Geometric Hypergraph Learning for Visual Tracking

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

***Abstract*—Graph-based representation is widely used in visual tracking field by finding correct correspondences between target parts in different frames. However, most graph-based trackers consider pairwise geometric relations between local parts. They do not make full use of the target's intrinsic structure, thereby making the representation easily disturbed by errors in pairwise affinities when large deformation or occlusion occurs. In this paper, we propose a geometric hypergraph learning-based tracking method, which fully exploits high-order geometric relations among multiple correspondences of parts in different frames. Then visual tracking is formulated as the mode-seeking problem on the hypergraph in which vertices represent correspondence hypotheses and hyperedges describe high-order geometric relations among correspondences. Besides, a confidence-aware sampling method is developed to select representative vertices and hyperedges to construct the geometric hypergraph for more robustness and scalability. The experiments are carried out on three challenging datasets (VOT2014, OTB100, and Deform-SOT) to demonstrate that our method performs favorably against other existing trackers.**

***Index Terms*—Confidence-aware sampling, correspondence hypotheses, deformation, geometric hypergraph learning, mode-seeking, occlusion, visual tracking.**

Manuscript received March 16, 2016; revised August 9, 2016; accepted October 28, 2016. Date of publication November 18, 2016; date of current version November 15, 2017. This work was supported in part by the National Natural Science Foundation of China under Grant 61620106009, Grant 61332016, Grant 61472388, and Grant 61429201, in part by the Key Research Program of Frontier Sciences, CAS under Grant QYZDJ-SSW-SYS013, in part by the ARO under Grant W911NF-15-1-0290, in part by the Faculty Research Gift Awards by NEC Laboratories of America and Blippar, and in part by the U.S. National Science Foundation Research Grant through the Division of Computing and Communication Foundations under Grant 1319800. This paper was recommended by Associate Editor H. Lu. *(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 was with University at Albany, State University of New York, Albany, NY 12222 USA. He is now with GE Global Research, NY 12309 USA (e-mail: longyin.wen@ge.com).

Q. Tian is with the Department of Computer Science, University of Texas at San Antonio, San Antonio, TX 78249-1604 USA (e-mail: qi.tian@utsa.edu).

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@jdl.ac.cn).

S. Lyu is with the Computer Science Department, University at Albany, State University of New York, Albany, NY 12222 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/TCYB.2016.2626275

I. Introduction

VISUAL tracking has attracted much research interest in computer vision field, because it is a critical step of various applications, including video surveillance, sport analysis, auto-drive car, etc. Despite having achieved promising progress over the past decade, it is still challenging to design a robust tracker that can handle appearance changes caused by various critical situations, such as large deformation, illumination variation, partial and full occlusion, and background clutter. In particular, the deformation and occlusion are the two most notable challenges that affect tracking performances.

For tracking scenarios where the target appearance is relatively stable, methods based on global appearance models can achieve satisfactory performances [9], [16], [18], [21], [22], [35], [49]. However, if large deformation and occlusion happen, such approaches usually fail to track the target robustly. To counter this problem, part-based approaches have received more attention [20], [32], [34], [36], [48]. Moreover, several different methods to represent the target geometric structure have been proposed, such as structural support vector machine (SVM) [46], Markov random field (MRF) [19], [33], keypoint constellation [30], [31], and graph model [6], [37]. However, most approaches just consider pairwise relations between target parts. The pairwise affinities are easily disturbed by errors, rendering difficulties to well preserve the geometric structure underlying the target representation.

In this paper, we present a novel geometric hypergraph tracker (GGT) to handle the visual tracking task, especially for the deformable targets. Different from most previous works that only consider pairwise geometric relations between local parts, our method exploits high-order relations using the *geometric hypergraph*. Specifically, the geometric hypergraph is constructed and learned to match the target part set and the candidate part set.[1] The possible correspondences between parts in the two sets are described by the *correspondence hypotheses*. Each vertex encodes a correspondence hypothesis, and hyperedges encode high-order geometric relations among correspondence hypotheses. Fig. 1(a) gives a schematic diagram of constructing the hypergraph. Thus, the target can be effectively characterized by extracting common appearance and geometric property of correspondences. Reliable correspondences lead to a large number of hyperedges with high

[1]The candidate part set Q consists of candidate parts extracted from the searching area in the current frame t. We employ the target part set P as the part representation of the target, which is consisted by the target parts up to the previous frame $t-1$.

2168-2267 © 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

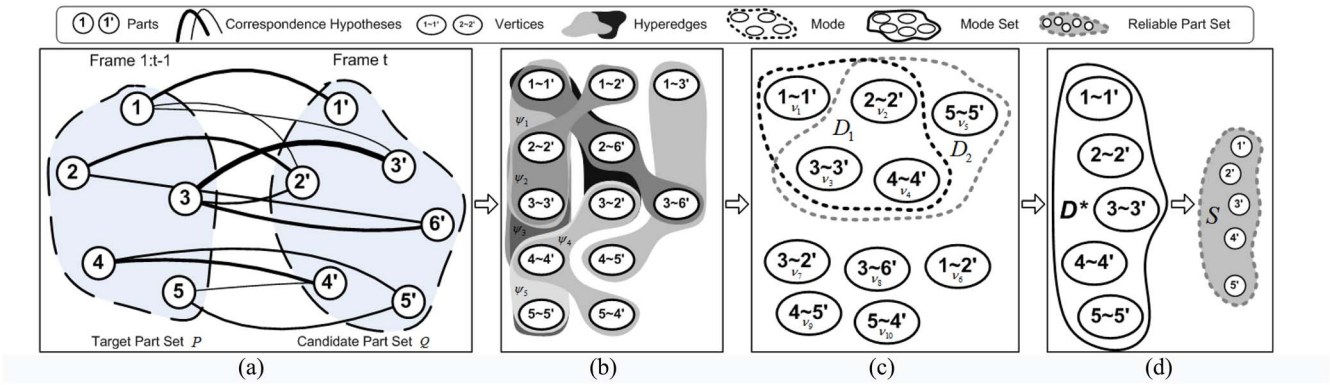

Fig. 1. Schematic illustration of our tracking framework. (a) Between the target part set P and the candidate part set Q, the correspondence hypotheses are generated and constrained by the appearance similarity (association confidence). The thickness of black lines denotes different similarity between two parts. (b) Vertices and hypergraphs of the geometric hypergraph are sampled from correspondence hypotheses based on the corresponding association confidences. For clarity, just a few vertices and hyperedges are shown. (c) Modes are extracted by searching the modes in the hypergraph (e.g., $\mathcal{D}_1$ and $\mathcal{D}_2$). (d) Reliable part set $\mathcal{S}$ is extracted from the mode set $\mathbf{D}^*$.

confidence; while other false correspondences are connected by very few hyperedges with low confidence. To reduce the computation complexity, a approximating hypergraph sampling strategy is used to extract the significant vertices and hyperedges. Moreover, we define the *mode* as a group of reliable correspondences between target parts with similar appearance and consistent geometric structure that are interconnected with the local maximum of confidences on the geometric hypergraph. Thus, the tracking problem is cast as a mode-seeking problem.

This paper makes the following contributions.

1) The geometric hypergraph is used to represent the target, which fully exploits high-order geometric relations among correspondence hypotheses in different frames.
2) The confidence-aware sampling method is proposed to approximate the hypergraph, which not only alleviates sensitivity to noises, but also is scalable to the large scale hypergraph. Thus, we seek modes on the hypergraph by the pairwise coordinate update method in [26] efficiently.
3) Our method is compared to existing methods on the challenging VOT2014, OTB100, and Deform-SOT dataset. The experimental results demonstrate the effectiveness and robustness of the proposed model.

II. Related Works

Tracking methods based on modeling relations between target parts have been shown to be less susceptible to the problem posed by object deformation and occlusion. Recently, many works have focused on how to incorporate geometric information as an important clue to facilitate visual tracking.

A. Keypoint-Based Tracking Methods

The keypoint-based trackers use the displacements of target parts to vote for the target center in consecutive frames. Hare *et al.* [17] proposed a keypoint-based tracking method, which combines feature matching, learning and object pose estimation into a coherent structured output learning framework. In [45], geometric structures are exploited among interest points by learning a structured visual dictionary. Yi *et al.* [47] performed tracking by combining each local feature of the target and the surroundings based on generalized Hough transform voting. Guo [15] formulated the tracking task as a maximum *a posteriori* estimation problem under the manifold representation, learned from local features preserving local appearance similarity and spatial structure. Nebehay and Pflugfelder [31] developed a keypoint-based tracking method in a combined matching-and-tracking framework, where each keypoint casts votes for the object center. Moreover, in [30], they employ a geometric dissimilarity measure to separate inlier correspondences from outliers. In [4], keypoints are considered as basic predictors localizing the target in a collaborative search strategy, where the persistence and the spatial consistency of a local feature are used to measure the most reliable features for tracking. Yu *et al.* [48] proposed a tracker to select discriminant keypoints by max pooling over the local descriptor responses from a set of filters. However, keypoint-based trackers focus on modeling the displacements between parts and the corresponding target center. It is insufficient to exploit relations between parts fully for geometric structure representation.

B. Part-Based Tracking Methods

To better solve the shape deformation and partial occlusion issue, part-based methods are gaining popularity in visual tracking. Wen *et al.* [38] presented a discriminative learning method to infer the position, shape and size of each part, using the Metropolis–Hastings algorithm integrated with an online SVM. The method of Wang and Nevatia [37] tracked nonrigid objects with multiple related parts, where the spatial relations among parts are formulated probabilistically. Improved from [16], Yao *et al.* [46] introduced an online latent structured learning-based tracking method, and use a global object box and a small number of part boxes to approximate the irregular object to reduce object drift. Cehovin *et al.* [7] employed a global representation to model target's global visual properties probabilistically. Meanwhile, the low-level patches are constrained and updated with the global model during tracking. A graph-based tracker is used in [6] to formulate the

tracking problem as subgraph matching between the geometric structure graph of the target and that of the candidate target proposals graph. Nam *et al.* [29] used a novel graphical model to adapt sequence structure and propagate the posterior over time. Hong *et al.* [19] proposed an MRF-based tracker to consider geometric structure by the hierarchical appearance representation across multilevel quantization (i.e., pixels, superpixels, and bounding boxes). Xie *et al.* [44] developed a local sparse coding-based tracking algorithm, and the complementary keypoint matching refinement can reject the incorrect matches and eliminate outliers for enhancing the tracking performance. In [27], a tracking method is proposed based on parts with multiple correlation filters, in which the structural constraint mask is adopted to handle various appearance changes. However, the existing part-based methods usually do not consider high-order geometric relations.

C. Segmentation-Based Tracking Methods

The segmentation-based methods consider the geometric information by finding out the precise location of each pixel in the target. Based on the generalized Hough-transform, Godec *et al.* [14] developed an improved online Hough Forests and couple the voting-based detection and back-projection with a rough segmentation based on GrabCut. Duffner and Garcia [12] presented a pixel-based nonrigid object tracking method, which consists of a generalized Hough transform with pixel-based descriptors-based detector and a probabilistic segmentation method based on a global model for foreground and background. Recently, Wen *et al.* [40] developed a joint online tracking and segmentation algorithm, which integrates the multipart tracking and segmentation into a unified energy optimization framework. Zhou *et al.* [53] proposed a level set tracking method, and introduce adaptive object shape modeling into the level set evolution process for more robustness in complex scenarios.

III. Methodology

We first introduce terms and notations to be used in the sequel. We denote the order of hypergraph as k. The hypergraph is a generalization of a graph in which an edge (hyperedge and strictly speaking) can connect more than k ($k \geq 3$) vertices, while a graph has its edges connecting two vertices. The unconnected graph is a graph without edges between vertices.

Our method is related with some previous works, including super-pixel tracker (SPT) [36], dynamic graph tracker (DGT) [6], and temporally coherent part based tracker (TCP) [24]. However, there are three aspects differing our method and these methods.

1) Though both our method and SPT use superpixel representation, our method provides complementary geometric information extracted from correspondence hypotheses. Then, both the superpixel representation and geometric constraint expect to improve performance on complex scenes. When the hypergraph degenerates into a disconnected graph ($k = 1$), SPT can be regarded as a special case of the proposed algorithm (see Section III-E1).
2) DGT uses a graph to exploit pairwise geometric relations between neighboring parts and matches the superpixels by spectral technique. On the contrary, our method employs a hypergraph that considers k-order geometric relations among correspondence hypotheses. When $k = 2$, our method is similar to DGT (see Section III-E2).
3) TCP and our method share similar hypergraph model, yet they have different motivations. TCP mainly exploits *temporal* high-order relations among parts in consecutive frames, ignoring the high-order geometric structure information. Our method focuses on modeling the *spatial* high-order relations among correspondence hypotheses. Moreover, the optimization method in TCP just weighs each part in the temporal hypergraph equally, but it is more reasonable to assign different confidences to the correspondence hypotheses in our method. Therefore, the optimization method for our formulation is improved from the one in [24] by adding the *association confidence term*, as presented in Section III-C.

To sum up, the related previous methods do not usually consider the high-order geometric structure information, and this is the most important characteristics of our method distinguishing other methods.

A. Geometric Hypergraph

The superpixel representation is more flexible for the deformable target compared to the holistic representation, while has low discriminability because of small size. Therefore, we construct a geometric hypergraph to alleviate the problem with geometric constraints. Given the annotated bounding box in the first frame, the target part set P is initialized, and the candidate part set Q is determined by the coarse labeling of superpixels in the rest frames.[2] Based on P and Q, we construct the vertex set $\mathcal{V}$ and hyperedge set Ψ of the geometric hypergraph $\mathcal{G}$ as

$$\begin{cases} \mathcal{V} = \{v_i\}_{i=1}^{N} = \{p \sim q | \forall p \in P, q \in Q : d_E(p,q) \leq \tau_d\} \\ \Psi = \{\psi | \forall v_i, v_j \in \psi \ : \ v_i \cap v_j = \emptyset\} \end{cases} \tag{1}$$

where N is the number of vertices. v_i and v_j are the ith and jth vertex in hyperedge $\psi = \{v_1, \ldots, v_k\}$ without conflicts or duplicates. $d_E(p,q)$ is the Euclidean distance between the centers of parts p and q in image plane. The distance threshold is set as $\tau_d = 3\sqrt{W \cdot H/\rho}$, where ρ is the number of superpixels in the searching area with width W and height H in the current frame, as shown in Fig. 2(b).

B. Formulation

As analyzed in the introduction, multiple correspondences with similar structural geometric properties form a mode.

[2]Similar as [6], we first use the SLIC algorithm [1] to over-segment the searching area of the target into multiple parts (superpixels), and employ the graph cut method [5] to coarsely separate the foreground parts from the background, as shown in the top-left of Fig. 2(a).

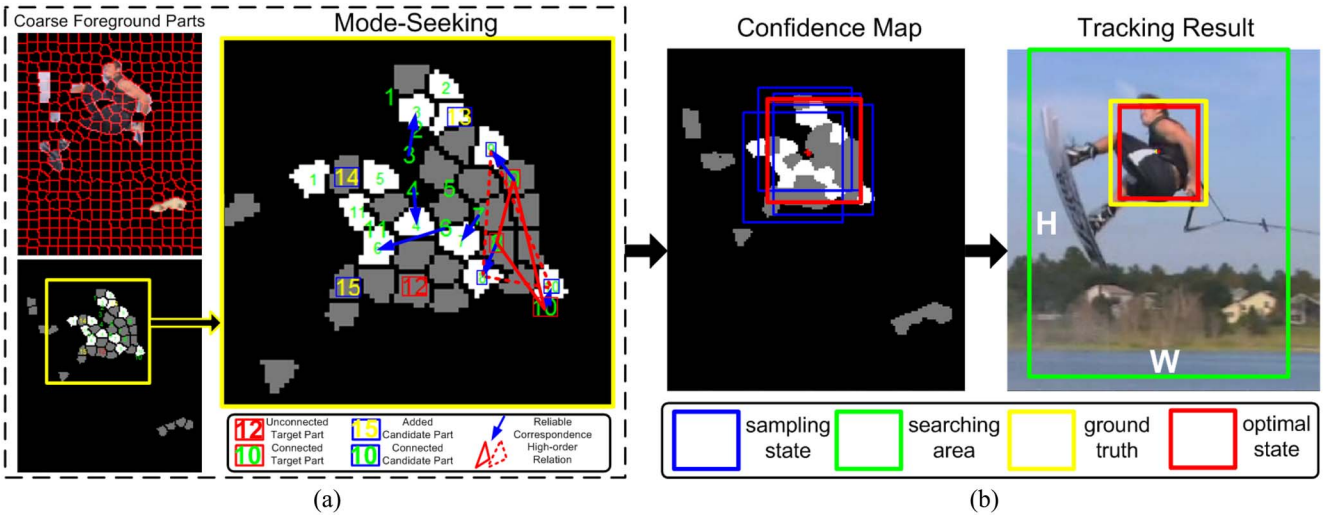

Fig. 2. Tracking process on the sequence *Waterski*. (a) Given the candidate part set Q, we aim to find their reliable correspondence with the target part set P. This is done by seeking modes with tolerance of deformation and scale change. For example, the blue arrows in the figure indicate the displacement between parts in P and Q. Thus, reliable parts can be determined (e.g., parts 8–10). (b) Confidence map is constructed based on the reliable part set, and the tracking result is output by uniform sampling on the confidence map.

By measuring the overall confidence of modes, the tracking problem is formulated as

$$\mathcal{D}^* = \underset{\mathcal{D}}{\operatorname{argmax}}\, \Omega(\mathcal{D}) \quad \text{s.t.} \quad \mathcal{D} \subset \mathcal{G}, |\mathcal{D}| = \kappa \tag{2}$$

where $\mathcal{D}$ is the mode including κ number of vertices. $\Omega(\mathcal{D})$ is the confidence function reflecting the confidence distribution in mode $\mathcal{D}$, which is described as follows.

C. Confidence Measure

We design two terms to encode both the association confidence of vertices and the geometric confidence among them, that is

$$\Omega(\mathcal{D}) = \omega_1 \cdot \underbrace{\sum_{v \in \mathcal{N}(\mathcal{D})} \Gamma(v)}_{\text{Association Confidence}} + \omega_2 \cdot \underbrace{\sum_{\psi \in \mathcal{E}(\mathcal{D})} \Xi(\psi)}_{\text{Geometric Confidence}} \tag{3}$$

where $\mathcal{N}(\mathcal{D})$ and $\mathcal{E}(\mathcal{D})$ denote the vertex set and the hyperedge set of mode $\mathcal{D}$, respectively. ω_1 and ω_2 are the balancing factors of the two terms.

D. Association Confidence

$\Gamma(v)$ encodes the probability of two parts in vertex v belonging to the same class. That is to say, this term assigns different weights to vertices in the hypergraph. We have

$$\Gamma(v) = \exp\left[-\frac{1}{\sigma_v^2} d_\chi(p, q)\right] \tag{4}$$

where $d_\chi(p, q)$ is the $\chi 2$ distance between the appearance feature of two parts, i.e., $v = \{p, q\}$, $p \in P$ and $q \in Q$. In the experiment, the appearance feature is concatenated by HSV histogram and LBP texture feature. σ_v is the scaling parameter of appearance similarity.

E. Geometric Confidence

$\Xi(\psi)$ describes the geometric relation among correspondence hypotheses in hyperedge ψ. We consider this function for different order of the hypergraph.

1) Disconnected Graph: If $k = 1$, the hypergraph becomes a graph with only loops, i.e., $\Xi(\psi) = \emptyset$. Thus, visual tracking only depends on the association confidence $\Gamma(v)$ without any geometric structural constraints. Similar to SPT [36], it is actually a part-based template matching method. The appearance information encoded in $\Gamma(v)$ is usually weak especially for small superpixels, resulting in worse performance in the scenarios with complex appearance variation.

2) Graph: If $k = 2$, the geometric information encoded in $\Xi(\psi)$ provides complementary pairwise geometric information of edge $\psi = \{v_1, v_2\}$ besides appearance than SPT [36] does. Thus, DGT [6] falls into this category. The pairwise similarity to compare two correspondence hypotheses is calculated as

$$\Xi(\psi) = \exp\left[-\frac{1}{\sigma_\psi^2} ||\vec{L}(p_1, p_2) - \vec{L}(q_1, q_2)||_2\right] \tag{5}$$

where p_1 and p_2 denote the parts in target part set P, q_1 and q_2 the parts in candidate part set Q. $\vec{L}(\cdot, \cdot)$ measures the consistency of the two *supporters*, which is calculated as the location displacement of two neighboring correspondence hypotheses, as shown in Fig. 3(a). σ_ψ is the scaling parameter of geometric constraint.

3) Hypergraph: In Fig. 3(a), the supporters provide pairwise relation measurement that are restricted to distances. Since the pairwise distances between far parts are expected to vary more drastically than the distances between close parts, such descriptors have low discriminating power, and many different correspondences have (e.g., the supporters are not matched with the dash blue ones in the target part set). It is

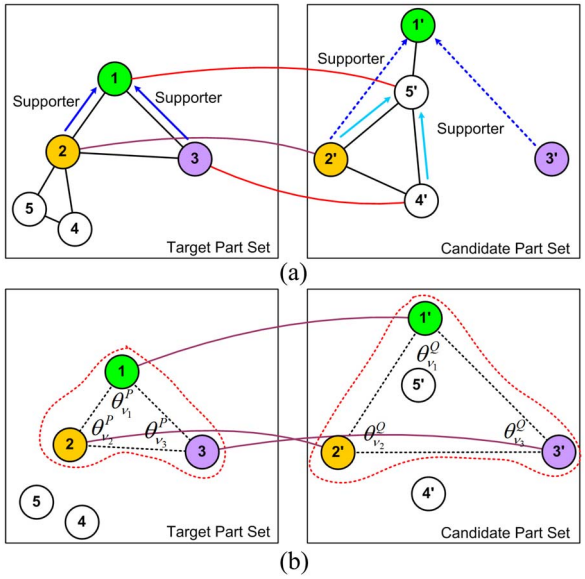

Fig. 3. (a) Pairwise geometric relations. When large scale changes occur, false correspondences (e.g., $1 \sim 5'$ and $3 \sim 4'$) are easily conducted since the supporters (shown in blue arrow) are no longer reliable. (b) High-order geometric relations. Different from the pairwise measure, the angles of the triplet hypotheses are invariant to large scale changes, e.g., $\angle\theta^P_{\nu_1} \sim \angle\theta^Q_{\nu_1}$, $\angle\theta^P_{\nu_2} \sim \angle\theta^Q_{\nu_2}$, and $\angle\theta^P_{\nu_3} \sim \angle\theta^Q_{\nu_3}$.

hard to handle large scale change and conducting false correspondences. We argue that the high-order representation makes it possible to construct more expressive model.

According to [23], the angles of a polygon are invariant under some rotation, translation and scale change. Here, we exemplify the high-order geometric relation with the $k = 3$ case. For example, in Fig. 3(b), three correspondence hypotheses form two triangles ($\triangle_{1,2,3}$ and $\triangle_{1',2',3'}$). We can describe each triangle by its three angles, leading to more reliable correspondences. Although the target scale changes drastically, the angles (high-order geometric relations) remain more stable compared to the relative displacement between parts (pairwise geometric relations). We also present some visual examples in Fig. 2(a), i.e., $\triangle_{8,9,10}$ and $\triangle_{8',9',10'}$. We use the sines of the angles to calculate the geometric confidence, as

$$\Xi(\psi) = \exp\left[-\frac{1}{\sigma_\psi^2}\sum_{i=1}^{k} |\sin(\theta^P_{\nu_i}) - \sin\left(\theta^Q_{\nu_i}\right)|\right] \tag{6}$$

where $\theta^P_{\nu_i}$ and $\theta^Q_{\nu_i}$ denote angles of parts related to vertex ν_i in the set P and Q, respectively. k ($k \geq 3$) is the order of the hypergraph, e.g., the 3-order relation denotes a triangle, and the 4-order relation denotes a quadrilateral, and so on.

IV. Optimization

Given the geometric hypergraph $\mathcal{G}$, the mode-seeking problem is solved by searching $\mathcal{G}$ (Section IV-B). Before that, We propose the confidence-aware sampling technique to improve the efficiency of the proposed method (Section IV-A).

A. Confidence-Aware Sampling

Suppose that the target part set and candidate part set consist of n parts, there are at most n^2 correspondence hypotheses. For the $k = 3$ case, the size of the resulting full-affinity hyperedges will be $\binom{n^2}{3}$, of order $O(n^6)$, which has a high memory complexity. It becomes then necessary to further reduce the computational complexity by introducing a sparse hypergraph structure with significant hypotheses. To this end, we develop a confidence-aware sampling method as follows.

1) First, we reduce the number of vertices deterministically by introducing two thresholds. We assume target parts move smoothly in consecutive frames, which means that the appearances change little in a very short time interval. To remove noises, for each target part $p \in P$, we only consider a few correspondence hypotheses with at most ς number of highest association confidence larger than the appearance threshold ϵ_a.
2) Second, the number of hyperedges is greatly decreased probabilistically. Based on a simple assumption that a vertex with higher association confidence has a higher possibility of being reliable correspondence, we sample more hyperedges around the vertex with higher association confidence. Specifically, starting from each vertex ν in the reduced vertex set, we sample $\eta = [\hat{\Gamma}(\nu) \cdot N_\nu]$ number of hyperedges comprising k vertices without conflicts. We regard the normalized confidence $\hat{\Gamma}(\nu)$ as the sampling probability, and the constant N_ν as the maximal number of sampled hyperedges for each vertex.

Different from other MRF or graph-based approaches considering pairwise relations between the nearest neighboring vertices, we sample hyperedges randomly without distance constraints to exploit the geometric information fully, so that the hypergraph is spanned globally over all correspondence hypotheses. The additional benefit is that we can consider context information between target parts and background parts for more robustness.

Based on the confidence-aware sampling method, we sample vertices and hyperedges of $\mathcal{G}$, obtaining a approximate geometric hypergraph $\mathcal{G}^*$, as shown in Fig. 1(b). Then we directly perform mode-seeking on $\mathcal{G}^*$ instead of $\mathcal{G}$. Specifically, the reduced vertex set $\mathcal{V}^*$ and hyperedge set Ψ^* of $\mathcal{G}^*$ are given as

$$\begin{cases} \mathcal{V}^* = \{\nu | \forall \nu \in \mathcal{V} : \Gamma(\nu) \geq \epsilon_a, |\mathcal{V}^p| \leq \varsigma\}, \\ \Psi^* = \{\psi | \forall \nu \in \mathcal{V}^*, \nu_i, \nu_j \in \psi : |\Psi^\nu| \leq \eta, \nu_i \cap \nu_j = \emptyset\}. \end{cases} \tag{7}$$

In (7), $|\mathcal{V}^p|$ denotes the number of vertices including part p, and $|\Psi^\nu|$ denotes the number of hyperedges including vertex ν. The sampling scheme ensures finding enough relevant correspondence hypotheses. Moreover, it decreases the number of vertices from n^2 to at most $n\varsigma$ and the number of hyperedges from $\binom{n^2}{3}$ to at most $n\varsigma\eta$, which removes the great majority of redundant vertices and hyperedges in $\mathcal{G}$ empirically.

B. Mode-Seeking Problem

Since the maxima of (2) indicates a structural correspondence mode, we fully search the hypergraph $\mathcal{G}$ by setting each vertex $\nu^\star$ in the hypergraph as a starting point. Let $\mathcal{D}_{\nu^\star}$ be the mode with vertex set $\mathcal{N}(\mathcal{D}_{\nu^\star})$ and hyperedge set $\mathcal{E}(\mathcal{D}_{\nu^\star})$. Let

$\mathcal{P} \in \mathbf{R}^N$ be the vector containing the probability of each vertex in the hypergraph belonging to mode $\mathcal{D}_{v^\star}$, i.e., if $\mathcal{P}_v > 0$, $v \in \mathcal{N}(\mathcal{D}_{v^\star})$; otherwise, $v \notin \mathcal{N}(\mathcal{D}_{v^\star})$. N is the number of vertices. Combined with (3), the problem in (2) is cast as optimizing $\mathcal{P}$ and further rewritten as

$$\mathcal{P}^* = \underset{\mathcal{P}_v : v \in \mathcal{N}(\mathcal{D}_{v^\star})}{\operatorname{argmax}} \left(\sum_{v \in \mathcal{N}(\mathcal{D}_{v^\star})} \Gamma(v)\mathcal{P}_v + \sum_{e \in \mathcal{E}(\mathcal{D}_{v^\star})} \Xi(e) \prod_{v \in e} \mathcal{P}_v \right)$$
$$\text{s.t.} \quad \sum_{v \in \mathcal{V}} \mathcal{P}_v = 1, \mathcal{P}_v \in \{0, \mu\}, \frac{1}{\mu} \geq k + 1 \tag{8}$$

In (8), the first term in the objective function penalizes the inclusion of vertices corresponding to less association confidence indicated by a lower $\Gamma(v)$, and the second term encourages the inclusion of hyperedges in the mode with larger geometric confidence $\Xi(e)$. Essentially, this is a NP-hard combinatorial optimization problem. To solve this problem, the constraint $\mathcal{P}_v \in \{0, \mu\}$ is relaxed to $\mathcal{P}_v \in [0, \mu]$, where $\mu \leq 1$ is a constant. Let the number of vertices in $\mathcal{D}_{v^\star}$ be m, the mode contains at least $m = (1/\mu)$ number of vertices when keeping the constraint $\sum_{v \in \mathcal{V}} \mathcal{P}_v = 1$. To avoid the degeneracy problem, we require the minimal vertices in a mode satisfying the constraint $(1/\mu) \geq k + 1$ to guarantee adequate structural correspondences included in one mode.

We briefly describe the details of the optimization method as follows. To initialize *starting probability* $\mathcal{P}$ at vertex v^*, we first sort the hyperedges connected to v^* in descending order according to their confidences. Then, we add μ number of vertices associated with the hyperedge involved v^* to a *starting mode* $\mathcal{D}$ by going through these hyperedges. Finally, we obtain $\mathcal{P}$ by setting the corresponding component $\mathcal{P}_v = 1/\mu$ in $\mathcal{P}$ for each vertex $v \in \mathcal{D}$. For each starting probability $\mathcal{P}$, the pairwise coordinate update method [26] is used to solve the problem in (8) and determine $\mathcal{D}_{v^\star}$ effectively, as shown in Fig. 1(c).

V. Tracking

A. Extracting Reliable Parts

Given $\mathcal{P}^*$, we obtain the vertices belonging to the corresponding mode $\mathcal{D}$, i.e., $\mathcal{D} = \{v | v \in \mathcal{V} : \mathcal{P}_v > 0\}$. All unique exploited modes are put in a mode set $\mathbf{D}^*$. For every vertex belonging to $\mathbf{D}^*$, we find a reliable correspondence between parts in the target part set P and the candidate part set Q. Finally, we determine the reliable part set $\mathcal{S}$ from Q in the current frame. The whole procedure is summarized in Fig. 1(d) and Algorithm 1.

Algorithm 1 Extracting Reliable Parts

Input: mode set $\mathbf{D}$
Output: reliable target set $\mathcal{S}$

```
Sort the mode set D = {D_1, ..., D_N} based on the confidence values {Ω(D_1), ..., Ω(D_N)} in descending order
Initialize the mode set without conflicts D* = ∅
for each non-empty mode D_i ∈ D, D_i ≠ ∅ do
    if no intersections with all members in the set, i.e., ∀j, D*_{v_j} ∈ D* : D_i ∩ D*_j = ∅ then
        Add to the mode set, i.e., D* ← D* ∪ {D_i}
    else
        Remove the overlapping part in the parsed modes, i.e., ∀j, D*_j ∈ D* : D̂_i ← D_i / D*_j
        Add to the mode set, i.e., D* ← D* ∪ {D̂_i}
    end if
end for
Obtain the reliable target set, i.e., S = {p | ∀v ∈ D*_i, D*_i ∈ D* : p ∈ v}
```

B. Reliable Parts-Based Voting

After obtaining $\mathcal{S}$, we determine the target state in the current frame t, including center ℓ_*^t and scale s_*^t of the target by reliable parts-based voting. Similar to the method of [6], we construct a confidence map C to represent location probability of the target in the searching area as

$$C(i,j) = \begin{cases} \lambda_1 & (i,j) \in \mathcal{R}_{\mathcal{D}^*} \\ \lambda_2 & (i,j) \in \mathcal{R}_{\mathcal{V}^*}, (i,j) \notin \mathcal{R}_{\mathcal{D}^*} \\ \lambda_3 & \text{otherwise} \end{cases} \tag{9}$$

where (i, j) is the position in the searching area. $\mathcal{R}_{\mathcal{D}^*}$ means the region of target parts belonging to the extracted modes, and $\mathcal{R}_{\mathcal{V}^*}$ means the region of candidate parts. $\{\lambda_1, \lambda_2, \lambda_3\}$ are weights reflecting the influence of each type of regions. Further, λ_1 controls the influence of mode-seeking, which is referred as the weight of mode-seeking.

To find the bounding box to cover more foreground regions with respect to center ℓ and scale s, we form the following optimization problem:

$$\{\ell_*^t, s_*^t\} = \underset{\ell, s}{\operatorname{argmax}} \sum_{(i,j) \in \mathcal{R}(\ell, s)} C(i,j) \tag{10}$$

where $\mathcal{R}(\ell, s)$ means the region with center ℓ and scale s.

The target center in the current frame t is largely determined by the one in the previous frame $t-1$ with the geometric constraint. To reduce computational complexity, we first estimate a rough target center by calculating the weighted mean of the target part center l_p^t with weight w_p^t, that is

$$\ell^t = \sum_{p \in \mathcal{S}} \left(\ell_*^{t-1} + l_p^t - l_p^{t-1} \right) \cdot \frac{w_p^t}{\sum_{p \in \mathcal{S}} w_p^t} \tag{11}$$

where ℓ_*^{t-1} is the optimal center in the previous frame $t-1$. w_p^t denotes the confidence of the mode, including reliable part p in the current frame t, i.e., $w_p^t = \Omega(\mathcal{D}), p \in v, v \in \mathcal{N}(\mathcal{D})$. After that, we modify the target center with the displacement perturbation term δ_ℓ^t and adjust the target scale with the scale perturbation term δ_s^t for a visually better location. The maximal values of two perturbation terms $\{\delta_\ell^t, \delta_s^t\}$ are set as the mean diameter of candidate parts in the current frame t. The final target state $\{\ell_*^t, s_*^t\}$ is obtained by optimizing (10) using

a sampling strategy, namely selecting the one with the maximal score out of numerous randomly sampled states $\{\ell + \delta_\ell^t, s + \delta_s^t\}$. Assembling all parts belonging to the target, we find the optimal target state, as shown in Fig. 2(b).

C. Online Updating of Hypergraph

To handle possible significant changes of target appearance, geometric hypergraph $\mathcal{G}$ is updated in two aspects, i.e., target part set P and candidate part set Q. As illustrated in Fig. 2(a), based on the parsed reliable part set $\mathcal{S}$, an old part in P (e.g., part 12) is deleted if it does not involve in any structural correspondence for a fixed number of frames (five frames in the experiment), while a new part (e.g., parts 14 and 15) not involved in existed modes is added in P such that its geometric distance to any other parts is larger than a threshold[3] to preserve the spatial sparsity of P. On the other hand, the appearance model in the MRF-based segmentation method is updated to generate Q every frame, as similar as in [6].

VI. Experiments

A. Datasets and Protocols

1) VOT2014 Dataset: The VOT2014 dataset [13] is popularly used in the tracking community, which is collected with representative 25 sequences selected from 394 sequences. Each sequence is annotated by several attributes, such as occlusion and illumination changes.

We evaluate the tracking methods following two protocols of the VOT2014 challenges, i.e., *Baseline* and *Region_noise*. *Baseline* corresponds to the experimental setting, where the tracker is run on each sequence 15 times by initializing it on the groundtruth bounding box, obtaining average statistic scores of the measures. *Region_noise* corresponds to the experiment setting, where the tracker is initialized with 15 noisy bounding boxes, which are randomly perturbed in order of 10% of the groundtruth bounding box size, in each sequence. As defined in [8], two performance metrics, *accuracy* (average bounding box overlap between the bounding box predicted by the tracker and the groundtruth one) and *robustness* (number of reinitializations once the overlap ratio measure drops to zero) are reported in the experiment.

2) OTB100 Dataset: For more comprehensive evaluation, we also compare the tracking methods on the popular OTB100 dataset [42], which contains 100 image sequences and involves 11 tracking attributes (e.g., illumination changes, scale variation, partial or full occlusion, and rotation).

The compared trackers are run throughout the sequence with initialization from the ground truth position in the first frame, which is referred as one-pass evaluation (OPE) in [43]. In addition, we use the *success plot* for evaluation. The plot draws the percentage of successfully tracked frames versus the bounding box overlap threshold, where area under the curve is used as *success score* for ranking.

3) Deform-SOT Dataset: To further evaluate the performance of trackers on deformation and occlusion, we evaluate our method on the Deform-SOT dataset [11], which includes 50 challenging sequences and different targets with deformation and occlusion in varying levels in unconstrained environments. The dataset is diverse in object categories, camera viewpoints, sequence lengths, and challenging levels in six aspects, including large deformation, severe occlusion, abnormal movement, illumination variation, scale change, and background clutter, for comparison.

[3]The threshold is set as double mean diameter of candidate parts in the current frame.

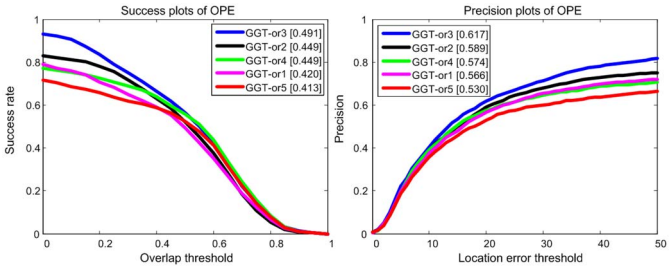

Fig. 4. Performance versus order of hypergraph.

To evaluate a tracker's performance with different initialization at a different start frame, we employ additional temporal robustness evaluation (TRE) and spatial robustness evaluation (SRE) by perturbing the initialization temporally and spatially according to [43]. We also use the *precision plot* for evaluation except the *success plot* measure. The precision plot shows the percentage of successfully tracked frames versus the center location error in pixels, which ranks the trackers as *precision score* at 20 pixels.

B. Parameter Analysis

The proposed tracker is implemented with MATLAB and C and runs at 0.5 frames/s on a machine with a 2.9 GHz Intel i7 processor and 16GB memory. We first study the influence of several important parameters or modules as follows, where the experiments are performed on 15 sequences from the Deform-SOT dataset.

1) Order of Hypergraph: The order of hypergraph decides the number of correspondence hypotheses we consider for the geometric relations simultaneously. We conduct baseline methods with different orders, namely, GGT-or$k$ ($k = 1, 2, 3, 4, 5$). From Fig. 4, GGT-or3 considering high-order geometric relations performs the best. It indicates the effectiveness of the high-order representation. In contrast, GGT-or2 and GGT-or1 consider just pairwise or no geometric relations, leading to big accuracy loss. We note that higher-order methods ($k > 3$) perform not well also. This is because a large amount of correspondences are taken into consideration with more relations with higher order hypergraphs, so that both false and reliable correspondences are removed.

2) Size of Superpixel: The size of superpixel means the number of pixels in it, and affects the number of vertices in the hypergraph. As Fig. 5 shows, we consider different size of superpixel ϱ, i.e., GGT-spϱ ($\varrho = 30, 50, 80, 100, 150, 200$). If the superpixel is too large (e.g., GGT-sp200), it is hard to exploit discriminative geometric structure cues of local

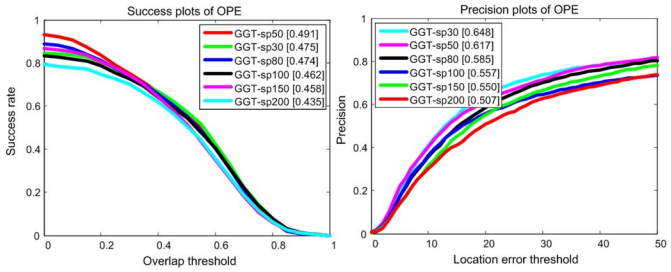

Fig. 5. Performance versus size of superpixel.

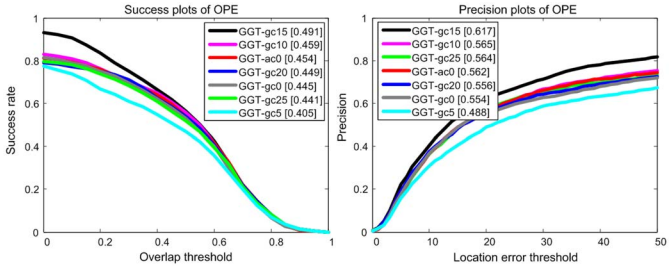

Fig. 6. Performance versus weights in confidence measure.

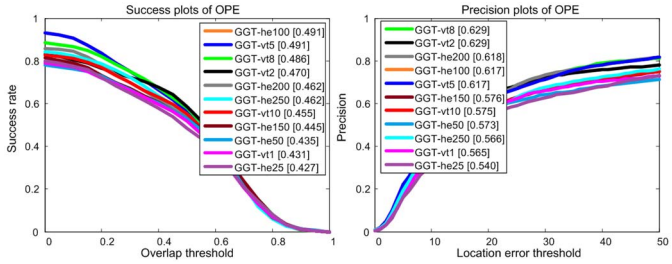

Fig. 7. Performance versus sampling number.

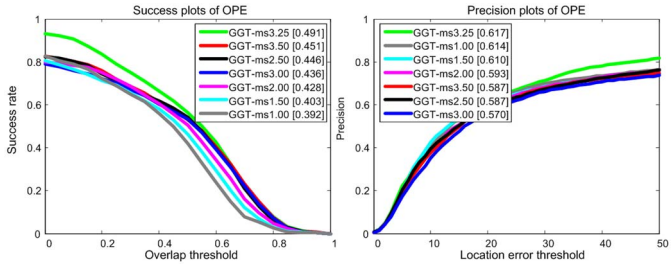

Fig. 8. Performance versus weight of mode-seeking.

parts to handle deformation. If it is too small (e.g., GGT-sp30), the large number of superpixels increases the computational complexity considerably without apparent performance improvement.

3) Weights in Confidence Measure: The weight of geometric confidence indicates the importance of geometric confidence. We set $\omega_1 = 10$ and enumerate the weight ω_2 in (3), i.e., GGT-gcω_2 ($\omega_2 = 0, 5, 10, 15, 20, 25$). Based on the performance in Fig. 6, an appropriate factor helps the tracker achieve higher performance by neither underestimating nor overestimating the geometric information. Besides, we verify the effectiveness of the association confidence in (3) by setting $\omega_1 = 0$, namely, GGT-ac0. It can be seen that the association confidence term brings some performance improvement.

4) Sampling Number: In Fig. 7, we show the effect of the confidence-aware sampling module by evaluating different number of sampled vertices and hyperedges [i.e., GGT-vtς ($\varsigma = 1, 2, 5, 8, 10$ with fixed $N_v = 100$) and GGT-heN_v ($N_v = 25, 50, 100, 150, 200, 250$ with fixed $\varsigma = 5$)]. If the number of sampled vertices and hyperedges is too small (e.g., GGT-vt1 and GGT-he25), the approximate hypergraph may drop too much useful information from the original hypergraph. If the number is too large (e.g., GGT-vt10 and GGT-he250), many noisy relations will be introduced, because reliable correspondences are much sparser than false ones.

5) Weight of Mode-Seeking: The weight of mode-seeking measures the importance of mode-seeking. In Fig. 8, we report the performance versus weight of mode-seeking with fixed other weights $\{\lambda_2, \lambda_3\} = \{1, -1\}$ in (9), denoted as GGT-msλ_1 ($\lambda_1 = 1.00, 1.50, 2.00, 2.50, 3.00, 3.25, 3.50$). When $\lambda_1 = 1.00$, we perform tracking by the superpixel initiation without mode-seeking. In this case, all the superpixels have the same contribution to the target position estimation. It is liable to cause target drifting in complex scenes. When we take appropriate value ($\lambda_1 = 3.25$), reliable parts can be assigned to corresponding weights in the target position estimation by the mode-seeking, leading to more robustness.

Based on the above parameter analysis, we set and fix all parameters in our algorithm empirically. The order of the geometric hypergraph is set as $k = 3$. For the searching area, we search the target location in current frame by three times the size of previous one. For the SLIC over-segmentation method, the number of pixels in each superpixel is set as $\kappa = 50$, and the range of number of superpixels $[100, 450]$. We use 8 bins for each channel of HSV feature to represent the appearance of target parts. The weights in (3) are set as $\{\omega_1, \omega_2\} = \{10, 15\}$. The scaling parameters $\sigma_v^2 = 1.0$ in (4), and $\sigma_\psi^2 = 1.0$ in (5) and (6). In the sampling method, the appearance threshold is set as $\epsilon_a = 0.3$, the maximal number of sampled vertices is set as $\varsigma = 5$, and the maximal number of sampled hyperedges is set as $N_v = 100$. In (9), the weight $\{\lambda_1, \lambda_2, \lambda_3\} = \{3.25, 1, -1\}$.

C. Evaluations on the VOT2014 Dataset

We compare our approach to several algorithms, including the winner of the VOT2014 challenge, discriminative scale space tracker (DSST) [9], and two of the top-performing trackers of the online tracking benchmark [43], namely, Struck [16] and kernelized correlation filter tracker (KCF) [18]. Furthermore, we include keypoint-based consensus-based matching and tracking [31] and initialization insensitive visual tracker [47], the part-based DGT [6], local-global tracker (LGT) [7], online graph-based tracker (OGT) [29], and pixel tracker [12], as well as the baseline trackers, including Frag [2], compressive tracker (CT) [50], and multiple instance learning tracker (MIL) [3]. To ensure a fair comparison, all the results are taken from the original submissions to the VOT2014 challenge by the corresponding authors or the VOT committee.

Examples of visual tracking results of top five trackers are shown in Fig. 9. As the results show, our method outperforms

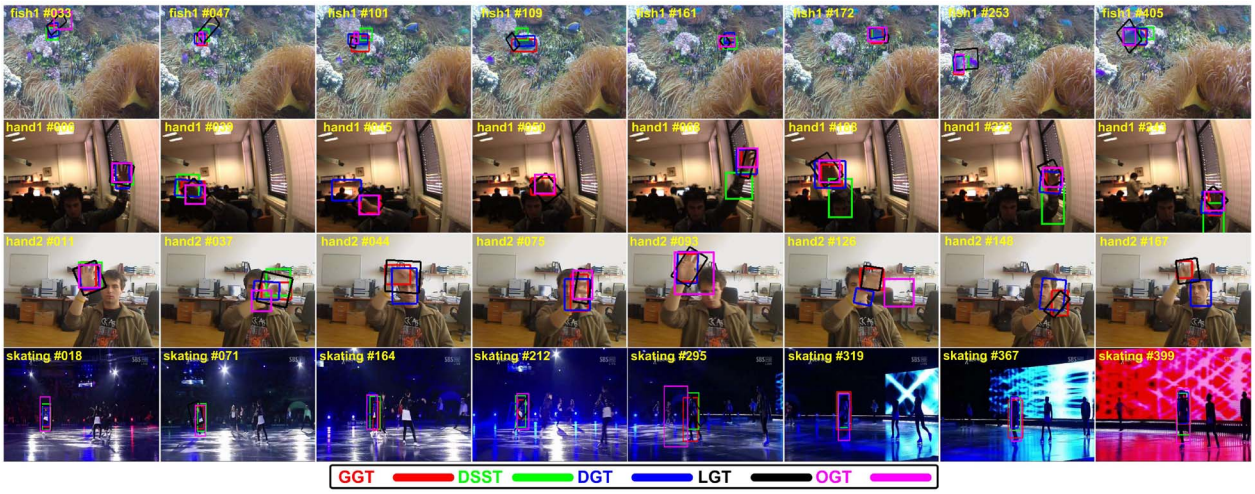

Fig. 9. Tracking results of five trackers (i.e., GGT, DSST [9], DGT [6], LGT [7], and OGT [29]), are denoted in different colors on the VOT2014 dataset (from top to down are *Fish1*, *Hand1*, *Hand2*, and *Skating*, respectively). Note that one tracker is not shown in some frames, which means it fails in tracking and will reinitialize later (e.g., DSST [9] fails in *Hand1* #050).

TABLE I
TRACKING RESULTS ON THE VOT2014 DATASET. ACCURACY SCORES AND RANKS (ACC. SC. AND ACC. RK. FOR SHORT) ARE REPORTED AS WELL AS THE ROBUSTNESS ONES. THE FIRST, SECOND, AND THIRD BEST VALUES ARE HIGHLIGHTED BY RED, BLUE, AND GREEN COLOR, RESPECTIVELY

	Baseline		*Region_noise*		**Overall**	
	Acc. Sc./Acc. Rk.	**Rob. Sc./Rob. Rk.**	**Acc. Sc./Acc. Rk.**	**Rob. Sc./Rob. Rk.**	**Acc. Sc./Acc. Rk.**	**Rob. Sc./Rob. Rk.**
GGT	0.58/6.16	0.55/4.98	0.57/4.81	0.65/4.93	0.57/5.48	0.59/4.95
DSST [9]	0.62/4.48	1.16/6.32	0.58/4.01	1.28/6.22	0.60/4.25	1.22/6.27
DGT [6]	0.58/5.81	1.00/5.02	0.58/4.97	1.17/5.31	0.58/5.39	1.09/5.16
KCF [18]	0.63/4.22	1.32/6.53	0.58/4.50	1.52/6.62	0.61/4.36	1.42/6.57
LGT [7]	0.47/9.29	0.66/5.96	0.46/8.73	0.64/5.42	0.47/9.01	0.65/5.69
Struck [16]	0.52/8.04	2.16/8.64	0.49/7.90	2.22/8.16	0.51/7.97	2.19/8.40
OGT [29]	0.55/7.09	3.34/9.78	0.51/7.19	3.37/10.30	0.53/7.14	3.36/10.04
PTp [12]	0.47/10.98	1.40/7.20	0.45/9.77	1.46/7.33	0.46/10.38	1.43/7.26
CMT [31]	0.48/9.18	2.64/9.16	0.44/9.97	2.64/9.14	0.46/9.58	2.64/9.15
FoT [41]	0.51/8.44	2.28/9.69	0.48/9.13	2.71/10.59	0.50/8.79	2.50/10.14
IIVT [47]	0.47/9.30	3.19/9.70	0.45/9.96	3.13/9.14	0.46/9.63	3.16/9.42
FSDT [13]	0.47/9.87	3.08/11.26	0.46/9.36	2.77/10.38	0.47/9.62	2.93/10.82
IVT [25]	0.47/9.87	2.76/10.44	0.44/10.69	2.86/10.20	0.46/10.28	2.81/10.32
CT [50]	0.43/11.76	3.12/10.23	0.43/11.04	3.34/10.45	0.43/11.40	3.23/10.34
Frag [2]	0.48/9.17	3.32/12.20	0.44/10.20	3.46/12.29	0.46/9.69	3.39/12.24
MIL [3]	0.40/12.03	2.27/8.80	0.35/13.67	2.60/9.67	0.38/12.85	2.44/9.23

other state-of-the-art trackers, such as DGT [6], LGT [7], and OGT [29]. When the figure skater in *Skating* moves under the challenges of background clutter and illumination variation, some trackers do not locate well (e.g., OGT [29] in #295 and DGT [6] in #212). Besides, DSST [9] fails in tracking the *Hand* in *Hand1* #050 and *Hand2* #167. This may attribute to the high-order geometrical correlation captured in our method. Such correlations reflect properties that are invariant to local transforms (i.e., angles in the polygon of the structures that are invariant to rotations, scale changes). Such invariance makes our algorithm more robust to large changes in geometric shape or appearance.

Table I shows the average performance of the compared trackers. As these results show, our algorithm achieves the overall best robustness score, and comparable performance in accuracy among all the methods compared. Moreover, the considerable improvement in *Region_noise* level indicates that the spatial high-order representation in our method can resist noises effectively, and recover from initialization errors to gain improvements both in terms of accuracy and robustness.

D. Evaluations on the OTB100 Dataset

On the large OTB100 dataset, we evaluate our algorithm against several existing methods, such as TLD [21], DGT [6], TCP [24], STT [39], IVT [25], CT [50], Struck [16], Frag [2], SCM [52], locally orderless tracker (LOT) [32], ASLA [20], DSST [9], and KCF [18]. All the results of other trackers are taken from the OTB website or reproduced from the available source codes.

As shown in Fig. 10(a), our proposed method achieves the best performance in all the ranking plots. Then in Fig. 10(b)–(l), we use the success plot to quantitatively evaluate the performance of each tracker with each tracking attribute. The results show that our GGT tracker achieves favorable performance than the other compared trackers, e.g., DSST, KCF, STT, and DGT. We explain the tracking performance in two aspects.

1) The part-based GGT algorithm employs the high-order representation to introduce more flexibility for the target with deformation and occlusion. Therefore, our method performs well for the videos with most attributes, such

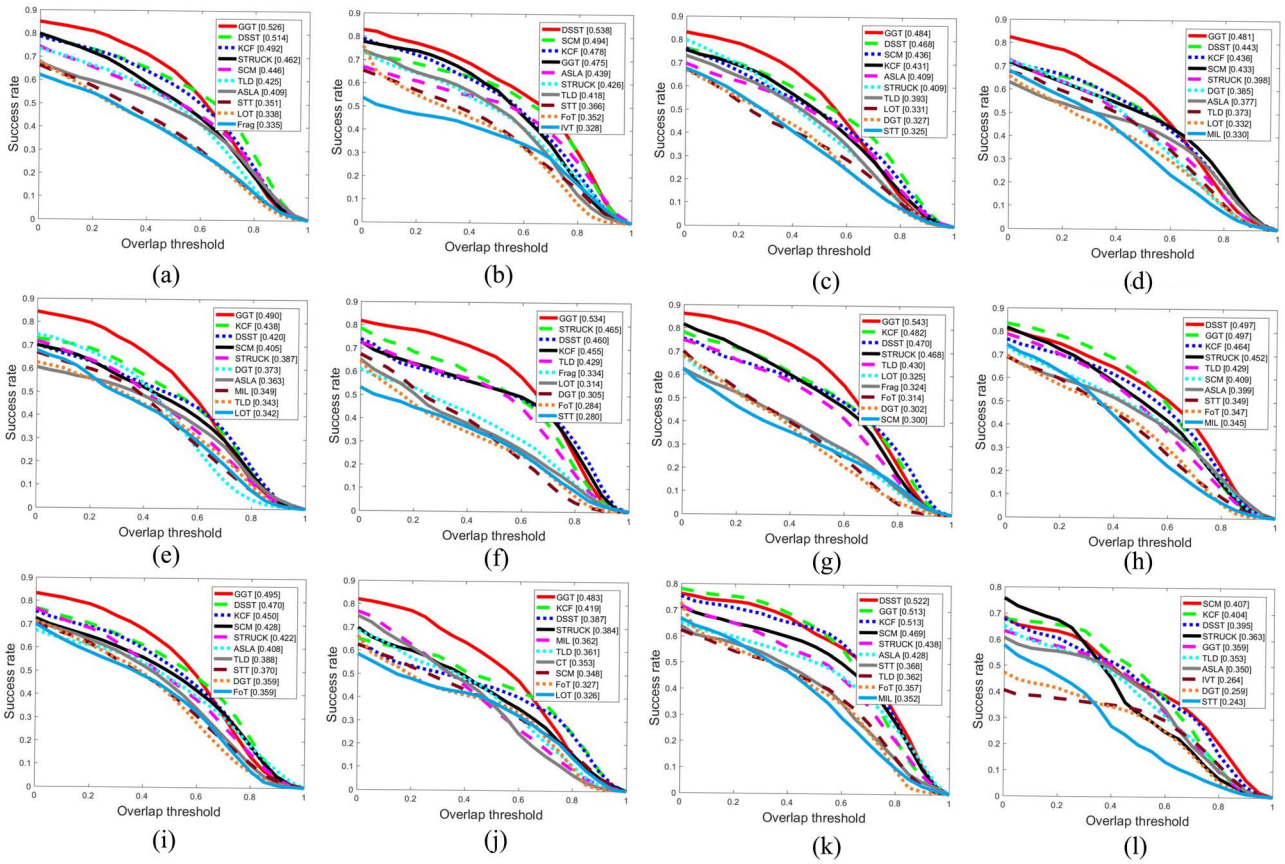

Fig. 10. Success plots of OPE with different attributes on the OTB100 dataset. (a) Success plots of OPE. (b) Success plots of OPE—illumination variation (37). (c) Success plots of OPE—scale variation (63). (d) Success plots of OPE—occlusion (48). (e) Success plots of OPE-deformation (43). (f) Success plots of OPE—motion blur (29). (g) Success plots of OPE-fast motion (39). (h) Success plots of OPE-in-plane rotation (51). (i) Success plots of OPE-out-of-plane rotation (63). (j) Success plots of OPE—out of view (14). (k) Success plots of OPE—background clutter (31). (l) Success plots of OPE—low resolution (9).

as scale variation, occlusion, deformation, and fast motion.

2) In Fig. 10(l), our method performs not well in the low resolution attribute because two small parts lose much discriminability. However, bounding box-based trackers (e.g., DSST and Struck) can extract appearance variation for small target by holistic representation effectively.

1) Failure Cases: Fig. 11 depicts three failure cases in the OTB100 dataset, i.e., *Soccer*, *Biker*, and *Kitesurf*. Our method failed partly because it is hard to extract small discriminative parts on the target with already small size (e.g., *Biker*). It is also partly due to heavy *background clutter* or *illumination variation* that confuse coarse foreground parts extraction (e.g., *Soccer* and *Kitesurf*). In addition, our method will lose the target out of the searching area facing very *fast motion* (e.g., *Soccer* and *Biker*). Note that this weakness also affects the other methods.

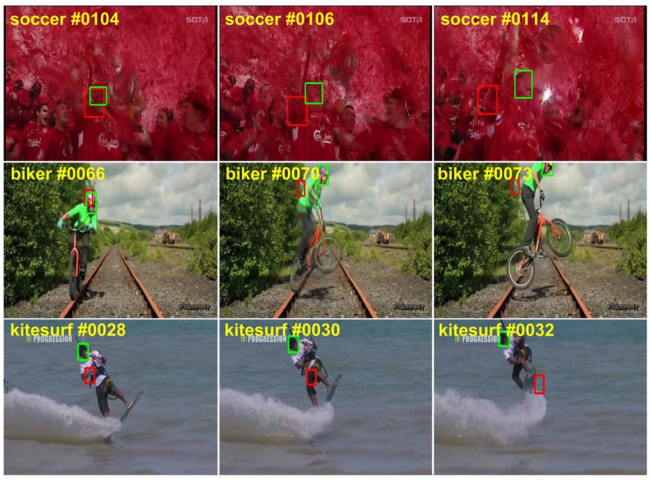

Fig. 11. Failure cases of our tracker in the OTB100 dataset. The red and green bounding box denote our tracking result and groundtruth, respectively.

E. Evaluations on the Deform-SOT Dataset

We evaluate the proposed algorithm against existing methods, including holistic model-based trackers (i.e., DSST [9], KCF [18], IVT [25], L1T [28], TLD [21], MIL [3], Struck [16], MTT [51], CT [50], CN [10], STT [39], and STC [49]) and part-based trackers (i.e., Frag [2], SPT [36], SCM [52], LOT [32], ASLA [20], LSL [46], LGT [7], DGT [6], and TCP [24]). For fair comparison, we use the *same* initial bounding box of each sequence for all trackers. The experimental results of other trackers are reproduced from the available source codes with recommended parameters.

As shown in Fig. 12, the evaluation results on OPE, SRE, and TRE indicate that our GGT tracker performs against other

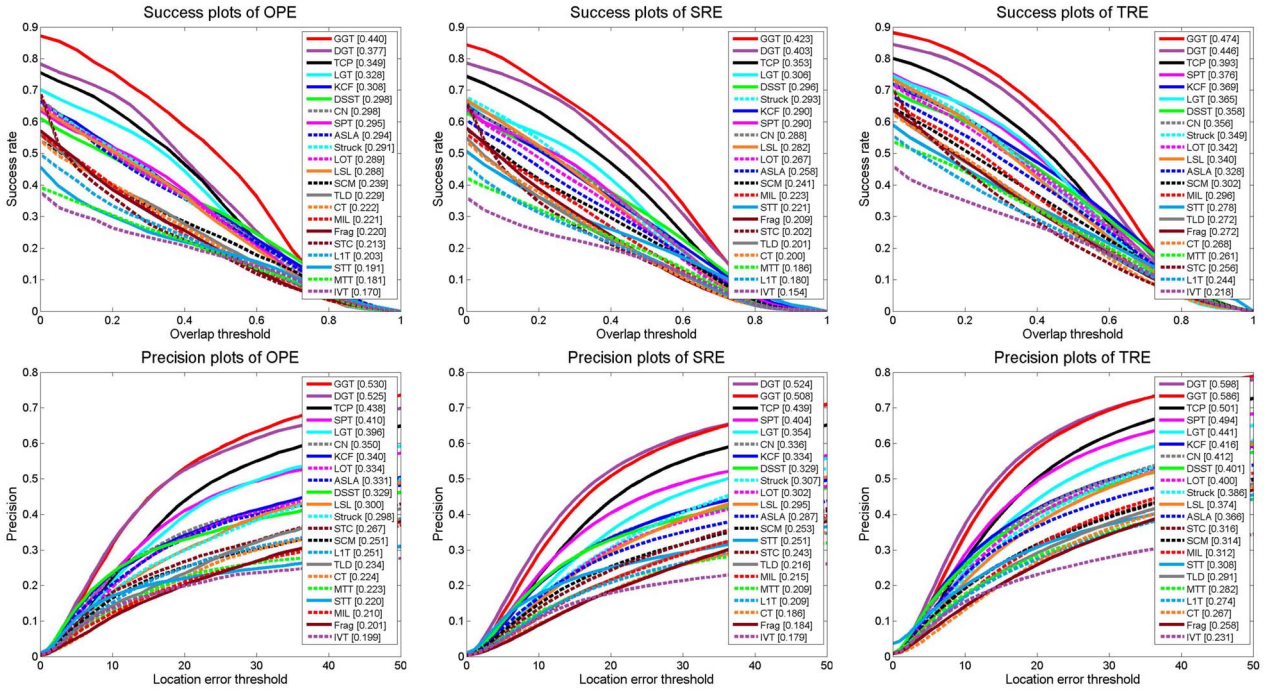

Fig. 12. Precision plot and success plot over the Deform-SOT dataset using OPE, SRE, and TRE.

Fig. 13. Tracking results, denoted in different colors and lines, on the Deform-SOT dataset (from left to right and top to down are *Bike*, *Boarding*, *Bolt*, *Carscale*, *Football*, *Run*, *Uneven-Bars*, and *Waterski*, respectively).

compared methods. In addition, Fig. 13 shows the tracking results of compared trackers on several sequences.

1) Attribute-Based Evaluation: We also compare the performance of all tracking algorithms for videos with varying degrees of six challenging factors shown in Fig. 14.

2) Large Deformation: Existing part-based trackers [6], [7], [36], [46] mainly consider pairwise geometric relations, which are not effective for the sequences with significant target deformation (e.g., *Boarding* in Fig. 13). According to Fig. 14(a) and (g), our tracker performs against other methods because high-order geometric relations instead of varying pairwise displacements preserve invariant angles to remove noises from a large set of correspondence hypotheses.

3) Severe Occlusion: Some trackers [20], [21], [28], [36], [46], [52] drift away from the target or do not scale well when the target is heavily occluded (e.g., *Boarding*, *Carscale*, *Run*, and *Waterski* in Fig. 13). However, our method performs tracking relative accurately because the modes exploit invariant local geometric structure of target parts. This information helps to avoid much influence of occlusion, as long as adequate modes are detected to vote for the target state.

4) Abnormal Movement: Abnormal movements consist of all kinds of nonrigid change, such as abrupt motion, pose variation, and rotation. For example, SCM [52] and TCP [24] drift away when the gymnast jumps to grab bars in *Uneven-Bars* #303. By comparison, our method performs well in estimating both scales and positions on these challenging sequences. It can be attributed to two reasons. First, the hypergraph is constructed with coarse foreground parts without unnecessary background parts [see Fig. 2(a)]. Moreover, reliable parts

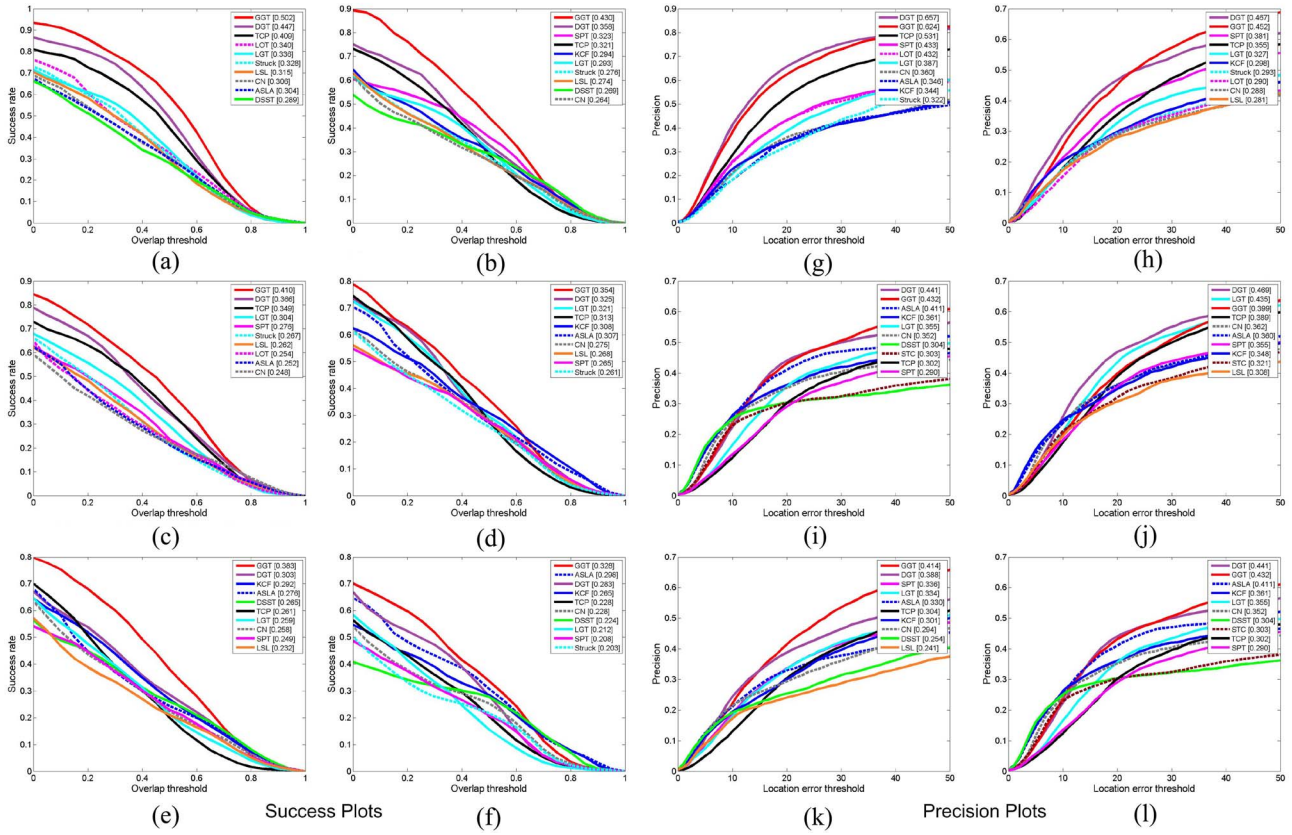

Fig. 14. Plots of OPE with different attributes on the Deform-SOT dataset. (a) Success plots of OPE—large deformation (22). (b) Success plots of OPE—severe occlusion (20). (c) Success plots of OPE—abnormal movement (26). (d) Success plots of OPE—illumination variation (17). (e) Success plots of OPE—scale change (22). (f) Success plots of OPE—background clutter (15). (g) Precision plots of OPE—large deformation (22). (h) Precision plots of OPE—severe occlusion (20). (i) and (l) Precision plots of OPE—background clutter (15). (j) Precision plots of OPE—illumination variation (17). (k) Precision plot of OPE—scale change (22).

corresponding to the optimal target state are determined from noisy parts by a majority vote.

5) Illumination Variation: Some trackers [10], [20] are insensitive to appearance changes caused by illumination variation. However, compared to our method, they perform poorly on the sequences undergoing other challenges such as large deformation and abnormal movement simultaneously (see *Bike* in Fig. 13). This is because the geometric hypergraph adapts to the appearance variations of the local parts.

6) Scale Change: In terms of sequences with significant scale change (e.g., *Boarding* and *Carscale* in Fig. 13), our tracker performs against other methods [6], [20], [24] as in Fig. 14(e) and (k). This is because we employ the angles of the triangle to measure the similarity of several correspondence hypotheses, which is invariant to scale change (see more in Section IV-A). Different from our algorithm, DGT [6] just considers neighboring pairwise relations between local parts, making it less flexible to handle scale changes.

7) Background Clutter: The background surrounding the target has similar appearance, leading to drift from the intended target to other objects when they appear in close proximity (e.g., *Football* #499 in Fig. 13). To handle this problem, some methods [39], [49] exploit the context information around the target, while the other ones [6], [7] employ a graph-based representation to capture geometric structure of the target. Owing to the confidence-aware sampling without distance constraint, sampled representative hyperedges not only consider the relations between target parts and background parts (*context*), but also model the inlier geometric relations among local target parts (*structure*) simultaneously. As a whole, our method ranks the first in success score in Fig. 14(f) and the second in precision score in Fig. 14(i).

VII. Conclusion

In this paper, we describe the GGT for visual tracking, where k-order geometric relations among *correspondence hypotheses* are integrated in the dynamically constructed *geometric hypergraph*. Our method is a general method in that the traditional graph-based tracking methods can be viewed as special cases with lower-order of hypergraph. Moreover, the confidence-aware sampling method is developed to reduce computational complexity and the scale of hypergraph for better efficiency. Experiments are performed on the VOT2014, OTB100, and Deform-SOT dataset to demonstrate the favorable performance of the proposed method compared to other existing methods.

There are several issues in our method that we plan to further improve in future works. To further improve tracking performance, we plan to exploit high-order temporal and

spatial relations among a large number of correspondence hypotheses in multiple consecutive frames simultaneously. Another possible direction is to learn a holistic target representation which is updated jointly with the local superpixel representation for more robustness and discriminability.

REFERENCES

[1] R. Achanta *et al.*, "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] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 1. New York, NY, USA, 2006, pp. 798–805.

[3] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1619–1632, Aug. 2011.

[4] W. Bouachir and G.-A. Bilodeau, "Collaborative part-based tracking using salient local predictors," *Comput. Vis. Image Understand.*, vol. 137, pp. 88–101, Aug. 2015.

[5] Y. Boykov and V. Kolmogorov, " An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1124–1137, Sep. 2004.

[6] 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.

[7] 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.

[8] L. Čehovin, A. Leonardis, and M. Kristan, "Visual object tracking performance measures revisited," *IEEE Trans. Image Process.*, vol. 25, no. 3, pp. 1261–1274, Mar. 2016.

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

[10] M. Danelljan, F. S. Khan, M. Felsberg, and J. Van de Weijer, "Adaptive color attributes for real-time visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Columbus, OH, USA, 2014, pp. 1090–1097.

[11] D. Du *et al.*, "Online deformable object tracking based on structure-aware hyper-graph," *IEEE Trans. Image Process.*, vol. 25, no. 8, pp. 3572–3584, Aug. 2016.

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

[13] M. Kristan *et al.*, "The visual object tracking VOT2014 challenge results," in *Proc. Workshops Conjunction Eur. Conf. Comput. Vis.*, Zürich, Switzerland, 2014, pp. 191–217.

[14] M. Godec, P. M. Roth, and H. Bischof, "Hough-based tracking of non-rigid objects," in *Proc. IEEE Int. Conf. Comput. Vis.*, Barcelona, Spain, 2011, pp. 81–88.

[15] Y. Guo, "Object tracking using learned feature manifolds," *Comput. Vis. Image Understand.*, vol. 118, pp. 128–139, Jan. 2014.

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

[17] S. Hare, A. Saffari, and P. H. S. Torr, "Efficient online structured output learning for keypoint-based object tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, 2012, pp. 1894–1901.

[18] 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.

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

[20] X. Jia, H. Lu, and M.-H. Yang, "Visual tracking via adaptive structural local sparse appearance model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, 2012, pp. 1822–1829.

[21] Z. Kalal, J. Matas, and K. Mikolajczyk, "P-N learning: Bootstrapping binary classifiers by structural constraints," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, San Francisco, CA, USA, 2010, pp. 49–56.

[22] Z. H. Khan and I. Y.-H. Gu, "Nonlinear dynamic model for visual object tracking on grassmann manifolds with partial occlusion handling," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 2005–2019, Dec. 2013.

[23] J. Lee, M. Cho, and K. M. Lee, "Hyper-graph matching via reweighted random walks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Colorado Springs, CO, USA, 2011, pp. 1633–1640.

[24] W. Li *et al.*, "Online visual tracking using temporally coherent part cluster," in *Proc. IEEE Win. Conf. Appl. Comput. Vis.*, 2015, pp. 9–16.

[25] J. Lim, D. A. Ross, R.-S. Lin, and M.-H. Yang, "Incremental learning for visual tracking," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, 2004, pp. 793–800.

[26] H. Liu, X. Yang, L. J. Latecki, and S. Yan, "Dense neighborhoods on affinity graph," *Int. J. Comput. Vis.*, vol. 98, no. 1, pp. 65–82, 2012.

[27] T. Liu, G. Wang, and Q. Yang, "Real-time part-based visual tracking via adaptive correlation filters," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Boston, MA, USA, 2015, pp. 4902–4912.

[28] X. Mei and H. Ling, "Robust visual tracking using $\ell 1$ minimization," in *Proc. IEEE Int. Conf. Comput. Vis.*, Kyoto, Japan, 2009, pp. 1436–1443.

[29] H. Nam, S. Hong, and B. Han, "Online graph-based tracking," in *Proc. Eur. Conf. Comput. Vis.*, Zürich, Switzerland, 2014, pp. 112–126.

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

[31] G. Nebehay and R. P. Pflugfelder, "Consensus-based matching and tracking of keypoints for object tracking," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Steamboat Springs, CO, USA, 2014, pp. 862–869.

[32] S. Oron, A. Bar-Hillel, D. Levi, and S. Avidan, "Locally orderless tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, 2012, pp. 1940–1947.

[33] X. Ren and J. Malik, "Tracking as repeated figure/ground segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Minneapolis, MN, USA, 2007, pp. 1–8.

[34] 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.

[35] Q. Wang, F. Chen, and W. Xu, "Tracking by third-order tensor representation," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 41, no. 2, pp. 385–396, Apr. 2011.

[36] S. Wang, H. Lu, F. Yang, and M.-H. Yang, "Superpixel tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, Barcelona, Spain, 2011, pp. 1323–1330.

[37] W. Wang and R. Nevatia, "Robust object tracking using constellation model with superpixel," in *Proc. Asian Conf. Comput. Vis.*, Daejeon, South Korea, 2012, pp. 191–204.

[38] L. Wen, Z. Cai, D. Du, Z. Lei, and S. Z. Li, "Learning discriminative hidden structural parts for visual tracking," in *Proc. Workshops Conjunction Asian Conf. Comput. Vis.*, Singapore, 2014, pp. 262–276.

[39] L. Wen, Z. Cai, Z. Lei, D. Yi, and S. Z. Li, "Online spatio-temporal structural context learning for visual tracking," in *Proc. Eur. Conf. Comput. Vis.*, Florence, Italy, 2012, pp. 716–729.

[40] L. Wen, D. Du, Z. Lei, S. Z. Li, and M.-H. Yang, "JOTS: Joint online tracking and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Boston, MA, USA, 2015, pp. 2226–2234.

[41] A. Wendel, S. Sternig, and M. Godec, "Robustifying the flock of trackers," in *Proc. Comput. Vis. Winter Workshop Citeseer*, Mitterberg, Austria, 2011, p. 91.

[42] 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.

[43] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Portland, OR, USA, 2013, pp. 2411–2418.

[44] Y. Xie *et al.*, "Discriminative object tracking via sparse representation and online dictionary learning," *IEEE Trans. Cybern.*, vol. 44, no. 4, pp. 539–553, Apr. 2014.

[45] F. Yang, H. Lu, and M.-H. Yang, "Learning structured visual dictionary for object tracking," *Image Vis. Comput.*, vol. 31, no. 12, pp. 992–999, 2013.

[46] 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.*, Portland, OR, USA, 2013, pp. 2363–2370.

[47] K. M. Yi, H. Jeong, B. Heo, H. J. Chang, and J. Y. Choi, "Initialization-insensitive visual tracking through voting with salient local features," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sydney, NSW, Australia, 2013, pp. 2912–2919.

[48] X. Yu, J. Yang, T. Wang, and T. S. Huang, "Key point detection by max pooling for tracking," *IEEE Trans. Cybern.*, vol. 45, no. 3, pp. 444–452, Mar. 2015.

[49] K. Zhang, L. Zhang, Q. Liu, D. Zhang, and M.-H. Yang, "Fast visual tracking via dense spatio-temporal context learning," in *Proc. Eur. Conf. Comput. Vis.*, Zürich, Switzerland, 2014, pp. 127–141.
[50] K. Zhang, L. Zhang, and M.-H. Yang, "Real-time compressive tracking," in *Proc. Eur. Conf. Comput. Vis.*, Florence, Italy, 2012, pp. 864–877.
[51] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, "Robust visual tracking via multi-task sparse learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, 2012, pp. 2042–2049.
[52] W. Zhong, H. Lu, and M.-H. Yang, "Robust object tracking via sparsity-based collaborative model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, 2012, pp. 1838–1845.
[53] X. Zhou, X. Li, and W. Hu, "Learning a superpixel-driven speed function for level set tracking," *IEEE Trans. Cybern.*, vol. 46, no. 7, pp. 1498–1510, Jul. 2016.

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 graph theory, visual tracking, and video segmentation.

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 video coding and very large scale integration design.

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, 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 in particular.

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, University of Texas at San Antonio, San Antonio, TX, USA. He has published over 340 refereed journal and conference papers. His current research interests include multimedia information retrieval and computer vision.

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

Qingming Huang (SM'08) received the B.S. degree in computer science and 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, 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.

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.