HYBRID STRUCTURE HYPERGRAPH FOR ONLINE DEFORMABLE OBJECT TRACKING

Shengkun Li[1], *Dawei Du*[2], *Longyin Wen*[3], *Ming-Ching Chang*[1], *Siwei Lyu*[1,4]

[1]University at Albany, State University of New York, NY.
[2]University of Chinese Academy of Sciences, Beijing, China.
[3]GE Global Research, Niskayuna, NY.
[4]Tianjin Normal University, Tianjin, China.

ABSTRACT

Recent advances in visual tracking field design part-based model to handle the deformation and occlusion challenges. Previous methods only consider the sole degree of dependencies (*e.g.*, pairwise or high-order dependencies) between object parts in consecutive frames. However, the degree of dependencies of different object parts in consecutive frames are not consistent, especially when large deformation and occlusion happen. To that end, we design a hybrid structure hypergraph based tracker, which use a non-uniform hypergraph to model the dependencies among object parts. The tracking task is further formulated as the dense structures extracting problem on the non-uniform hypergraph, which is solved by an approximate algorithm efficiently. Several experiments are carried out on publicly available online deformable object tracking dataset, *i.e.*, Deform-SOT dataset, to demonstrate the favorable performance of the proposed method against the state-of-the-art online tracking methods.

Index Terms— Deformable object tracking, hybrid structure, non-uniform hyper-graph, dense structures extracting

1. INTRODUCTION

Although significant progress has been achieved in recent years, tracking a deformable object remains a challenging task due to large changes of targets in appearance caused by deformation, occlusions, and clutter background. Most of previous methods focus on modeling the appearance variations of targets in the bounding box, *e.g.*, correlation filters [1, 2], sparse representation [3, 4, 5], online boosting [6, 7], deep learning [8, 9, 10], etc., which suffer from the drift problem due to considerable pixels of the background are encompassed in the bounding box, especially when large deformation and occlusion exist.

To this end, some recent methods use the part-based model to encode the appearance variations of target accurately, *e.g.*, [12, 13, 14, 15]. However, these methods use the bounding box based parts to describe the target appearance, which include background pixels affecting the tracking performance. Another related work [16] design a novel dissimilarity measure for the clustering of correspondences between keypoints between consecutive frames to complete the deformable object tracking task. However, the interest point generating strategy makes it sensitive to local noise in severe occlusion or clutter backgrounds.

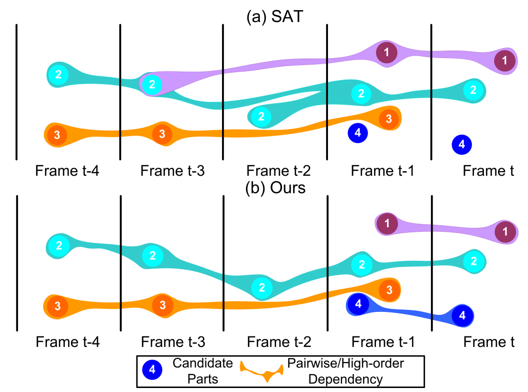

Fig. 1. An example to demonstrate the effectiveness of introducing the non-uniform hypergraph in tracking. We compare (a) SAT [11] that considers fixed dependencies among target parts in a uniform hypergraph and (b) our method that exploits different degree of dependencies in a non-uniform hypergraph. Our method is more effective in handling large deformation and severe occlusion challenges.

Recently, Cai *et al.* [17] use the over-segmented superpixels to replace the bounding box based parts and formulate the deformable object tracking task as tracking the dynamic undirected geometric structure graph of the target. Du *et al.* [18] improve the method [17] by using hypergraph instead of graph to capture the high-order interactions among target parts. Specifically, the geometric hypergraph is constructed and learned to match the target parts and the candidate parts in two consecutive frames. Furthermore, Du *et al.* [11] exploit the high-order constraints of target parts in multiple consecutive frames rather than only two frames in [17, 18] by constructing a *uniform hypergraph*, where each hyperedge

Longyin Wen and Siwei Lyu are the corresponding authors. This work is partially supported by US NSF Research Grant (CCF-1319800).

is constructed by the same number of nodes, and formulate the tracking task as the dense subgraph searching on the hypergraph, as illustrated in Fig. 1(a). However, the dependencies of different object parts in consecutive frames are different with each other, especially when large deformation and occlusion happen. Using the uniform hypergraph is less effective in handling deformation and occlusion challenges in unconstrained scenes.

In this paper, we proposed a hybrid structure hypergraph based tracker, which uses non-uniform hypergraph to describe the dependencies among target parts in consecutive frames, as illustrated in Fig. 1(b). Specifically, we first use the SLIC over-segmentation algorithm [19] to generate the superpixels in each video frame and apply the graph cut algorithm [20] to produce the candidate parts. Then, we construct a non-uniform hypergraph to capture the hybrid dependencies among candidate parts across multiple frames, where each node corresponds to a candidate part, and each edge [1] encodes the dependencies of the parts, *i.e.*, the consistencies in both appearance and motion. Inspired by [21, 11], we propose an approximate algorithm to extract the dense structures on the hypergraph to decide the parts belonging to the target. After that, the target state (*i.e.*, center location in pixel and scale) is determined by analyzing the extracted parts belonging to the target.

The main contributions of this paper are summarized as follows. (1) We formulate the online deformable object tracking task as the dense structure extracting problem on the constructed *non-uniform hypergraph*, which is solved by an approximate algorithm efficiently. (2) To control the scale of the hypergraph for running efficiency, we design an approximate sampling strategy to focus on the edges including the nodes in the current frame in hypergraph construction. (3) Several experiments are carried out on publicly available Deform-SOT dataset, to demonstrate the favorable performance of the proposed method against state-of-the-art trackers.

2. HYPERGRAPH BASED TRACKER

Constructing hypergraph. As discussed above, we process multiple consecutive frames at a time. Similar to [11], we construct a *frame buffer* Γ of size T to store the frames to be processed [2]. Then, we use the SLIC over-segmentation method [19] to generate several superpixels of each frame in the frame buffer Γ, *i.e.*, $\mathcal{P} = \{p_1, \cdots, p_n\}$, where p_i is the i-th generated superpixel, and n is the total number of superpixels. After that, similar to [11], we form the energy function and use the graph cut algorithm [20] to get coarse labeling of each superpixel as belonging to the target or background.

Given the candidate parts of the target, we construct a non-uniform hypergraph $\mathcal{G} = (V, E, \Psi)$ to describe the hybrid dependencies among parts in consecutive frames, where V is the node set of the hypergraph, E is the edge set, and Ψ is the weight set corresponding to the edges. A hypergraph is an extension of a conventional graph, where an edge can connect more than two nodes. In other words, an edge is a subset of nodes. If all edges are constructed by the same number of nodes, the hypergraph is called uniform hypergraph. Otherwise, we call it *non-uniform hypergraph*.

To ensure the running efficiency of the tracker, we design a simple approximate sampling strategy to focus on the edges including the nodes in the current frame. We enumerate degree d, (*i.e.*, $1 \leq d \leq D$ [3]) to generate several edges with different degrees. For the d-th degree edges, we first collect several node arrays from the first $T-1$ frames in Γ. Each node array consists of $d-1$ nodes. Specifically, any two nodes in each node array should not belong to the same frame. Then, we construct the edges with d-degree by combining the collected node arrays with the nodes of the T-th frame in Γ (*i.e.*, the current frame). In contrast to the previous method [11] constructing the edges by enumerating all combinations of the nodes in Γ, the proposed method focuses on the edges including the nodes in the current frame to control the scale of the hypergraph for running efficiency.

The weight of self-loop $\Psi(v_i)$ in the hypergraph indicates the probability of a node belonging to the target. Since we do not have accurate prior information of the nodes, thus we set $\Psi(v_i) = 1.0$ for all nodes (*i.e.*, superpixels) coarsely labeled as the target by the graph cut algorithm. Meanwhile, the pairwise dependency $\Psi(v_i, v_j)$ encodes the appearance similarity between two nodes v_i and v_j, which is calculated as $\Psi(v_i, v_j) = e^{-\lambda_1 \cdot \chi\left(U(v_i), U(v_j)\right)}$, where $U(v_i)$ and $U(v_j)$ are the features corresponding to the node v_i and v_j, respectively, which is constructed by catenating the HSV color histogram and Local Binary Pattern (LBP) texture histogram, $\chi(\cdot, \cdot)$ calculates the chi-squared distance between two features $U(v_i)$ and $U(v_j)$, and λ_1 is the parameter controlling the sensitive of the distance to the pairwise dependency.

Furthermore, the high-order dependency $\Psi(\mathbf{v}_{1:d})$ (*i.e.*, $d \geq 1$) encodes the motion consistency among the nodes $\mathbf{v}_{1:d} = \{v_1, \cdots, v_d\}$. Intuitively, the target parts move smoothly in a short time interval. We compute $\Psi(\mathbf{v}_{1:d})$ based on fitting the motions of nodes $\mathbf{v}_{1:d}$ using quadratic spline interpolation. Specifically, we fit a piecewise parametric curve to a subset of the nodes $\mathcal{S}_{\mathbf{v}} \subset \mathbf{v}_{1:d}$ with equally interval in temporal domain. Then, we calculate the ℓ_2 distance of the center locations of the remaining nodes with their predictions based on the fitted curve to get the high-order dependency,

[1] In this paper, we use the terminology "edge" to indicate the self-loop, conventional edge and hyperedge, and use the terminology "hyperedge" to indicate the edge involving more than two nodes specifically.

[2] Notably, when the latest frame index $t \leq T$, we make $T-t+1$ copies of the previous frames of Γ. When receiving a new frame, we will remove the earliest frame in Γ and add the new receiving frame. In this way, we can process the video online.

[3] Notably, the edges with degree $d = 1$ indicate the self-loops of the graph.

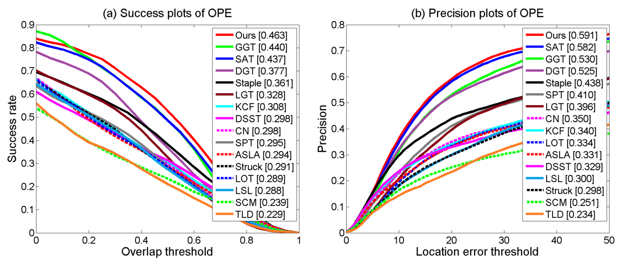

Fig. 2. The success and precision plots over the dataset using OPE. The representative scores for each tracker are reported in the legends.

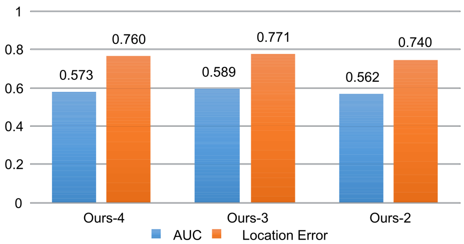

Fig. 3. The tracking performance with different maximal degree of hypergraph.

i.e., $\Psi(\mathbf{v}_{1:d}) = e^{-\lambda_2 \cdot \sum_{v \in \mathcal{S}_\mathbf{v}} \|C(v)-Q(t_v)\|_2^2}$, where $C(v)$ is the center location of node v, t_v is the frame index of node v, $Q(t_v)$ is the predicted location of node v, and λ_2 is the parameter controlling the sensitive of the predicted deviation to high-order dependency.

Extracting dense structures. Inspired by [11], we set each node as a starting point, and search the corresponding dense structure. In this way, we can extract all dense structures on the hypergraph to determine the state of target parts in the current frame. Specifically, for the node v_p, we define the corresponding structure (sub-hypergraph) "dense", if the nodes in the structure are interconnected by a large number of different degree edges with large weights. We assume there exits β nodes in the structure. Let $\mathbf{x}_p = (x_{p,1}, \cdots, x_{p,n})$ to be the indicator variable corresponding to node v_p. That is, $x_{p,i} = 1/\beta$, if the node v_i belongs to the dense structure; otherwise $x_{p,i} = 0$. Thus, we have $\sum_{i=1}^n x_{p,i} = 1$. To extract the dense structure of node v_p, we maximize the weight summation of the edges included in the structure, *i.e.*,

$$\begin{array}{l} \mathbf{x}_p^* = \mathrm{argmax}_{\mathbf{x}_p} \sum_{d=1}^{D} \omega_d \sum_{\mathbf{v}_{1:d} \in V} \Psi(\mathbf{v}_{1:d}) \overbrace{x_{p,1} \cdots x_{p,d}}^{d} \\ \textbf{s.t.} \ \sum_{i=1}^n x_{p,i} = 1, \quad \forall i,\ x_{p,i} \in \{0, 1/\beta\}, \end{array} \tag{1}$$

where D is the maximal degree of the edges in $\mathcal{G}$, and ω_d is the preset influence factors of edges with different degree. Notably, we would like to highlight that the objective function for tracking in [11] is a specific case of (1). That is, if we set $\omega_{d^\star} \neq 0$ for a specific $d^\star \geq 3$, and $\omega_d = 0$, $\forall d \neq d^\star$, the non-uniform hypergraph $\mathcal{G}$ will reduce to a $d^\star$-uniform hypergraph, similar to that in [11].

The optimization problem in (1) is a combinational optimization problem, since we do not know any prior information about the number of nodes β in the dense structure. To reduce the computational complexity, we convert (1) to an approximate optimization problem, *i.e.*, relax the constraint $x_{p,i} \in \{0, 1/\beta\}$ to $x_{p,i} \in [0, 1/\beta]$. Meanwhile, to avoid the degeneration problem, we require that each extracted dense structure include at least $\hat{\beta}$ nodes, *i.e.*, $\beta \geq \hat{\beta}$. Then, the constraint $x_{p,i} \in [0, 1/\beta]$ is converted to $x_{p,i} \in [0, 1/\hat{\beta}]$. We extend the greedy pairwise updating proposed in [22] to extract the dense structure on non-uniform hypergraph efficiently. Specifically, we can increase one component $x_{p,k}$ and decrease another one $x_{p,j}$ appropriately, to increase the objective, *i.e.*, $\Omega(\mathbf{x}_p) = \sum_{d=1}^{D} \omega_d \sum_{\mathbf{v}_{1:d} \in V} \Psi(\mathbf{v}_{1:d}) \overbrace{x_{p,1} \cdots x_{p,d}}^{d}$, while satisfying the constraint $\sum_{i=1}^n x_{p,i} = 1$, *i.e.*, $x'_{p,l} = x_{p,l}, \forall l \neq k, l \neq j$; $x'_{p,l} = x_{p,l} + \alpha$, $l = k$; $x'_{p,l} = x_{p,l} - \alpha$, $l = j$. The different of objective $\Omega(\mathbf{x}_p)$ after updating is $\Omega(\mathbf{x}'_p) - \Omega(\mathbf{x}_p) = \psi_{k,j}(\mathbf{x}_p)\alpha^2 + \big(\phi_k(\mathbf{x}_p) - \phi_j(\mathbf{x}_p)\big)\alpha$, where $\mathbf{x}'_p = (x'_{p,1}, \cdots, x'_{p,n})$, and

$$\begin{array}{l} \psi_{k,j}(\mathbf{x}_p) = -\omega_2 \cdot \Psi(k,j) \\ -\sum_{d=3}^{D} \omega_d \sum_{\mathbf{v}_{1:d-2} \neq k,j} \Psi(\mathbf{v}_{1:d-2}, k, j) \prod_{i=1}^{d-2} x_{p,v_i}, \\ \phi_k(\mathbf{x}_p) = \sum_{d=2}^{D} \omega_d \sum_{\mathbf{v}_{1:d-1} \in V} \Psi(\mathbf{v}_{1:d-1}, k) \prod_{i=1}^{d-1} x_{p,v_i}. \end{array} \tag{2}$$

Thus, to maximize the difference to increase the objective $\Omega(\mathbf{x}_p)$, we can select the updating step α as follows [4].

$$\alpha = \begin{cases} \min(x_{p,j}, \frac{1}{\hat{\beta}} - x_{p,k}), \text{ if } \psi_{k,j}(\mathbf{x}_p) \geq 0; \\ \min\big(x_{p,j}, \frac{1}{\hat{\beta}} - x_{p,k}, \frac{\phi_j(\mathbf{x}_p) - \phi_k(\mathbf{x}_p)}{2 \cdot \psi_{k,j}(\mathbf{x}_p)}\big), \text{ if } \psi_{k,j}(\mathbf{x}_p) < 0; \\ \min(x_{p,j}, \frac{1}{\hat{\beta}} - x_{p,k}), \text{ if } \phi_k(\mathbf{x}_p) = \phi_j(\mathbf{x}_p), \psi_{k,j}(\mathbf{x}_p) > 0. \end{cases} \tag{3}$$

In this way, we can use the similar heuristic strategy in [22] to compute the local maximizer $\mathbf{x}_p^*$ for the dense structure extraction. After extracting the dense structures on the non-uniform hypergraph, we use the similar strategy proposed in [11] to estimate the target state and update the appearance model online. In this way, we can complete the object tracking task effectively.

3. EXPERIMENT

Dataset. We carry out several experiments on the publicly available online deformable object tracking dataset, *i.e.*, Deform-SOT dataset presented in [11], to demonstrate the performance of the proposed method. The Deform-SOT dataset consists of 50 challenging sequences including large deformation, severe occlusion, abnormal movement, etc.

Experimental setup. We implement the proposed tracker in with Matlab and C++ [5], which can be run at 1.0 frames per second (fps) on a machine with a 3.4 GHz Intel i7 processor

[4] Notably, we can assume $\phi_k(\mathbf{x}_p) > \phi_j(\mathbf{x}_p)$. If $\phi_k(\mathbf{x}_p) < \phi_j(\mathbf{x}_p)$, we exchange indexes k and j to maximize the difference $\Omega(\mathbf{x}'_p) - \Omega(\mathbf{x}_p)$.

[5] The code is available at www.cbsr.ia.ac.cn/users/lywen.

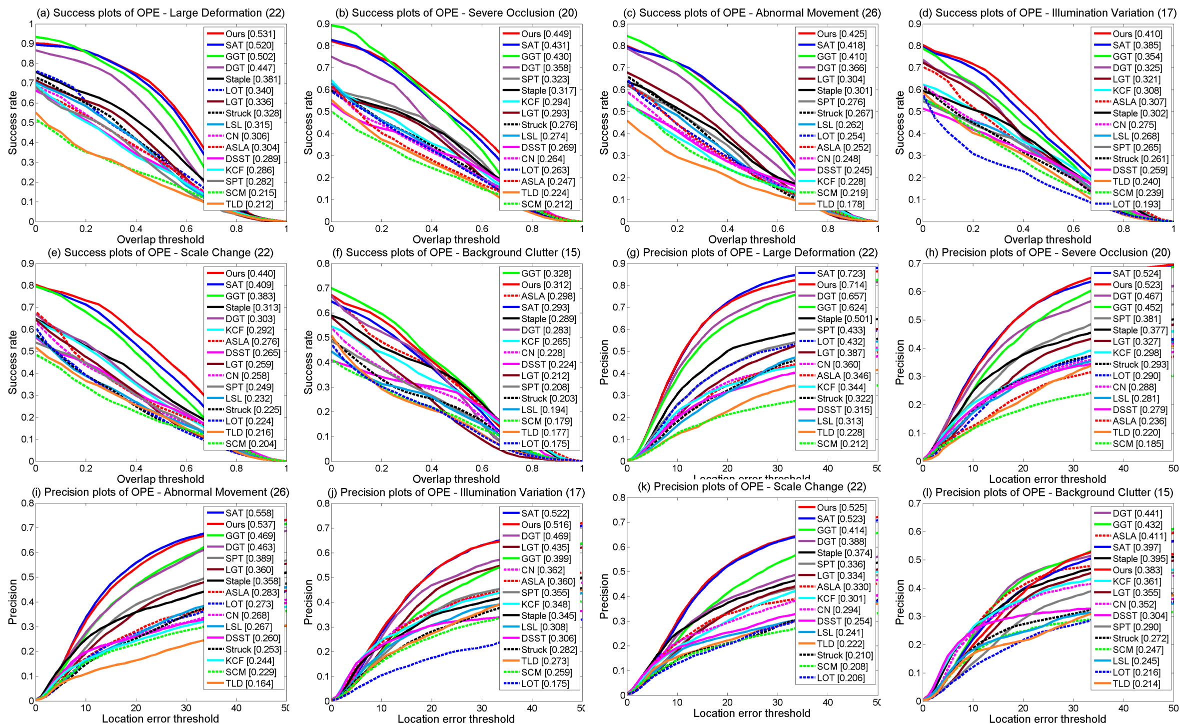

Fig. 4. The plots of OPE with different attributes over the dataset. The representative scores for each tracker are reported in the legends.

and 8 GB memory. To evaluate the performance of trackers, we run One-Pass Evaluation (OPE) [23], which uses the ground truth in the first frame to initialize the trackers. Two measures, namely success plot and precision plot, are used to compare performances. The Area Under Curve (AUC) of each success plot is used to represent the performance of the tracking algorithms. The precision score of the center location error for the threshold equals to 20 pixels is also reported to indicate the tracking performance comprehensively.

We study the effect of a critical parameter, *i.e.*, the maximal degree D of the non-uniform hypergraph $\mathcal{G}$ to the tracking performance. To examine the effect of D, we report the performance of the proposed method on 15 sequences from Deform-SOT dataset with different maximal degree, *i.e.*, Ours-D ($D = 2, 3, 4$) in Fig. 3. As shown in Fig. 3, we find that Ours-2 performs worse than Ours-3, since the high-order dependencies among parts (*i.e.*, the information of motion consistency) are considered to improve the performance. Meanwhile, Ours-3 performs better than Ours-4, because the hybrid structure hypergraph with the degree that is too high fails to adapt the large deformations and occlusions of target parts effectively. Thus, we choose the maximal degree $D = 3$ in our experiments. Other parameters in our algorithm are chose empirically using the grid search strategy, *i.e.*, $\lambda_1 = 1.0$ in pairwise dependency, $\lambda_2 = 0.0625$ in high-order dependencies, and $\omega_1 = 1.0$, $\omega_2 = 3.0$, and $\omega_d = 9.0$ ($d \geq 3$) in (1). The minimal size of the searched dense structures $\hat{\beta} = 2$.

Quantitative evaluation. To demonstrate the favorable performance of the proposed method, we evaluate the proposed method against 15 trackers, (*i.e.*, GGT [18], SAT [11], DGT [17], Staple [24], LGT [25], KCF [2], DSST [26], CN [27], SPT [28], ASLA [4], Struck [29], LOT [30], LSL [13], SCM [31], and TLD [32]) with the top performance on the Deform-SOT dataset. As shown in Fig. 2, the success and precision plots of the overall dataset indicate that our method performs favorable against 15 state-of-the-art trackers, which demonstrates the effectiveness of using non-uniform hypergraph in tracking task. Our method achieves the best performance in 5 challenging scenarios out of 6 according to the success plot, especially when severe occlusion and large deformation challenges happen, as presented in Fig. 4. In summary, hybrid structure hypergraph considers several degrees of dependencies among target parts jointly, which is more accurate to describe the dependencies among different parts to improve the tracking performance, comparing to the uniform hypergraph in [11].

4. CONCLUSION

In this paper, we propose a hybrid structure hypergraph to complete the online deformable object tracking task, which formulates the tracking task as the dense structure extracting problem on a non-uniform hypergraph, solved by an approximate algorithm efficiently. Experimental results on the public available online deformable object tracking dataset, *i.e.*, Deform-SOT dataset, demonstrate that our method performs favorable against state-of-the-art methods.

5. REFERENCES

[1] Martin Danelljan, Gustav Häger, Fahad Shahbaz Khan, and Michael Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *ICCV*, 2015, pp. 4310–4318. 1

[2] João F. Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista, "High-speed tracking with kernelized correlation filters," *TPAMI*, vol. 37, no. 3, pp. 583–596, 2015. 1, 4

[3] Xue Mei and Haibin Ling, "Robust visual tracking using $\ell 1$ minimization," in *ICCV*, 2009, pp. 1436–1443. 1

[4] Xu Jia, Huchuan Lu, and Ming-Hsuan Yang, "Visual tracking via adaptive structural local sparse appearance model," in *CVPR*, 2012, pp. 1822–1829. 1, 4

[5] Tianzhu Zhang, Bernard Ghanem, Si Liu, and Narendra Ahuja, "Robust visual tracking via multi-task sparse learning," in *CVPR*, 2012, pp. 2042–2049. 1

[6] Boris Babenko, Ming-Hsuan Yang, and Serge Belongie, "Robust object tracking with online multiple instance learning," *TPAMI*, vol. 33, no. 8, pp. 1619–1632, 2011. 1

[7] Longyin Wen, Zhaowei Cai, Zhen Lei, Dong Yi, and Stan Z. Li, "Robust online learned spatio-temporal context model for visual tracking," *TIP*, vol. 23, no. 2, pp. 785–796, 2014. 1

[8] Chao Ma, Jia-Bin Huang, Xiaokang Yang, and Ming-Hsuan Yang, "Hierarchical convolutional features for visual tracking," in *ICCV*, 2015, pp. 3074–3082. 1

[9] Hyeonseob Nam and Bohyung Han, "Learning multi-domain convolutional neural networks for visual tracking," in *CVPR*, 2016, pp. 4293–4302. 1

[10] Ran Tao, Efstratios Gavves, and Arnold W. M. Smeulders, "Siamese instance search for tracking," in *CVPR*, 2016, pp. 1420–1429. 1

[11] Dawei Du, Honggang Qi, Wenbo Li, Longyin Wen, Qingming Huang, and Siwei Lyu, "Online deformable object tracking based on structure-aware hyper-graph," *TIP*, vol. 25, no. 8, pp. 3572–3584, 2016. 1, 2, 3, 4

[12] Junseok Kwon and Kyoung Mu Lee, "Tracking of a non-rigid object via patch-based dynamic appearance modeling and adaptive basin hopping monte carlo sampling," in *CVPR*, 2009, pp. 1208–1215. 1

[13] Rui Yao, Qinfeng Shi, Chunhua Shen, Yanning Zhang, and Anton van den Hengel, "Part-based visual tracking with online latent structural learning," in *CVPR*, 2013, pp. 2363–2370. 1, 4

[14] Yang Li, Jianke Zhu, and Steven C. H. Hoi, "Reliable patch trackers: Robust visual tracking by exploiting reliable patches," in *CVPR*, 2015, pp. 353–361. 1

[15] Alan Lukezic, Luka Cehovin, and Matej Kristan, "Deformable parts correlation filters for robust visual tracking," *CoRR*, vol. abs/1605.03720, 2016. 1

[16] Georg Nebehay and Roman P. Pflugfelder, "Clustering of static-adaptive correspondences for deformable object tracking," in *CVPR*, 2015, pp. 2784–2791. 1

[17] Zhaowei Cai, Longyin Wen, Zhen Lei, Nuno Vasconcelos, and Stan Z. Li, "Robust deformable and occluded object tracking with dynamic graph," *TIP*, vol. 23, no. 12, pp. 5497–5509, 2014. 1, 4

[18] Dawei Du, Honggang Qi, Longyin Wen, Qi Tian, Qingming Huang, and Siwei Lyu, "Geometric hypergraph learning for visual tracking," *TCYB*, 2016. 1, 4

[19] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Susstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *TPAMI*, vol. 34, no. 11, pp. 2274–2282, 2012. 2

[20] Yuri Boykov and Vladimir Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *TPAMI*, vol. 26, no. 9, pp. 1124–1137, 2004. 2

[21] Longyin Wen, Wenbo Li, Junjie Yan, Zhen Lei, Dong Yi, and Stan Z Li, "Multiple target tracking based on undirected hierarchical relation hypergraph," in *CVPR*, 2014, pp. 1282–1289. 2

[22] Hairong Liu, Xingwei Yang, Longin Jan Latecki, and Shuicheng Yan, "Dense neighborhoods on affinity graph," *IJCV*, vol. 98, no. 1, pp. 65–82, 2012. 3

[23] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang, "Online object tracking: A benchmark," in *CVPR*, 2013, pp. 2411–2418. 4

[24] Luca Bertinetto, Jack Valmadre, Stuart Golodetz, Ondrej Miksik, and Philip H. S. Torr, "Staple: Complementary learners for real-time tracking," in *CVPR*, 2016, pp. 1401–1409. 4

[25] Luka Cehovin, Matej Kristan, and Ales Leonardis, "Robust visual tracking using an adaptive coupled-layer visual model," *TPAMI*, vol. 35, no. 4, pp. 941–953, 2013. 4

[26] Martin Danelljan, Gustav Häger, Fahad Shahbaz Khan, and Michael Felsberg, "Accurate scale estimation for robust visual tracking," in *BMVC*, 2014. 4

[27] Martin Danelljan, Fahad Shahbaz Khan, Michael Felsberg, and Joost Van de Weijer, "Adaptive color attributes for real-time visual tracking," in *CVPR*, 2014. 4

[28] Shu Wang, Huchuan Lu, Fan Yang, and Ming-Hsuan Yang, "Superpixel tracking," in *ICCV*, 2011, pp. 1323–1330. 4

[29] Sam Hare, Amir Saffari, and Philip HS Torr, "Struck: Structured output tracking with kernels," in *ICCV*, 2011, pp. 263–270. 4

[30] Shaul Oron, Aharon Bar-Hillel, Dan Levi, and Shai Avidan, "Locally orderless tracking," in *CVPR*, 2012, pp. 1940–1947. 4

[31] Wei Zhong, Huchuan Lu, and Ming-Hsuan Yang, "Robust object tracking via sparsity-based collaborative model," in *CVPR*, 2012, pp. 1838–1845. 4

[32] Zdenek Kalal, Jiri Matas, and Krystian Mikolajczyk, "P-N learning: Bootstrapping binary classifiers by structural constraints," in *CVPR*, 2010, pp. 49–56. 4