | 14.80 | 1.58 | 0.49 |
| **Dtracker** | BB | 10.37 | 21.38 | 1.98 | 0.49 |
| **HRP** | BB | 11.32 | 21.27 | 2.28 | 0.48 |
| **MTSA-KCF** | BB | 13.17 | 22.37 | 2.45 | 0.48 |
| **KCFDP** | BB | 13.15 | 21.52 | 2.53 | 0.48 |
| **NCC** | BB | 12.40 | 50.27 | 8.18 | 0.48 |
| **MCT** | BB | 14.73 | 19.62 | 1.74 | 0.47 |
| **Struck** [16] | BB | 12.65 | 13.47 | 1.50 | 0.46 |
| **CMIL** | BB | 15.70 | 20.27 | 2.13 | 0.46 |
| **L1APG** | BB | 15.98 | 35.60 | 4.03 | 0.46 |
| **EBT** [49] | BB | 15.40 | 7.17 | 0.81 | 0.46 |
| **TGPR** [15] | BB | 11.98 | 22.77 | 2.24 | 0.46 |
| **SRAT** | BB | 17.35 | 20.25 | 1.98 | 0.45 |
| **DFT** | BB | 14.87 | 33.52 | 3.82 | 0.45 |
| **OAB** | BB | 18.92 | 36.63 | 3.92 | 0.43 |
| **MIL** | BB | 19.47 | 26.73 | 2.61 | 0.42 |
| **IVT** | BB | 20.83 | 37.38 | 4.10 | 0.42 |
| **SCBT** | BB | 21.58 | 25.88 | 2.64 | 0.41 |
| **CT** | BB | 27.28 | 33.37 | 3.56 | 0.38 |
| **S3Tracker** | local | 8.90 | 17.30 | 1.67 | 0.53 |
| **SumShift** | local | 9.72 | 17.58 | 1.62 | 0.51 |
| **AOG** | local | 9.85 | 14.57 | 9.85 | 0.51 |
| **DAT** [32] | local | 14.15 | 22.48 | 1.88 | 0.48 |
| **G2T** | local | 19.02 | 22.77 | 1.94 | 0.46 |
| **TRIC-track** | local | 15.07 | 20.97 | 1.99 | 0.45 |
| **PKLTF** | local | 18.73 | 29.33 | 2.62 | 0.45 |
| **HoughTrack** | local | 22.73 | 29.83 | 2.81 | 0.43 |
| **FCT** | local | 20.82 | 30.18 | 2.95 | 0.42 |
| **FoT** | local | 19.32 | 32.85 | 3.58 | 0.42 |
| **LT-FLO** | local | 20.68 | 41.40 | 3.94 | 0.42 |
| **MatFlow** | local | 21.52 | 27.58 | 2.82 | 0.41 |
| **LGT** [5] | local | 24.38 | 19.92 | 2.05 | 0.40 |
| **FragTrack** | local | 23.23 | 44.32 | 4.68 | 0.40 |
| **BDF** [28] | local | 24.35 | 28.00 | 2.83 | 0.38 |
| **CMT** [30] | local | 25.85 | 36.33 | 3.90 | 0.38 |
| **ZHANG** | local | 35.30 | 35.28 | 3.73 | 0.32 |

mean failures. These results show the effectiveness of combining local part selection and part matching, as well as global target state estimation.

In addition, IGS performs against several local based methods (*e.g.*, LGT [5], BDF [27], DAT [31], CMT [29]). For example, CMT obtains just 0.38 mean overlap score and 3.90 mean failures. This drop in performance is due to the reason that only local appearance information captured in these methods is insufficient to represent the object well especially in cluttered background.

### C. Results on OTB2013 and OTB2015

We conduct more experiments on the OTB2015 [40] (100 sequences) and OTB2013 [39] (51 sequences) benchmarks. The performances of compared methods are quantitatively validated by success plot and precision plot. The success plot draws the percentage of successfully tracked frames vs. the bounding box overlap threshold, where area under the curve is defined as *success score* for ranking. The precision plot illustrates the percentage of frames whose tracked centers are within the given threshold distance to the groundtruth, where *precision score* with the threshold equal to 20 pixels of distance error is used to rank the trackers. Specifically, the overlap score in the success plot is defined as overlap $= \text{Area}(\mathcal{B}_\text{T} \cap \mathcal{B}_\text{G})/\text{Area}(\mathcal{B}_\text{T} \cup \mathcal{B}_\text{G})$, where $\mathcal{B}_\text{T}$ is the tracked bounding box and $\mathcal{B}_\text{G}$ the groundtruth bounding box. The distance error in the precision plot is the center position error between $\mathcal{B}_\text{T}$ and $\mathcal{B}_\text{G}$.

By running the One Pass Evaluation (OPE) [39], we show the success and precision plots in Figure 5. Specifically, IGS is compared to state-of-the-art methods, including BB based methods (STT [37], MEEM [45], TGPR [14]), CF based methods (DSST [7], SAMF [22], SRDCF [9], LCT [26], KCF [16], Staple [2] and SCT [6]), local based methods (DGT [4], RPT [23], MQT [19], and CNT [46]) and deep learning based methods (CNN-SVM [17], CF2 [25], and HDT [32]).

Compared to the baseline CF based DSST tracker[4] with 0.520 success score and 0.693 precision score, IGS has about 8% and 11% improvements, respectively. Also, IGS performs better or on par with regards to other related trackers (*i.e.*, SAMF [22], SRDCF [9], LCT [26], KCF [16], Staple [2], and SCT [6]) on OTB2015 and OTB2013.

*Attribute-Based Evaluation:* Since OTB2013 is the subset of OTB2015, we present the success plots of 11 attributes (*e.g.*, occlusion, motion blur, and low resolution) on the full OTB2015 benchmark in Fig. 7 to analyze the performance of the proposed algorithm thoroughly in various aspects. In addition, qualitative comparison with several related trackers is shown in Fig. 6 for better visual inspection.

According to Fig. 6, DGT suffers from the low discriminative power of local superpixel representation, resulting in ambiguous matches with target parts in cluttered background such as *Shaking* and *Skating1*. MQT also performs not well for background clutters because of lack of global appearance model with strong disacriminablitiy. Besides, DSST only focuses on global appearance representation, leading to inferior performance in case of deformation (*e.g.*, *Basketball*

[4]We also evaluate the improved DSST method using the same feature maps as that in the proposed IGS tracker, *i.e.*, HOG features and color name. The improved DSST method achieves 0.519 success score and 0.690 precision score, which is similar to the original DSST method. Therefore, we do not add it in this section.

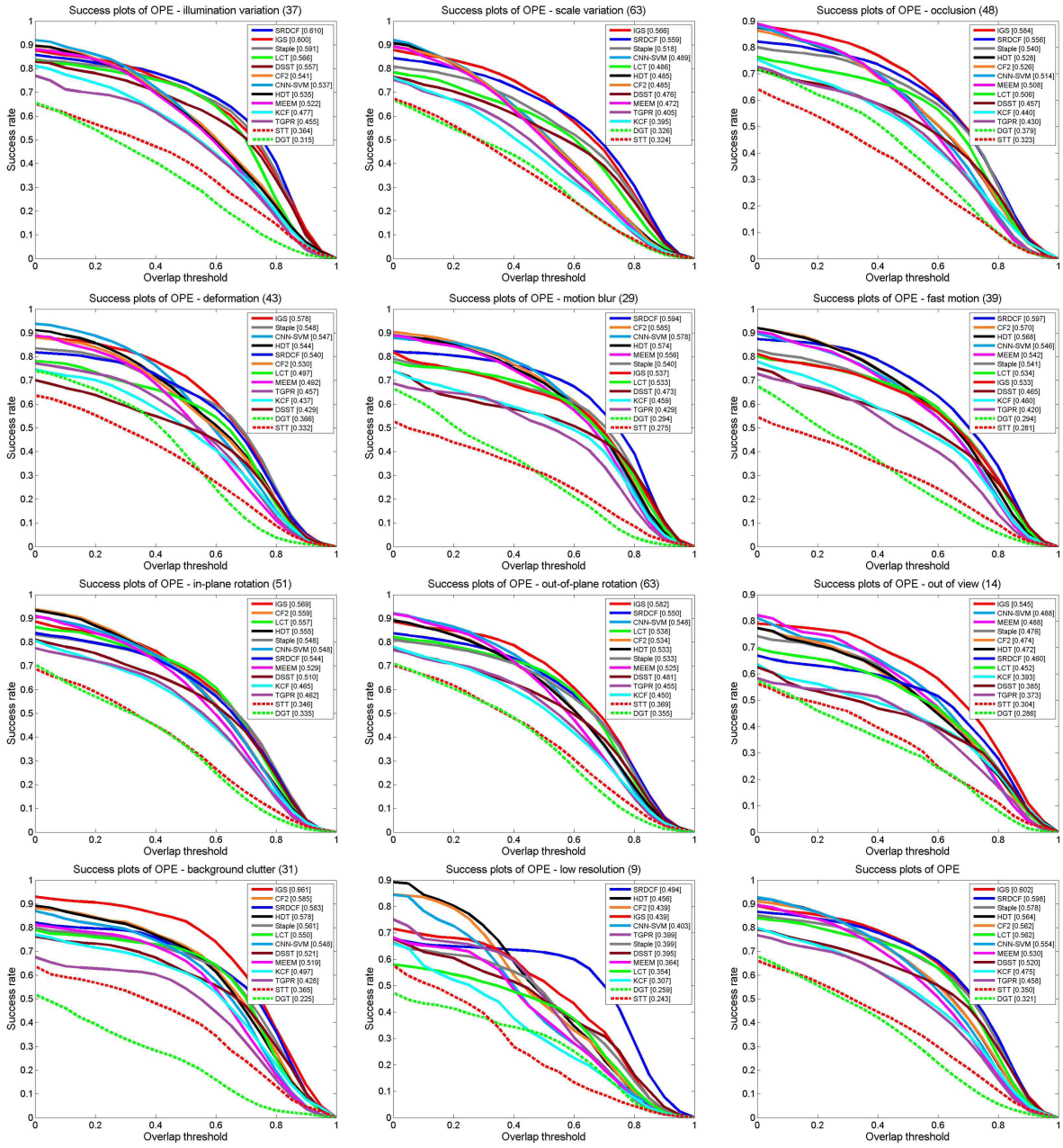



Fig. 7. Attribute-based analysis on the OTB2015 benchmark.

#638) and occlusion (*e.g.*, *Jogging-2* #076). To this end, Staple achieves a large gain over DSST in case of scale variation and deformation by combining color statistics and correlation filters. Since it gives little consideration on the geometric structure of the target, its performance is inferior in the presence of in-plane rotation, out-of-plane rotation and background clutter (*e.g.*, *Basketball* #725, *Couple* #105 and *Soccer* #107). SRDCF significantly improves the tracking performance under illumination variation and background clutter by learning the correlation filter coefficients with spatial regularization. However, global geometric constraint in SRDCF is sensitive to handle the target with out-of-view and in-plane rotation (*e.g.*, *Shaking* #059 and *Skating1* #372).

As shown in Fig. 7, compared to the two most competitive methods (*i.e.*, Staple and SRDCF), IGS performs well in all the aforementioned attributes, and the best in 7 out of the 11 attributes. This may be attributed to two factors in IGS. Firstly, target part selection, part matching and state estimation are combined in a unified framework to support each other and improve the performance. Moreover, the most plausible geometric graph is sought with the constraint of three components in the optimization by minimizing the overall energy function. Thus missing target parts can be recovered and redundant background parts are removed effectively even in cluttered background.

### *D. Failure Cases*

Our method is based on traditional CF based trackers such as KCF and DSST, which suffer from boundary effects [9]. Therefore, IGS may not work well in the presence of motion


Fig. 8. Three Failure cases of IGS in OTB2015, *i.e.*, *Biker*, *Kitesurf* and *Ironman*. The red and green bounding box denote our tracking result and groundtruth, respectively.

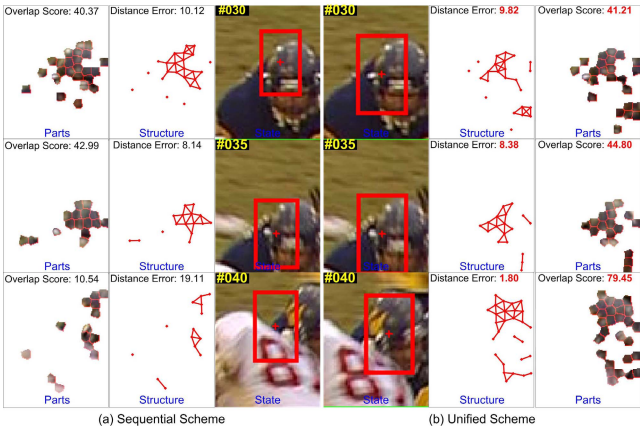



Fig. 9. Comparison with different iterations (*i.e.*, $\tau_{\max} = 1$ in the left and $\tau_{\max} = 10$ in the right) in sequence *Football1*.

blur or fast motion. Besides, our method gives less accurate bounding box in case of low resolution videos, due to the small size of the target. Failure cases in OTB2015 are shown in Fig. 8.

### *E. Discussion*

We further perform experiments to study the effect of different aspects of IGS on the tracking performance. We choose all sequences from the OTB2013 benchmark to conduct the experiments. The average performance and corresponding running speeds of IGS variants are summarized in Table II.

*1) Effectiveness of Iterative Optimization:* In Table II, we present the performance of IGS with different iterations, *i.e.*, IGS_$\tau_{\max}(i)$, $(i = 1, 5, 10, \infty)$, where $\tau_{\max}$ is the maximal number of iterations. If $\tau_{\max} = 1$, the three components in (2) are considered sequentially and do not share the complementary information. If $\tau_{\max} = \infty$, formulation (2) is optimized iteratively until the energy is no longer reduced. The results in Table II show that considerable performance improvements (*i.e.*, larger overlap score and smaller distance error) are achieved in more iterations. However, too many iterations can slightly reduce performance because of the overfitting of the appearance model.

TABLE II
TRACKING PERFORMANCES OF IGS VARIANTS

| IGS variants | Success score | Precision score | Speed (FPS) |
|---|---|---|---|
| IGS_$\tau_{\max}(1)$ | 0.526 | 0.697 | **1.667** |
| IGS_$\tau_{\max}(5)$ | 0.571 | 0.753 | 1.040 |
| IGS_$\tau_{\max}(10)$ | **0.611** | **0.792** | 1.034 |
| IGS_$\tau_{\max}(\infty)$ | 0.606 | 0.788 | 1.032 |
| IGS_$m(75)$ | 0.558 | 0.749 | 0.640 |
| IGS_$m(100)$ | 0.574 | 0.761 | 0.874 |
| IGS_$m(125)$ | **0.611** | **0.792** | 1.034 |
| IGS_$m(150)$ | 0.585 | 0.780 | 1.153 |
| IGS_$m(175)$ | 0.570 | 0.753 | 1.355 |
| IGS_$m(200)$ | 0.576 | 0.744 | 1.427 |
| IGS_$\lambda^c(0.1)$ | 0.566 | 0.743 | 0.932 |
| IGS_$\lambda^c(0.2)$ | 0.587 | 0.769 | 0.953 |
| IGS_$\lambda^c(0.3)$ | **0.611** | **0.792** | **1.034** |
| IGS_$\lambda^c(0.4)$ | 0.567 | 0.738 | 0.900 |
| IGS_$\varepsilon(1d)$ | 0.597 | **0.792** | 0.932 |
| IGS_$\varepsilon(2d)$ | 0.576 | 0.767 | 0.959 |
| IGS_$\varepsilon(3d)$ | **0.611** | **0.792** | **1.034** |
| IGS_$\varepsilon(4d)$ | 0.568 | 0.737 | 0.991 |
| IGS_w/o_PS | 0.568 | 0.737 | **1.181** |
| IGS_w/o_PM | 0.516 | 0.677 | 1.134 |
| IGS_w/o_SE | 0.292 | 0.406 | 1.122 |
| IGS | **0.611** | **0.792** | 1.034 |

As shown in Fig. 1, the target geometric graph is optimized to remove noisy parts in the iteration. If we do not carry out the iterative optimization, the error in the target geometric graph will accumulate in subsequent frames, resulting in target failure. Similarly, as shown in Fig. 9, if we do not optimize the graph iteratively, the tracker fails to track the target in sequence *Football1* after 10 frames. In practice, it usually takes no more than 10 iterations to solve (2) in most sequences. Therefore, we set the maximal iteration as $\tau_{\max} = 10$, which speeds up the optimization process while maintaining good accuracy.

*2) Influence of Superpixel Size:* We vary the superpixel size $m = \{75, 100, 125, 150, 175, 200\}$ to evaluate its influence on the tracking performance. From Table II, it is difficult to exploit accurate geometric structure from superpixels with large size (*e.g.*, $m = 200$). Small superpixel size means more superpixels in the searching region, which increases the computational complexity considerably and reduces the discriminability for each part (*e.g.*, $m = 75$). Considering the performance of the tracker, $m = 125$ in our implementation.

*3) Influence of Correlation Filters:* We analyze the influence of correlation filters by enumerating the scaling factor $\lambda^c = \{0.1, 0.2, 0.3, 0.4\}$ in (9). Based on Table II, we find that the two scores will increase by augmenting $\lambda^c$ before the optimal performance ($\lambda^1 = 0.3$) is achieved. However, the performance does not monotonically increase. This maybe because local appearance representation is also crucial to the success of object tracking especially in deformation and occlusion.

*4) Influence of Neighboring Distance:* To study the influence of neighboring distance in (7), we report the performance with different distance thresholds $\varepsilon = \{1d, 2d, 3d, 4d\}$, where $d$ is the average diameter of superpixel in the current searching region. Experimental evaluations given in Table II show that good performance can be achieved when we take $\varepsilon = 3d$.

*5) Effectiveness of Each Component:* To demonstrate the effectiveness of each term in (2), we construct three variants

of IGS. Specifically, IGS_w/o_PS indicates that the tracker is performed without part selection, *i.e.*, $E^{\mathrm{PS}}(\mathcal{L}, \mathcal{M}) = 0$. IGS_w/o_PM means that part matching is not included in the framework, *i.e.*, $E^{\mathrm{PM}}(\mathcal{L}, \mathcal{G}) = 0$. IGS_w/o_SE denotes that target state is estimated only by part selection and part matching, *i.e.*, $\mathcal{F}(\mathcal{B}, \mathcal{R}) = 0$. Based on Table II, IGS considering all the components shows a significant improvement of the performance over its variants. If part selection is excluded, part matching is conducted with all superpixels in the searching region, and hence the estimated target state is less accurate. Without constraining the geometric relations between parts, extracted target parts have reduced discriminability, resulting in drifting away in complex scenes. If there is no global constraint to the correlation filters $\mathcal{F}$, the performance is severely degraded only based on local appearance variations. Using all the three components is critical to the accuracy of tracking results for IGS.

## VI. Conclusion

In this paper, we propose an iterative graph seeking based object tracking method, which integrates target part selection, part matching, and state estimation using a unified energy minimization framework. We develop an alternative iteration scheme to minimize the energy function efficiently. Along with the decrease of the overall energy, the target geometric graph is refined iteratively to capture target structure in local parts variations using the global constraint. Experiments on several challenging benchmarks demonstrate the effectiveness of the proposed method compared to state-of-the-arts.

There are several future works we would like to further investigate. Firstly, it has been shown that using hypergraph can better capture higher-order dependencies from geometric structures and motion smoothness [10], [38]. Secondly, recent studies have shown that CNN features can significantly improve the tracking performance [8], [28], [32]. We will expand our method to incorporate such information into the proposed framework. Thirdly, we will optimize the code to improve the run time efficiency of the IGS tracker so that it can run in real-time.

## References


[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.

[2] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. S. Torr, "Staple: Complementary learners for real-time tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1401–1409.

[3] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1222–1239, Nov. 2001.

[4] Z. Cai, L. Wen, Z. Lei, N. Vasconcelos, and S. Z. Li, "Robust deformable and occluded object tracking with dynamic graph," *IEEE Trans. Image Process.*, vol. 23, no. 12, pp. 5497–5509, Dec. 2014.

[5] L. Cehovin, M. Kristan, and A. Leonardis, "Robust visual tracking using an adaptive coupled-layer visual model," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 4, pp. 941–953, Apr. 2013.

[6] J. Choi, H. J. Chang, J. Jeong, Y. Demiris, and J. Y. Choi, "Visual tracking using attention-modulated disintegration and integration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 4321–4330.

[7] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *Proc. Brit. Mach. Vis. Conf.*, 2014.

[8] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Convolutional features for correlation filter based visual tracking," in *Proc. Workshops Conjunct IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 621–629.

[9] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 4310–4318.

[10] D. Du, H. Qi, W. Li, L. Wen, Q. Huang, and S. Lyu, "Online deformable object tracking based on structure-aware hyper-graph," *IEEE Trans. Image Process.*, vol. 25, no. 8, pp. 3572–3584, Aug. 2016.

[11] D. Du, H. Qi, L. Wen, Q. Tian, Q. Huang, and S. Lyu, "Geometric hypergraph learning for visual tracking," *IEEE Trans. Cybern.*, vol. 47, no. 12, pp. 4182–4195, Dec. 2017.

[12] S. Duffner and C. Garcia, "PixelTrack: A fast adaptive algorithm for tracking non-rigid objects," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2480–2487.

[13] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

[14] J. Gao, H. Ling, W. Hu, and J. Xing, "Transfer learning based visual tracking with gaussian processes regression," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 188–203.

[15] S. Hare, A. Saffari, and P. H. S. Torr, "Struck: Structured output tracking with kernels," in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2011, pp. 263–270.

[16] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, Mar. 2015.

[17] S. Hong, T. You, S. Kwak, and B. Han, "Online tracking by learning discriminative saliency map with convolutional neural network," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 597–606.

[18] Z. Hong, Z. Chen, C. Wang, X. Mei, D. V. Prokhorov, and D. Tao, "Multi-store tracker (MUSTer): A cognitive psychology inspired approach to object tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 749–758.

[19] Z. Hong, C. Wang, X. Mei, D. Prokhorov, and D. Tao, "Tracking using multilevel quantizations," in *Proc. Eur. Conf. Comput. Vis.*, vol. 8694. 2014, pp. 155–171.

[20] Y. Hua, K. Alahari, and C. Schmid, "Online object tracking with proposal selection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 3092–3100.

[21] M. Kristan *et al.*, "The visual object tracking VOT2015 challenge results," in *Proc. Workshops Conjunct IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 564–586.

[22] Y. Li and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," in *Proc. Workshops Conjunct Eur. Conf. Comput. Vis.*, 2014, pp. 254–265.

[23] Y. Li, J. Zhu, and S. C. H. Hoi, "Reliable patch trackers: Robust visual tracking by exploiting reliable patches," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 353–361.

[24] A. Lukežič, L. Č. Zajc, and M. Kristan, "Deformable parts correlation filters for robust visual tracking," *IEEE Trans. Cybern.*, to be published.

[25] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Hierarchical convolutional features for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 3074–3082.

[26] C. Ma, X. Yang, C. Zhang, and M.-H. Yang, "Long-term correlation tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 5388–5396.

[27] M. E. Maresca and A. Petrosino, "Clustering local motion estimates for robust and efficient object tracking," in *Proc. Workshops Conjunct Eur. Conf. Comput. Vis.*, 2014, pp. 244–253.

[28] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 4293–4302.

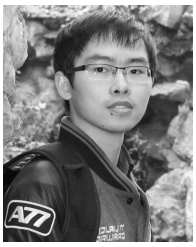
[29] G. Nebehay and R. Pflugfelder, "Clustering of static-adaptive correspondences for deformable object tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 2784–2791.

[30] S. Oron, A. Bar-Hillel, D. Levi, and S. Avidan, "Locally orderless tracking," *Int. J. Comput. Vis.*, vol. 111, no. 2, pp. 213–228, Jan. 2015.

[31] H. Possegger, T. Mauthner, and H. Bischof, "In defense of color-based model-free tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 2113–2120.

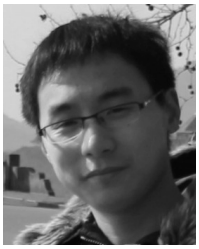
[32] Y. Qi *et al.*, "Hedged deep tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 4303–4311.

[33] M. Tang and J. Feng, "Multi-kernel correlation filter for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 3038–3046.
[34] J. van de Weijer, C. Schmid, J. Verbeek, and D. Larlus, "Learning color names for real-world applications," *IEEE Trans. Image Process.*, vol. 18, no. 7, pp. 1512–1523, Jul. 2009.
[35] J. Wang and Y. Yagi, "Many-to-many superpixel matching for robust tracking," *IEEE Trans. Cybern.*, vol. 44, no. 7, pp. 1237–1248, Jul. 2014.
[36] N. Wang, S. Li, A. Gupta, and D.-Y. Yeung, "Transferring rich feature hierarchies for robust visual tracking," *CoRR*, vol. abs/1501.04587, 2015. [Online]. Available: http://arxiv.org/abs/1501.04587
[37] L. Wen, Z. Cai, Z. Lei, D. Yi, and S. Z. Li, "Robust online learned spatio-temporal context model for visual tracking," *IEEE Trans. Image Process.*, vol. 23, no. 2, pp. 785–796, Feb. 2014.
[38] L. Wen, Z. Lei, S. Lyu, S. Z. Li, and M.-H. Yang, "Exploiting hierarchical dense structures on hypergraphs for multi-object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 10, pp. 1983–1996, Oct. 2016.
[39] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2411–2418.
[40] Y. Wu, J. Lim, and M. H. Yang, "Object tracking benchmark," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1834–1848, Sep. 2015.
[41] F. Yang, H. Lu, and M.-H. Yang, "Robust superpixel tracking," *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1639–1651, Apr. 2014.
[42] R. Yao, Q. Shi, C. Shen, Y. Zhang, and A. van den Hengel, "Part-based visual tracking with online latent structural learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2363–2370.
[43] X. Yu, J. Yang, T. Wang, and T. Huang, "Key point detection by max pooling for tracking," *IEEE Trans. Cybern.*, vol. 45, no. 3, pp. 430–438, Mar. 2015.
[44] Y. Yuan, J. Fang, and Q. Wang, "Robust superpixel tracking via depth fusion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 1, pp. 15–26, Jan. 2014.
[45] J. Zhang, S. Ma, and S. Sclaroff, "MEEM: Robust tracking via multiple experts using entropy minimization," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 188–203.
[46] K. Zhang, Q. Liu, Y. Wu, and M.-H. Yang, "Robust visual tracking via convolutional networks without training," *IEEE Trans. Image Process.*, vol. 25, no. 4, pp. 1779–1792, Apr. 2016.
[47] L. Zhao, X. Li, J. Xiao, F. Wu, and Y. Zhuang, "Metric learning driven multi-task structured output optimization for robust keypoint tracking," in *Proc. Conf. Arti. Intell.*, 2015, pp. 3864–3870.
[48] G. Zhu, F. Porikli, and H. Li, "Tracking randomly moving objects on edge box proposals," *CoRR*, vol. abs/1507.08085, 2015. [Online]. Available: http://arxiv.org/abs/1507.08085



**Dawei Du** received the B.Eng. degree in automation and the M.S. degree in detection technology and automatic engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2010 and 2013, respectively. He is currently pursuing the Ph.D. degree with the School of Computer and Control Engineering, University of Chinese Academy of Sciences, Beijing, China.

His current research interests include hypergraph representation, visual tracking, video segmentation, and deep learning.



**Longyin Wen** (M'15) received the B.Eng. degree in automation from the University of Electronic Science and Technology of China, Chengdu, China, in 2010, and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2015.

He is currently a Computer Vision Scientist with GE Global Research, Niskayuna, NY, USA. He was a Post-Doctoral Researcher with University at Albany, State University of New York, Albany, NY, USA, from 2015 to 2016. His current research interests include computer vision, pattern recognition, and video analysis.

**Honggang Qi** (M'14) received the M.S. degree in computer science from Northeast University, Shenyang, China, in 2002, and the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2008.

He is currently an Associate Professor with the School of Computer and Control Engineering, University of Chinese Academy of Sciences. His current research interests include computer vision, video coding, and very large scale integration design.

**Qingming Huang** (F'17) received the B.S. degree in computer science and the Ph.D. degree in computer engineering from the Harbin Institute of Technology, Harbin, China, in 1988 and 1994, respectively.

He is currently a Professor and the Deputy Dean of the School of Computer and Control Engineering, University of Chinese Academy of Sciences, Beijing, China. He has authored over 300 academic papers in international journals, such as the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON MULTIMEDIA, and the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, and top level international conferences, including the ACM Multimedia, International Conference on Computer Vision, Computer Vision and Pattern Recognition, European Conference on Computer Vision, International Conference on Very Large Data Bases, and International Joint Conference on Artificial Intelligence. His current research interests include multimedia computing, image/video processing, pattern recognition, and computer vision.

**Qi Tian** (F'16) received the B.Eng. degree in electronic engineering from Tsinghua University, Beijing, China, in 1992, the M.S. degree in electrical and computer engineering from Drexel University, Philadelphia, PA, USA, in 1996, and the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana–Champaign, Champaign, IL, USA, in 2002.

He took a one-year faculty leave at Microsoft Research Asia, Beijing, China, from 2008 to 2009. He is currently a Full Professor with the Department of Computer Science, The University of Texas at San Antonio, San Antonio, TX, USA. He has published over 360 refereed journal and conference papers. His current research interests include multimedia information retrieval and computer vision.

Dr. Tian is currently an Associate Editor of the IEEE TRANSACTIONS ON MULTIMEDIA, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, and the *Multimedia System Journal*. and he is on the Editorial Board of the *Journal of Multimedia* and the *Journal of Machine Vision and Applications*. He is a Guest Editor of the IEEE TRANSACTIONS ON MULTIMEDIA and the *Journal of Computer Vision and Image Understanding*.

**Siwei Lyu** (SM'16) received the B.S. degree in information science and the M.S. degree in computer science from Beijing University, Beijing, China, in 1997 and 2000, respectively, and the Ph.D. degree in computer science from Dartmouth College, Hanover, NH, USA, in 2005.

He was a Post-Doctoral Research Associate with the Center for Neural Science, Howard Hughes Medical Institute, New York University, New York, NY, USA, and an Assistant Researcher with Microsoft Research Asia, Beijing. He is currently an Associate Professor with the Computer Science Department, University at Albany, State University of New York (SUNY), Albany, NY, USA, and a Visiting Professor with the School of Computer and Information, Tianjin Normal University, Tianjin, China. He has authored one book, one book chapter, and over 70 refereed journal and conference papers. His research projects are funded by NSF, NIJ, UTRC, IBM, and University at Albany, SUNY. His current research interests include digital image forensics, computer vision, and machine learning.

Dr. Lyu was a recipient of the IEEE Signal Processing Society Best Paper Award in 2011 and the U.S. NSF CAREER Award in 2010.