UUCT - HyMP: Towards Tracking Dispersed Crowd Groups from UAVs

Tonmoay Deb*, Mahieyin Rahmun, Shahriar Ali Bijoy, Mayamin Hamid Raha, and Mohammad A Khan

***Abstract*— Aerial tracking of dispersed crowd groups with a single target window is a novel and one of the most challenging problems in Computer Vision and Robotics. Considering crowd group as a multi-object tracking problem can often lead to computational burden and frequent target mismatch due to numerous occlusions, whereas a single window can efficiently focus on the target. Recent progress on single object tracking (SOT) algorithms is achieved by learning a generic discriminator model from object tracking datasets, continuously updated during the testing steps. However, while tracking a group of crowd with a single window, the rigid discriminator can not generalize frequent group reformation, binomial dispersion, and crowd shape changes due to less knowledge about human-to-human interactions. To alleviate the issues, we propose a novel photo-realistic Unreal UAV Crowd Tracking (UUCT) dataset, which benchmarks aerial crowd group movements into several attributes. Second, we formulate a novel algorithm, Hybrid Motion Pooling (HyMP), which extends the existing SOT algorithm, DiMP, by exploiting graph convolutional networks for learning human groups and low-rank bilinear pooling for capturing temporal group reformations end-to-end. Then, we compare HyMP with state-of-the-art (SOTA) trackers on UUCT to demonstrate HyMP's effectiveness in group tracking. Also, we illustrate the generalizability of HyMP by evaluating on the existing benchmarks. On average, HyMP outperforms SOTA approaches by 7.5% on UUCT and 4.3% on related datasets.**

I. Introduction

Visual Object Tracking (VOT) is one of the fundamental and active research fields in computer vision. VOT is still an open problem, with many applications in robotics, autonomous systems, and surveillance [1], [2], [3], [4]. Despite having significant progress in recent years [5], [6], [7], [8], the problem yet remains challenging due to several limiting factors including occlusion, appearance variation, motion abruptness, fast scale change, etc. To alleviate the issues, some recent works introduced several invariant feature descriptors, e.g., Histogram of Oriented Gradients (HOG) [9], pre-trained Convolutional Neural Networks (CNN) [5], and adaptively learned lightweight CNN [10] weights. These descriptors are widely used in correlation filters for target translation [9], [11], [12]. The correlation filter-based approach is effective due to the high tracking speed for using correlation computation over the Fourier domain. However, having high learning rates for filter update leads it to usually failing to maintain long-term memory target appearance, which was further alleviated by Siamese Networks [13], [14] and other deep learning approaches [15], [16], [17] by designing a robust discriminator model to distinguish target from the background. However, these are still prone to multiple instances of the same target, shape deformation, and viewpoint changes, leading to domain-specific visual trackers', i.e., algorithms for a specific purpose design.

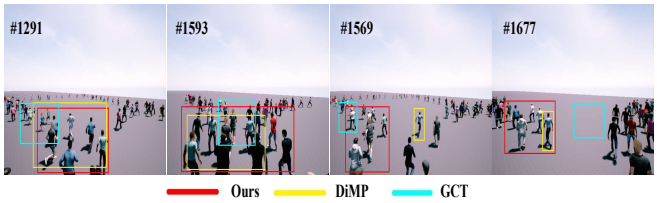


Fig. 1: Comparison of our HyMP tracker with state-of-the-art SOT DiMP [16] and GCT [18]. For both sequences (1291-1593 and 1569-1677), DiMP and GCT fail to detect crowd reformation and translate wrong target window.

Unmanned Aerial Vehicles (UAVs) with vision have unveiled a new research direction to many novel applications, including surveillance [19], wild-life monitoring [20], crowd tracking, and aerial cinematography [21]. Aerial tracking can achieve capabilities to be applied in a diverse set of objects, including animals, boats, humans, which might be difficult to achieve from the ground, a major reason behind the development of unified tracking algorithms for UAVs. Recently, several generic benchmark datasets including UAV123 [22], UAVDT [23], DTB70 [24] have been proposed to evaluate tracking algorithms from UAVs. To this end, a specific benchmark for tracking human-formed crowd group is required as it has emerged into diverse applications.

Tracking a crowd group of humans from the aerial viewpoint is a fairly new problem in the community. The primary goal of crowd group tracking includes focusing on a target group, regardless of binomial dispersion from the track window, which can be easily formulated as a multi-object or person (MOT) tracking task [25], [26], [27]. However, we argue that, when a set of crowd moves, there would be frequent group formation and deformation by humans. This phenomenon would require tracking a large crowd sample and prone to mislead the MOT algorithm with occlusions and unpredictable motion changes [28]. As a crowd group movement belongs to almost identical motion instances, a well-trained single object tracker (SOT) can track with a single yet larger bounding box. This approach introduces additional complexity of crowd dispersion, continuous reformation, out of camera view, global crowd target translation mismatch, and exhaustive scale update. Existing SOTs would largely fail to meet the above constraints, because box prediction and scale estimation approaches of existing algorithms usually designed for single objects [29] which do not face any abrupt

Authors are with the Department of Electrical & Computer Engineering, North South University, Dhaka 1229, Bangladesh {tonmoay.deb, mahieyin.rahman, shahriar.ali, mayamin.raha, mohammad.khan02}@northsouth.edu
*Corresponding author

978-0-7381-3366-9/21/$31.00 ©2021 IEEE

Fig. 2: Five random frames (first row) and corresponding segmentation images (second row) of our UUCT dataset.

shape reformation over frames. As a result, both correlation filter [30], [31], [32], [12] and deep learning [16], [15], [5], [33] trackers provide wrong target box prediction and scale. Moreover, according to recent evaluations on aerial tracking datasets [22], [23], [24], very few trackers became successful in the long-term tracking having false target prediction and inaccurate scale change estimation, accordingly. To alleviate both challenges, we propose a novel, photo-realistic crowd group tracking dataset, Unreal UAV Crowd Tracking (UUCT), which consists of 70 long-term sequences of crowd group distributed in 7 distinct attributes, captured from UAVs. To our best knowledge, UUCT is the first dataset for aerial crowd group tracking. Furthermore, we propose a novel tracking architecture that learns human-to-human interaction without direct supervision using graph convolution [34] and results in better bounding box regression on the group with long-term tracking in the pipeline. Our four-fold contributions are summarized below:

- We introduce crowd group tracking problems from aerial viewpoints, followed by a photo-realistic benchmark dataset, UUCT.
- We evaluate state-of-the-art algorithms' performance and unveil their major limitations on the UUCT dataset.
- We exploit human-human interaction and crowd group formation learning using graph and propose a novel approach, Hybrid Motion Pooling (HyMP) on DiMP [16], to track crowd groups with a single target window.
- We evaluate our proposed algorithm in SOT benchmarks including UUCT to demonstrate our algorithm's generalizability and robustness compared to existing SOT approaches, including our baseline tracker DiMP.

II. Related work

A. Existing Tracking Benchmarks

Along with UAV-only tracking datasets such as UAV123, UAV20L [22], and DTB70 [24], we benchmark our proposed algorithm on widely used OTB100 [35] and VOT2018 [36]. The reasons behind these selection is described below:

OTB100 [35]: This tracking dataset is composed of 100 SOT, non-UAV sequences distributed in 11 attributes. Some of the tracking sets have taken from aerial viewpoints.

VOT2018 [36]: VOT2018 is 2018 version from the Visual Object Tracking challenge with 60 short, yet, challenging videos, annotated with rotating bounding boxes.

UAV123 and UAV20L [22]: This dataset comprise 123 short and 20 long tracking sequences taken from the UAVs, annotated in 12 attributes, with 8 sequences using simulators.

DTB70 [24]: DTB70 is a challenging dataset with 70 UAV tracking sets, where the UAV also moves during target tracking. In UUCT, we included 55 UAV motion scenarios to ensure the robustness while benchmarking the algorithms.

Table I compares these datasets with UUCT on several parameters including total length, attributes, and clip duration. In terms of total duration and frames, UUCT excels all by a wide margin. Moreover, UUCT holds mean 2535 frames, which is far higher than other datasets except for UAV20L, has only 20 long-term sequences compared to 70 by UUCT.

B. Previous Tracking Algorithms

Correlation Filter was popularized with seminal work MOSSE [37], extended by multi-scale correlation filter proposed on the DSST tracker [9], which computes adaptive filters using HOG features, including a single-dimensional filter to handle scale change responses. To alleviate the limitation of context information, online update, long-term tracking, and target drift, background-aware [31], context-aware [32], spatially regularized [30] filters have been proposed. The transition from correlation filters to deep learning started with applying correlation filters over deeply learned image features than traditional hand-crafted features. HCFT [38] algorithm employed pre-trained CNN layer weights hierarchically to learn multiple correlation filters. LCT [5] tracker addressed the limitation of long-term memory of target appearance mismatch and proposed target re-localization. Later, end-to-end tracking have initiated with siamese networks [13], followed by improvements with graph convolutions [34], robust scale update [15], and target regression [16].

C. Spatio-Temporal Graphs

Since graph has been explored for scene representation and retrieval in images [39], [40], [41], it had caught attention for several tasks including person re-identification [42], skeleton-based action recognition [43], video representation [44], localization [45], [46], visual relationship [47], etc. Recently, [18] performed visual tracking using graph convolutional networks [34] which leveraged straightforward graph feature concatenation. However, instead of regular fusion, we apply computationally efficient low-rank bilinear pooling [48] to capture temporal crowd reformation with higher order interaction, discussed in detail on section IV-D.

III. Proposed UUCT Dataset

UUCT aims to design a robust benchmark to facilitate crowd group tracking algorithm development. Figure 2 contains some random frames from UUCT. This section discusses four key protocols of dataset development as well as simulation setup and ground truth generation methods.

- **Photorealistic Simulation:** In order to keep the dataset congruent with the real world, we study real world scenarios thoroughly to develop diverse human characters with realistic movement. We are also able to obtain segmentation and depth images along with RGB frames.

TABLE I
COMPARISON OF UUCT WITH STATE-OF-THE-ART RELATED BENCHMARK (BOTH UAV AND GENERIC) DATASETS IN THE LITERATURE

Benchmark	**OTB100** [35]	**VOT2018** [36]	**UAV123** [22]	**UAV20L** [22]	**DTB70** [24]	**UUCT** (ours)
Number of Videos	100	60	**123**	20	70	70
Minimum Frames	71	41	109	1717	-	1334
Mean Frames	590	356	915	**2934**	-	2535
Maximum Frames	3872	1500	3085	5527	-	4283
Total Frames	59K	21K	113K	59K	-	**161.6K**
Total Duration (minutes)	32.8	11.9	62.5	32.6	-	**89.8**
Frame Rate	30 FPS	30 FPS	30 FPS	30 FPS	60 FPS	30 FPS
Num. of Attributes	11	N/A	**12**	**12**	11	7
UAV Sequences (Yes/No)	No	No	Yes	Yes	Yes	**Yes**
Segmentation (Yes/No)	No	No	Yes*	Yes*	No	**Yes**
Depth Image (Yes/No)	No	No	No	No	No	**Yes**

- **Long-Term Tracking:** We ensure long-term tracking on each sequence to capture abrupt crowd group reformations. According to Table I, each UUCT sequence holds an average of 2535 frames on 30 FPS.
- **Attribute Diversity:** We build UUCT on 7 distinct attributes described in Table II, where same videos can fall into multiple attributes, ensuring sequence diversity.
- **High-Quality Ground Truth (GT):** One of our key goals is to compute GT without dependency on human input. The last column of Table II presents category-wise box overlap error [15] (lower is better) between our GT generator and human annotation on 70 videos.

A. Data Collection: Simulation Setup

We developed the simulator using Unreal Engine [1] with Airsim [49], which consists of a hardware-in-the-loop simulation to easily transfer control to a UAV hardware. We used Adobe Mixamo [2] for character modeling and animations. AI Behavior Toolkit [3] has been utilized to define the behavior of the non-player characters, i.e., humans, when they are spawned into the simulator world. Then, predefined pathways were laid out for humans to follow and move accordingly. In some levels no paths have been set to enable random human movement. We set simulator world to have only humans and no environmental clutter, which is arguably important for initial benchmarking. We also ensured the humans move at variable speeds, including suspended motion, slow walk, and moderate jogging. To capture the frames, we have utilized Airsim API to deploy single UAV in our pre-generated levels and fetch RGB, segmentation, and depth image sequences, which we normalize to a framerate of 30 later on.

B. Annotation: Ground Truth Design Principles

We leverage the segmentation images to generate ground truth boxes of crowd group. Initially, we mask the humans using previously set segmentation color code, followed by grouping using contour search. Then, we apply the principles below to determine ground truth:

- When all segmentation bounding boxes are clustered within a certain location with frequent overlaps according to Figure 3 (a) or aligned diagonally as stated in Figure 3 (b), the ideal final box in these cases would be a placement where the Intersection over Union (IoU) of the final box with each segmentation boxes maximized.
- When the segmentation bounding boxes have extreme relative distances between themselves as shown in Figure 3 (c), the ideal final box placement would be where it would cover any one among the boxes, but not both.

[1] https://www.unrealengine.com
[2] https://www.mixamo.com
[3] https://www.youtube.com/watch?v=BpbXnaTh-sk

IV. HYBRID MOTION POOLING ARCHITECTURE

A. Baseline: Discriminative Model Prediction

Discriminative Model Prediction (DiMP) tracker [16] incorporates two fundamental components, model initializer and optimizer. The initializer captures the target object properties where the optimizer updates the model iteratively by discriminating between the target and background information as a robust classifier. The trained model weight $f = \rho(\chi_{sample})$ is the convolutional filter which is learned from the sample training pairs $\chi_{sample} = \{(v_j, c_j)\}_{j=1}^{m}$ with a model predictor network ρ. Here, v_j are the feature set extracted by a pre-trained CNN δ [50] and $c_j \in \mathbb{R}^2$ is the centroid of the box, respectively. f is further utilized for target localization, learned using the following squared loss:

$$\ell(f) = \frac{1}{|\chi_{sample}|} \sum_{j=1}^{m} \|\nabla(v_j * f, c_j)\|^2 + \|\tau f\|^2 \quad (1)$$

Here, $*$ is the convolution operator between v_j and f, further passed to a residual function ∇ to calculate the difference between the predicted output from the convolution and the Gaussian [37] of the actual centroid of the box c_j. f is iteratively updated offline based on m samples as both training and testing from COCO [51], TrackingNet [52], LaSOT [53], and GOT10k [54] dataset with backbone

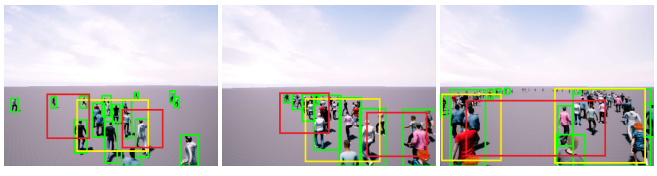


Fig. 3: Principles for generating ground truth. Derived segmentation boxes are in green. Yellow and Red boxes are preferred and rejected candidates.

TABLE II
ATTRIBUTE DETAILS WITH CLIP NUMBERS ***C***, TOTAL DURATION IN MINUTES ***D***, AND OVERLAP ERROR ***OE*** TO BENCHMARK THE UUCT DATASET

UUCT Dataset Attribute Names	Attribute Description	*C*	*D*	*OE*
CSMO: Crowd Split and Merge Occlusion	A moving crowd splits into two distinct crowd groups and merge into one	21	27.1	0.59
CSDOV: Crowd Split and Dispersed Out of View	Similar to CSMO, but one of the groups leaves the field-of-view (FOV) of UAV	21	27.7	0.76
SUMC: Static UAV and Moving Crowd	Crowds move within the FOV of a static UAV positioned at a fixed height	15	18.7	1.34
MUMC: Moving UAV and Moving Crowd	Both the UAV and the crowd are moving in certain directions	55	71.8	2.23
CSSRM: Crowd Sudden Stop and Resume Movement	A crowd starts moving, suddenly stops and resumes movement	14	16.9	1.38
JCM: Jittery Camera Motion	The UAV uses fast, random rotation causing jerky camera motion	36	43.1	1.40
SV: Scale Variation	Bounding box around the crowd alters notably compared to first frame	45	55.1	1.64

weight from ImageNet data [55]. Moreover, bounding box regression [15] is performed in the second stage using the loss between predicted box from the test sample and ground truth box IoU and added with the regular loss ℓ. During online tracking, f is updated after each empirical timestep. Despite sufficient discrimination capabilities between target and background, DiMP has two fundamental drawbacks. First, the algorithm counts the target window as objective and learns to find a particular region with the greatest correlation response. This property will objectively down perform when several single-category objects, i.e., humans, form groups, and continue frequent deformation. Second, f is trained on a fixed set of object movement information, which lacks crowd group data mentioned earlier. Because crowd groups impose radically distinguishable movement patterns across space and time On the other hand, crowd groups' movement can not be easily discriminated due to frequent spatial interaction. Also, crowd groups deform in uncountable patterns, as discussed in Figure 1. We propose a novel approach by formulating the multiple-human interaction in spatial graph Φ^S, followed by incorporating their motion information and direction with hybrid motion pooling using temporal graph Φ^T. The resultant knowledge captures crowd interaction patterns jointly with the filter f, which results in robust, single box crowd group localization and tracking.

B. Spatial Graph Learning

To capture spatial interaction between the human agents along with group detection, we design a spatial graph, Φ^S. At time step t, the frame f_t is passed through a object detector network to form $\gamma_t = \{(\vartheta_t^1, b_t^1), (\vartheta_t^2, b_t^2), ..., (\vartheta_t^n, b_t^{n_t})\}$, where $\vartheta_t^i \in \mathbb{R}^{1\times d}$ denotes d dimensional object features and $b_t^i \in \mathbb{R}^4$ is respective bounding box coordinates at frame t. We further truncate γ_t and keep the objects only labeled as $person$. To capture interaction between the persons, we observe that, people having relative closer spatial location are more likely to form groups. Based on the assumption, we build spatial graph by connecting the bounding boxes b_t^i with respective normalized Intersection over Union (IoU) ζ:

$$\Phi^S_{t_{ij}} = \frac{e^{\zeta_t^{ij}}}{\sum_{j=1}^{n_t} e^{\zeta_t^{ij}}} \tag{2}$$

Here, $\Phi^S_{t_{ij}} \in \mathbb{R}^{n\times n}$ is the spatial adjacency matrix element at position (i, j), captures the interaction between two persons with $\zeta_t^{ij} = IoU(b_t^i, b_t^j)$. According to prior practices [56], [57], we incorporate softmax function for normalization rather than adding identity matrix to the diagonals. The interaction information from $\Phi^S_{t_{ij}}$ indicated potential crowd groups among the detected persons, followed by their global movement pattern, captured by motion pooling.

C. Adaptive Crowd Group Learning

While spatial graph can detect crowd groups and appearance changes, it can not discriminate between human movements and group deformation over time. At first, we exploit the temporal information by connect the frames through pair-wise cosine similarity of the features ϑ_t^i and build temporal graph Φ^T as:

$$\Phi^T_{t_{ij}} = \frac{e^{\cos(\vartheta_t^i, \vartheta_{t+1}^j)}}{\sum_{j=1}^{n_t+1} e^{\cos(\vartheta_t^i, \vartheta_{t+1}^j)}} \tag{3}$$

Here, $\Phi^T_{t_{ij}} \in \mathbb{R}^{n\times n+1}$ defines temporal adjacency matrix element at position (i, j), captures the crowd movement by connecting the persons over time step t using cosine similarity function $\cos(\vartheta_t^i, \vartheta_{t+1}^j)$ between two vectors.

The temporal and spatial adjacency matrix elements, i.e., edges aggregate a fully connected graph over time. After that, the spatio-temporal graph, Φ^{ST} is built by merging both spatial and temporal matrices as below:

$$\Phi^{ST} = \begin{bmatrix} \Phi_1^S & \Phi_1^T & 0 & \cdots & 0 \\ 0 & \Phi_2^S & \Phi_2^T & \cdots & 0 \\ 0 & 0 & \Phi_3^S & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & \cdots & \Phi_t^S \end{bmatrix} \tag{4}$$

Here, $\Phi^{ST} \in \mathbb{R}^{N\times N}$ is composed of Φ_t^S and Φ_t^T. $N = \sum_{t=1}^{T} n_t$ is total number of detected persons over the frames, $1-T$. The 0s are the zero-valued matrices, varies shaped by adjacent spatial and temporal matrices.

Now, we update the graph by using the standard Graph Convolution Network (GCN) [34] as below:

$$\mathbb{C}^{l+1} = \sigma(\mathbb{C}^l + \lambda^{-\frac{1}{2}}\Phi^{ST}\lambda^{-\frac{1}{2}}\mathbb{C}^l W^l) \tag{5}$$

Here, $W^l \in \mathbb{R}^{dm\times dm}$ is the feature matrix at layer l, dm is the dimension defined by the model. λ is the diagonal degree matrix, where $\lambda_{ii} = \sum_j \Phi_{ij}^{ST}$. Due to the effectiveness of nonlinear activation function [43], we use ReLU as activation

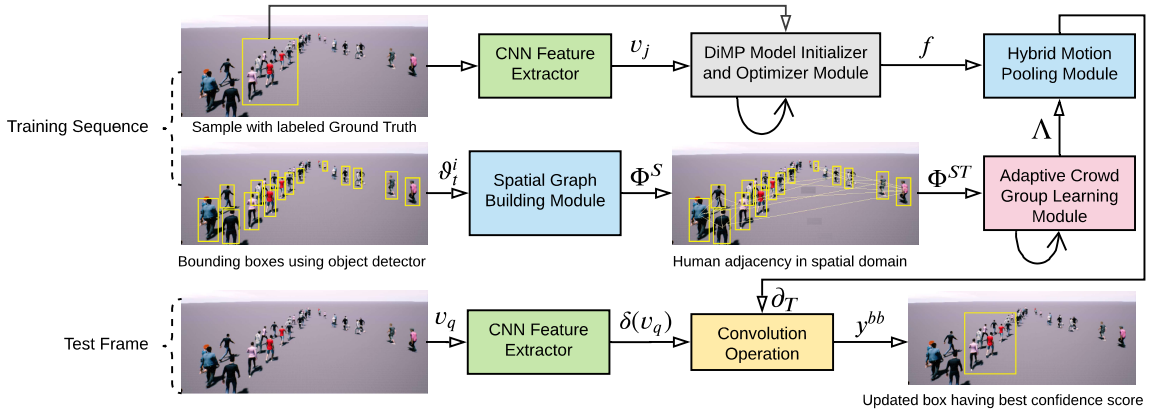


Fig. 4: An end-to-end pipeline of continuous crowd target y^{bb} translation using HyMP architecture. Initially, the target's global image feature and box is trained using DiMP (2nd left row). The same sequence's human bounding box coordinates (bottom left row) and features for each box are passed to the graph building module. Later, learned representation is fused with DiMP features using hybrid motion pooling, further convolving with each test frame (top left row), consecutive locations are predicted. The self-loop on DiMP optimizer and adaptive crowd learning module iteratively update ∂_T.

function σ. $\mathbb{C}^l + 1 \in \mathbb{R}^{N \times dm}$ is the activation from layer $l+1$, formulated by $^{1\times1\times1}$ convolution on the $\mathbb{C}^l$, followed by multiplication with $\lambda^{-\frac{1}{2}} \Phi^{ST} \lambda^{-\frac{1}{2}}$. $\mathbb{C}^0$ is the stacked γ_0 to bootstrap the features, defined as: $\mathbb{C}^0 = \cup \gamma W^0$, where $W^0 \in \mathbb{R}^{d \times dm}$ transforms the features ϑ_t^i from d dimension to dm, followed by a row-wise feature aggregation, $\cup$ on all time steps. Further, average pooling is performed on the L^{th} activation layer, $\mathbb{C}^L$. We denote the resultant matrix as $\Lambda \in \mathbb{R}^{T \times dm}$, which comprises the latent human interaction and crowd group motion information.

D. Hybrid Motion Pooling

To calculate fine-grained target locations, we incorporate the learned filter f from DiMP to interact with Λ at each time step and result in a refined model that holds object properties along with the direction from the graph information. We perform low-rank bilinear pooling to connect Query f with Value Λ in higher-order interaction by:

$$\beta_i = \sigma\left(W_f f\right) \odot \sigma\left(W_\Lambda \Lambda_i\right) \tag{6}$$

Here, $W_f \in \mathbb{R}^{k \times D_f}$ and $W_\Lambda \in \mathbb{R}^{k \times D_\Lambda}$ are the embedding matrices that projects f and Λ into an unified dimension $\mathbb{R}^k$. σ is the activation function. The projected f performs element-wise multiplication, $\odot$, with each projected key σ_i. The bilinear pooled features are further aggregated along with the temporal axis to present a compact representation:

$$\partial_T = MaxPool_T\left([\beta_1, \beta_2, \ldots, \beta_{T-1}]\right) \in \mathbb{R}^{1 \times D_\Lambda} \tag{7}$$

Here the $MaxPool_T$ operation is operated in the time domain T. However, the value of T varies during training and testing does not affect the final filter ∂_T dimension. ∂_T later convolves with the test feature and localize crowd targets.

E. Training and Tracking

1) Offline Training: We train the network based on two distinguished datasets. The initial model f is trained on the input from the single-object tracking training sets χ_{sample} similar to DiMP [16]. In parallel, to capture crowd group properties, the training split γ from the UUCT dataset is used as input followed by further processing. For training, both T and m are set to 10, sampled sequentially from the input videos, respectively. The unified, pooled feature ∂_T derived from the training pairs represent robust information for both generic object tracking with crowd motion pattern knowledge, later integrated with the test window pairs as:

$$\ell_{t\,\mathrm{arg}\,et} = \frac{1}{\mathbb{N}} \sum_{i=1}^{\mathbb{N}} \sum_{j=1}^{m} \left\| \kappa\left(v_j * \partial_T^i, g_{c_j}\right) \right\|^2 \tag{8}$$

Where, $\mathbb{N}$ is the number of training iterations performed, and the test loss ℓ_{final} captures both foreground and background information discrimination at each iteration with hinge $\kappa\left(v, c\right)$ based on the convolution and gaussian function centered at target $_{c_j}$. The regression is performed as:

$$\kappa\left(v, c\right) = \begin{cases} v - c, & \text{if } c > \varepsilon \\ \max(0, c), & \text{otherwise} \end{cases} \tag{9}$$

Here, ε is a empirical threshold which penalizes positive confidence values c for background prediction $c \leq \varepsilon$. Along with target center prediction, respective bounding box loss ℓ_{bb} is learned according to IoU overlap [15], resulting final loss $\ell_{final} = \eta \ell_{t\,\mathrm{arg}\,et} + \ell_{bb}$. During training, step size η is set to 10 to better training convergence. Moreover, we set the unified feature dimension k to 256, Faster R-CNN [58] as object detector with feature dimension $d = 1024$. We limit maximum number of persons n_t found at a time step t to 50 to reduce computational burdens. Similarly, dm is set to 128 concerning the computational overhead with increased number of frames. Total number of training iterations $\mathbb{N}$ is fixed to 500. The graph learning layer l is 10 during the training and 5 while testing, respectively. The network architecture outlined in Figure 4, developed using PyTorch[4].

2) Online Tracking: We augment the annotated first frame into 10 samples using the data augmentation approach [62] to construct initial χ_{sample}. On the other hand, we extract γ_0 from the first frame and repeat 10 times to initiate the

[4]https://pytorch.org

Algorithm 1 Online HyMP Tracking Algorithm

Input : Initial Trained Model weight ∂_T, Initial Test sample $\chi_{test} = (v_0, c_0)$. New frames v_q where $q > 0$

Output: Estimated new positions with updated model

$\chi_{sample} \leftarrow$ augment $\chi_{test} = (v_0, c_0)$ pair into 10 samples
$\gamma_0 \leftarrow$ apply detector and fetch feature, box pair from χ_{test}
$\gamma \leftarrow$ repeat γ_0 10 times
$\partial_T \leftarrow$ forward pass χ_{sample} and γ to the model ρ and perform 10 training iterations
repeat
  /* bounding box prediction stage */
  $y^{tc} \in \mathbb{R}^2 \leftarrow$ Perform convolution on $\delta(v_q) * \partial_T$
  $y^{bb} \in \mathbb{R}^4, p_q \leftarrow$ Regress candidate boundinng boxes and choose one with the highest confidence score.
  plot y^{bb} in the frame v_q
  /* model update step */
  if $p_q > \varepsilon$ **then**
    /* append frame (+) to memory */
    $\chi_{sample} \leftarrow \chi_{sample} + \delta(v_q)$
    $\gamma_t \leftarrow$ feature, box pair for v_q
    $\gamma \leftarrow \gamma - \gamma_t$
  end
  if $length(\chi_{sample}) > 50$ **then**
    /* remove frame (-) from memory */
    $\chi_{sample} \leftarrow \chi_{sample} - v_0$
    $\gamma \leftarrow \gamma + \gamma_0$
  end
  if 30 *new frames are appended* **then**
    $\partial_T \leftarrow$ forward pass χ_{sample} and γ to the model ρ and perform 1 training iteration
  end
until *Until novel v_q without bounding boxes is received*

tracking. The resulting ∂_T is further leveraged with test samples, and the respective bounding box y^{bb} is predicted. The resultant frame is further used to update ∂_T if the target box is predicted with a sufficient confidence score. For efficiency, we update ∂_T after 30 new training frames are stacked online. To ensure long-term information capture, we set tracker memory up to 50 frames, where a newer candidate will discard the oldest ones. With the parameters mentioned in [15], the bounding boxes are estimated and updated online using their end-to-end proposal generation technique. The pseudo code of online tracking is illustrated in Algorithm 1.

V. Experimental Results

Results on UUCT: We first evaluate existing state-of-the-art correlation filter and deep learning trackers as mentioned in Section II-B, followed by our HyMP in UUCT. Similar to OTB100 [35], we select the one-pass evaluation (OPE) to understand spatial robustness among the trackers. DP is set to 20 for UUCT. In OPE, each evaluated tracker processes over 161.6K frames from 70 sequences and averaged for final result as shown in Figure 5 and Table III (scaled to 100).

According to Figure 5, the best performing correlation filter-based tracker is STRCF [61] with 30.5 precision and

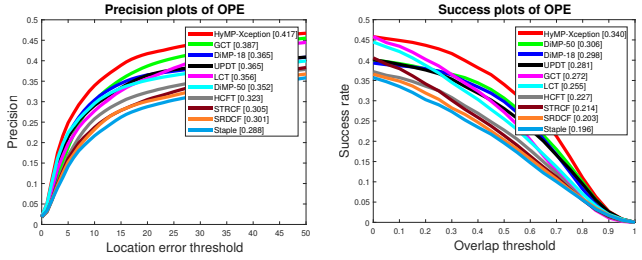
TABLE III
Comparison of algorithms' AUC on the benchmark datasets. For VOT2018, we compared EAO score [36]. HyMP-X is HyMP with Xception [59] as backbone feature extractor

Tracker	VOT-2018	OTB100	UAV123	UAV20L	DTB70	UUCT
Correlation Filter Trackers						
Staple [60]	16.9	58.6	45.0	33.1	35.1	19.6
SRDCF [30]	11.9	59.8	46.4	34.3	36.3	20.3
STRCF [61]	14.3	64.1	48.1	35.4	40.7	20.3
Tracekrs applied Correlation Filter on deeply learned features						
HCFT [38]	19.9	64.7	48.6	36.8	41.5	22.7
LCT [5]	22.1	65.2	49.4	35.9	43.1	25.5
Tracekrs exploited end-to-end deep learning pipeline						
GCT [18]	27.6	64.8	50.8	46.1	44.2	27.2
UPDT [62]	38.3	**70.2**	54.5	49.5	45.7	28.1
DiMP-18 [16]	40.2	66	64.3	51.7	46.9	29.8
DiMP-50 [16]	43.1	68.4	65.4	52.8	47.5	30.6
End-to-end HyMP Tracker (ours)						
HyMP-18	40.1	66.8	66.2	52.5	47.1	31.7
HyMP-50	42.8	69.1	67.6	52.9	48.6	32.9
HyMP-X	**43.6**	69.4	**68.2**	**53.7**	**49.2**	**34.0**

21.5 success scores. Inherent temporal regularization during target translation yields better performance. However, our major focus was on the deep learning trackers due to the data complexity. On recent approaches, DiMP-50, trained using ResNet-50 [50] achieves best performance with 30.6 success area under the curve (AUC). Our HyMP algorithm with similar backbone outperforms DiMP-50 with a large relative gain of 7.5%, showing the impact of crowd information learning and pooling fine-grained features. Using a powerful yet light model, Xception [59], HyMP achieved 34.0 AUC, established a new baseline for benchmarking future experiments on UUCT. Table IV shows attribute-wise evaluation of compared algorithms. The table, along with overall AUC results in Figures 6 and 7, depicts about high complexity of CSDOV and SV due to frequent group reformation. Although HyMP-X outperformed DiMP-50 in both categories, more research is required to improve the overall performance. We refer the readers to **supplementary video**[5] for further demo.

UAV123 and UAV20L [22]: We demonstrate the generalizability of HyMP architecture by evaluating it on aerial SOT benchmark, UAV123, and challenging UAV20L. Our approach with Xception backbone achieves 68.2 AUC on

[5] https://www.youtube.com/watch?v=EdwClc71kq8



(a) Precision (b) Overlap Success

Fig. 5: Overall benchmark of SOTA trackers on UUCT

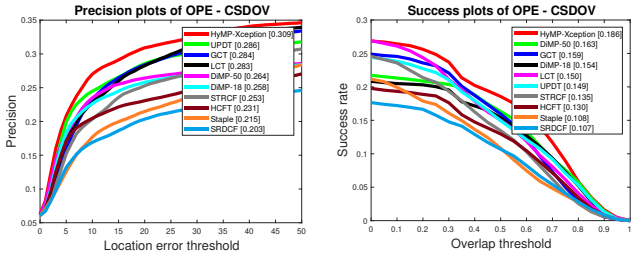
TABLE IV
OVERLAP SUCCESS AUC COMPARISON ON THE UUCT ATTRIBUTES

Model	CSMO	CSDOV	SUMC	MUMC	CSSRM	JCM	SV
HyMP-X	31.4	18.6	32.8	26.6	23.6	22.6	28.7
DiMP-50	28.2	16.3	31.9	22.5	22.1	13.4	22.5

UAV123 and 53.7 on UAV20L, which outperforms existing SOTA algorithms in Table III by maximum 4.3% margin. The better performance interprets HyMP's robustness to overriding human occlusion occurred in several sequences.

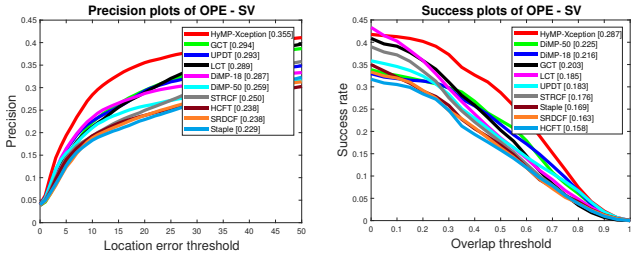
DTB70 [24]: While UAV123 mostly had stabilized UAV tracking sets, similar to UUCT, DTB70 comprises of sequences with motions, cluttered scenes, and objects, making the dataset challenging. We observe that HyMP outperforms closest performing DiMP-50 [16] by gaining 3.6% AUC. The slight improvement can be due to UAV's high altitude and detectable tiny objects by Faster R-CNN.

VOT2018 [36] and OTB100 [35]: To establish our approach more generalizable, we evaluate HyMP on these widely used benchmarks. In VOT2018, HyMP achieves state-of-the-art result by having 43.6 EAO, outperforming DiMP by 1.1%. However, our approach achieves competitive 69.4 AUC, where best model, UPDT [62] scores 70.2 on OTB100.



(a) Precision (b) Overlap Success

Fig. 6: Benchmark on Crowd Split and Dispersed Out of View (CSDOV)



(a) Precision (b) Overlap Success

Fig. 7: Benchmark on Scale Variation (SV) Attribute

VI. CONCLUSION

We introduce the crowd group tracking problem from the UAVs using a single target window. To establish the baseline evaluation of algorithms in this area, we propose a photorealistic benchmark, UUCT. The dataset comprises of 161.6K frames in 70 clips, distributed in 7 attributes. We address current SOT algorithms' limitations on crowd group tracking context due to objective mismatch. To this end, we propose a new approach, HyMP, to learn human-to-human interactions and group reformations by exploiting graph convolutions and low-rank bilinear pooling. We then applied HyMP on the baseline DiMP tracker and outperformed on UUCT and other SOT benchmarks. Our future work will focus on extending UUCT with diverse yet challenging data, better GT generator for localizing crowd groups with lower OE, and designing a lightweight, robust crowd-aware discriminator model.

REFERENCES

[1] X. Li, W. Hu, C. Shen, Z. Zhang, A. Dick, and A. V. D. Hengel, "A survey of appearance models in visual object tracking," *ACM transactions on Intelligent Systems and Technology (TIST)*, vol. 4, no. 4, p. 58, 2013.

[2] A. W. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: An experimental survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 7, pp. 1442–1468, 2013.

[3] M. Kristan, J. Matas, A. Leonardis, T. Vojíř, R. Pflugfelder, G. Fernandez, G. Nebehay, F. Porikli, and L. Čehovin, "A novel performance evaluation methodology for single-target trackers," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 11, 2016.

[4] M. Fiaz, A. Mahmood, S. Javed, and S. K. Jung, "Handcrafted and deep trackers: Recent visual object tracking approaches and trends," *ACM Computing Surveys (CSUR)*, vol. 52, no. 2, p. 43, 2019.

[5] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Adaptive correlation filters with long-term and short-term memory for object tracking," *International Journal of Computer Vision*, vol. 126, no. 8, 2018.

[6] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Discriminative scale space tracking," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 8, pp. 1561–1575, 2016.

[7] S. Hare, S. Golodetz, A. Saffari, V. Vineet, M.-M. Cheng, S. L. Hicks, and P. H. S. Torr, "Struck: Structured output tracking with kernels," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 10, pp. 2096–2109, Oct. 2016.

[8] D. Huang, L. Luo, Z. Chen, M. Wen, and C. Zhang, "Applying detection proposals to visual tracking for scale and aspect ratio adaptability," *International journal of computer vision*, vol. 122, no. 3, pp. 524–541, May 2017.

[9] M. Danelljan, G. Häger, F. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *British Machine Vision Conference, Nottingham, September 1-5, 2014*. BMVA Press, 2014.

[10] M. Danelljan, G. Bhat, F. Shahbaz Khan, and M. Felsberg, "ECO: Efficient convolution operators for tracking," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[11] C. Ma, X. Yang, C. Zhang, and M.-H. Yang, "Long-term correlation tracking," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 5388–5396.

[12] Y. Xiao, J. Li, J. Chang, Y. Zhou, and W. Zhang, "Correlation filter tracking with multiscale spatial view," in *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2018, pp. 1–6.

[13] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, "Fully-convolutional siamese networks for object tracking," in *European conference on computer vision*. Springer, 2016.

[14] X. Chen, X. Zhang, H. Tan, L. Lan, Z. Luo, and X. Huang, "Multi-granularity hierarchical attention siamese network for visual tracking," in *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2018, pp. 1–8.

[15] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "Atom: Accurate tracking by overlap maximization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[16] G. Bhat, M. Danelljan, L. V. Gool, and R. Timofte, "Learning discriminative model prediction for tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[17] C. Liu, P. Zhu, and Q. Hu, "Spatio-temporal active learning for visual tracking," in *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2019, pp. 1–8.

[18] J. Gao, T. Zhang, and C. Xu, "Graph convolutional tracking," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 4649–4659.

[19] J. P. Rodríguez-Gómez, A. G. Eguíluz, J. Martínez-de Dios, and A. Ollero, "Asynchronous event-based clustering and tracking for intrusion monitoring in uas," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 8518–8524.

[20] M. Olivares-Mendez, C. Fu, P. Ludivig, T. Bissyandé, S. Kannan, M. Zurad, A. Annaiyan, H. Voos, and P. Campoy, "Towards an autonomous vision-based unmanned aerial system against wildlife poachers," *Sensors*, vol. 15, no. 12, pp. 31 362–31 391, 2015.

[21] R. Bonatti, Z. Yanfu, S. Choudhury, W. Wang, and S. Scherer, "Autonomous drone cinematographer: Using artistic principles to create smooth, safe, occlusion-free trajectories for aerial filming," in *International Symposium on Experimental Robotics*. Springer, November 2018.

[22] M. Mueller, N. Smith, and B. Ghanem, "A benchmark and simulator for uav tracking," in *European conference on computer vision*. Springer, 2016, pp. 445–461.

[23] D. Du, Y. Qi, H. Yu, Y. Yang, K. Duan, G. Li, W. Zhang, Q. Huang, and Q. Tian, "The unmanned aerial vehicle benchmark: Object detection and tracking," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 370–386.

[24] S. Li and D.-Y. Yeung, "Visual object tracking for unmanned aerial vehicles: A benchmark and new motion models." in *AAAI*, 2017.

[25] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, "Mot16: A benchmark for multi-object tracking," *arXiv preprint arXiv:1603.00831*, 2016.

[26] S. Sharma, J. A. Ansari, J. K. Murthy, and K. M. Krishna, "Beyond pixels: Leveraging geometry and shape cues for online multi-object tracking," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 3508–3515.

[27] J. Portmann, S. Lynen, M. Chli, and R. Siegwart, "People detection and tracking from aerial thermal views," in *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2014.

[28] W. Luo, J. Xing, A. Milan, X. Zhang, W. Liu, and T.-K. Kim, "Multiple object tracking: A literature review," *Artificial Intelligence*, 2020.

[29] M. Danelljan, G. Bhat, F. Shahbaz Khan, and M. Felsberg, "Eco: efficient convolution operators for tracking," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 6638–6646.

[30] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *Proceedings of the IEEE international conference on computer vision*, 2015.

[31] H. Kiani Galoogahi, A. Fagg, and S. Lucey, "Learning background-aware correlation filters for visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[32] M. Mueller, N. Smith, and B. Ghanem, "Context-aware correlation filter tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1396–1404.

[33] D. Zhao and Y. Zeng, "Dynamic fusion of convolutional features based on spatial and temporal attention for visual tracking," in *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2019, pp. 1–8.

[34] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[35] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 2411–2418.

[36] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, L. ˇCehovin Zajc, T. Vojir, G. Bhat, A. Lukezic, A. Eldesokey, *et al.*, "The sixth visual object tracking vot2018 challenge results," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 0–0.

[37] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 2010, pp. 2544–2550.

[38] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Robust visual tracking via hierarchical convolutional features," *IEEE transactions on pattern analysis and machine intelligence*, Aug. 2018.

[39] J. Johnson, R. Krishna, M. Stark, L.-J. Li, D. Shamma, M. Bernstein, and L. Fei-Fei, "Image retrieval using scene graphs," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3668–3678.

[40] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, *et al.*, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," *International journal of computer vision*, vol. 123, no. 1, pp. 32–73, 2017.

[41] J. Yang, J. Lu, S. Lee, D. Batra, and D. Parikh, "Graph r-cnn for scene graph generation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 670–685.

[42] Y. Shen, H. Li, S. Yi, D. Chen, and X. Wang, "Person re-identification with deep similarity-guided graph neural network," in *Proceedings of the European conference on computer vision (ECCV)*, 2018.

[43] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[44] X. Wang and A. Gupta, "Videos as space-time region graphs," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 399–417.

[45] P. Ghosh, Y. Yao, L. Davis, and A. Divakaran, "Stacked spatio-temporal graph convolutional networks for action segmentation," in *The IEEE Winter Conference on Applications of Computer Vision*, 2020, pp. 576–585.

[46] E. Mavroudi, B. B. Haro, and R. Vidal, "Neural message passing on hybrid spatio-temporal visual and symbolic graphs for video understanding," *arXiv preprint arXiv:1905.07385*, 2019.

[47] Y.-H. H. Tsai, S. Divvala, L.-P. Morency, R. Salakhutdinov, and A. Farhadi, "Video relationship reasoning using gated spatio-temporal energy graph," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 424–10 433.

[48] J. Kim, K. W. On, W. Lim, J. Kim, J. Ha, and B. Zhang, "Hadamard product for low-rank bilinear pooling," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

[49] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and service robotics*. Springer, 2018, pp. 621–635.

[50] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[51] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014.

[52] M. Muller, A. Bibi, S. Giancola, S. Alsubaihi, and B. Ghanem, "Trackingnet: A large-scale dataset and benchmark for object tracking in the wild," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 300–317.

[53] H. Fan, L. Lin, F. Yang, P. Chu, G. Deng, S. Yu, H. Bai, Y. Xu, C. Liao, and H. Ling, "Lasot: A high-quality benchmark for large-scale single object tracking," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 5374–5383.

[54] L. Huang, X. Zhao, and K. Huang, "Got-10k: A large high-diversity benchmark for generic object tracking in the wild," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[55] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. IEEE, 2009.

[56] X. Wang and A. Gupta, "Videos as space-time region graphs," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 399–417.

[57] J. Wu, L. Wang, L. Wang, J. Guo, and G. Wu, "Learning actor relation graphs for group activity recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[58] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[59] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.

[60] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. Torr, "Staple: Complementary learners for real-time tracking," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1401–1409.

[61] F. Li, C. Tian, W. Zuo, L. Zhang, and M.-H. Yang, "Learning spatial-temporal regularized correlation filters for visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[62] G. Bhat, J. Johnander, M. Danelljan, F. Shahbaz Khan, and M. Felsberg, "Unveiling the power of deep tracking," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.