TubeTK: Adopting Tubes to Track Multi-Object in a One-Step Training Model

Bo Pang, Yizhuo Li, Yifan Zhang, Muchen Li†, Cewu Lu*
Shanghai Jiao Tong University, †Huazhong University of Science and Technology
{pangbo, liyizhuo, zhangyf_sjtu lucewu}@sjtu.edu.cn, muchenli@alumni.hust.edu.cn

Abstract

Multi-object tracking is a fundamental vision problem that has been studied for a long time. As deep learning brings excellent performances to object detection algorithms, Tracking by Detection (TBD) has become the mainstream tracking framework. Despite the success of TBD, this two-step method is too complicated to train in an end-to-end manner and induces many challenges as well, such as insufficient exploration of video spatial-temporal information, vulnerability when facing object occlusion, and excessive reliance on detection results. To address these challenges, we propose a concise end-to-end model ***TubeTK*** *which only needs one step training by introducing the “bounding-tube” to indicate temporal-spatial locations of objects in a short video clip. TubeTK provides a novel direction of multi-object tracking, and we demonstrate its potential to solve the above challenges without bells and whistles. We analyze the performance of TubeTK on several MOT benchmarks and provide empirical evidence to show that TubeTK has the ability to overcome occlusions to some extent without any ancillary technologies like Re-ID. Compared with other methods that adopt private detection results, our one-stage end-to-end model achieves state-of-the-art performances even if it adopts no ready-made detection results. We hope that the proposed TubeTK model can serve as a simple but strong alternative for video-based MOT task. The code and model will be publicly available accompanying this paper.*

1. Introduction

Video multi-object tracking (MOT) is a fundamental yet challenging task that has been studied for a long time. It requires the algorithm to predict the temporal and spatial location of objects and classify them into correct categories. The current mainstream trackers such as [65, 3, 9, 1, 13] all adopt the tracking-by-detection (TBD) framework. As a two-step method, this framework simplifies the tracking problem into two parts: detecting the spatial location of objects and matching them in the temporal dimension. Although this is a successful framework, it is important to note that TBD method suffers from some drawbacks:

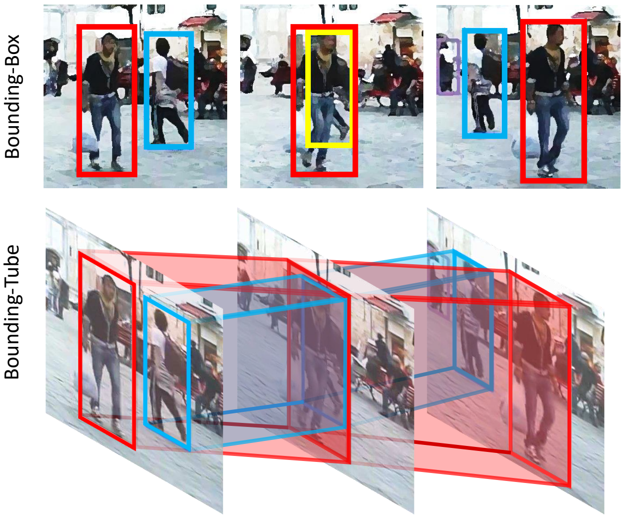


Figure 1. Bounding-boxes and bounding-tubes. As shown in the first row, it is difficult to detect the severely occluded target (the yellow box) by the spatial box without temporal information. In our TubeTK (the second row), it generates bounding-tubes based on temporal-spatial features that encode targets' spatial location and moving trail at the same time. This leads to a one-step training tracking method which is more robust when facing occlusions.

1. As shown in [65, 18], the performances of models adopting TBD framework dramatically vary with detection models. This excessive reliance on image detection results limits performances of the MOT task. Although there are some existing works aiming at integrating the two steps more closely [67, 20, 3], the problems are still not solved fundamentally because of the relatively independent detection model.
2. Due to image-based detection models employed by TBD, the tracking models are weak when facing severe object occlusions (see Fig. 1). It is extremely difficult to detect occluded objects only through spatial representations [3]. The low quality detection further makes tracking unstable, which leads to more compli-

*Cewu Lu is the corresponding author, member of Qing Yuan Research Institute and MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, China.

cated design of matching mechanism [53, 57].

3. As a video level task, MOT requires models to process spatial-temporal information (STI) integrally and effectively. To some extent, the above problems are caused by the separate exploration of STI: detectors mainly model spatial features and trackers capture temporal ones [50, 9, 18, 53], which casts away the semantic consistency of video features and results in incomplete STI at each step.

Nowadays, many video tasks can be solved in a simple one-step end-to-end method such as the I3D model [6] for action recognition [36], TRN [68] for video relational reasoning, and MCNet [56] for video future prediction. As one of the fundamental vision tasks, MOT still does not work in a simple elegant method and the drawbacks of TBD mentioned above require assistance of some other techniques like Re-ID [3, 41]. It is natural to ask a question: *Can we solve the multi-object tracking in a neat one-step framework?* In this way, MOT can be solved as a stand-alone task, without restrictions from detection models. We answer it in the affirmative and for the first time, we demonstrate that the much simpler one-step tracker even achieves better performance than the TBD-based counterparts.

In this paper, we propose the **TubeTK** which conducts the MOT task by regressing the bounding-tubes (Btubes) in a 3D manner. Different from 3D point-cloud [64], this 3D means 2D spatial and 1D temporal dimensions. As shown in Fig. 1, a Btube is defined by 15 points in space-time compared to the traditional 2D box of 4 points. Besides the spatial location of targets, it also captures the temporal position. More importantly, the Btube encodes targets' motion trail as well, which is exactly what MOT needs. Thus, Btubes can well handle spatial-temporal information integrally and largely bridge the gap between detection and tracking.

To predict the Btube that captures spatial-temporal information, we employ a 3D CNN framework. By treating a video as 3D data instead of a group of 2D image frames, it can extract spatial-temporal features simultaneously. This is a more powerful and fully automatic method to extract tracking features, where the handcrafted features such as optical flow [52], segmentation [57, 15, 62], human pose [17, 16, 58] or targets interactions [50, 37, 14, 46] are not needed. The network structure is inspired by recent advances of one-stage anchor-free detectors [55, 11] where the FPN [38] is adopted to better track targets of different scales and the regression head directly generates Btubes. After that, simple IoU-based post-processing is applied to link Btubes and form final tracks. The whole pipeline is made up of fully convolutional networks and we show the potential of this compact model to be a new tracking paradigm.

The proposed TubeTK enjoys the following advantages:

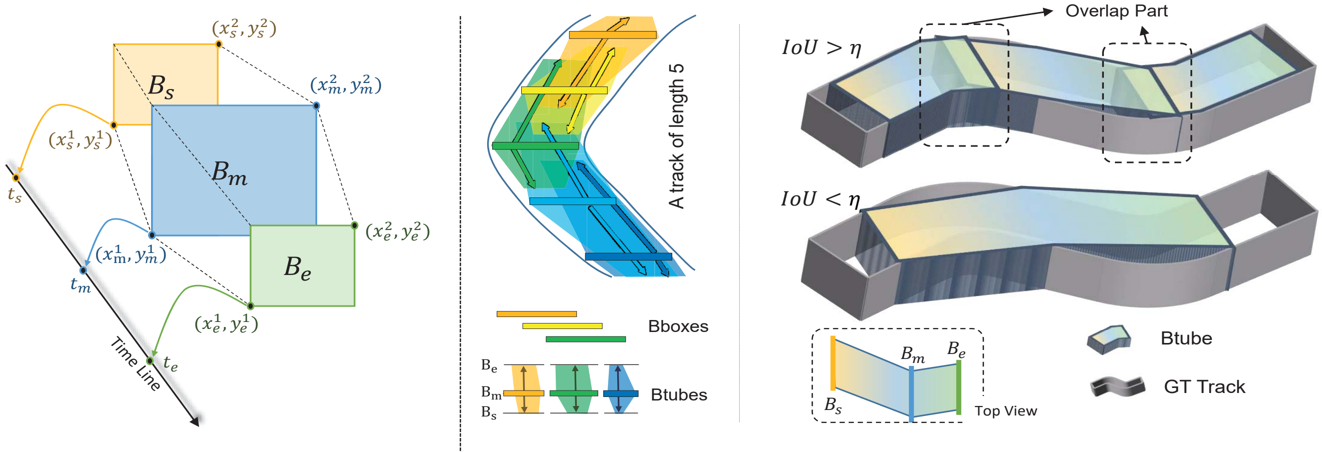
1. With TubeTK, MOT now can be solved by a simple one-step training method as other video tasks. Without constraint from detection models, assisting technologies, and handcrafted features, TubeTK is considerably simpler when being applied and it also enjoys great potential in future research.
2. TubeTK adequately extracts spatial-temporal features simultaneously and these features capture information of motion tendencies. Thus, TubeTK is more robust when faced with occlusions.
3. Without bells and whistles, the end-to-end-trained TubeTK achieves better performances compared with TBD-based methods on MOT15, 16, and 17 dataset [34, 44]. And we show that the Btube-based tracks are smoother (fewer FN and IDS) than the ones based on pre-generated image-level bounding-boxes.

2. Related Work

Tracking-by-detection-based model Research based on the TBD framework often adopts detection results given by external object detectors [47, 40, 42] and focuses on the tracking part to associate the detection boxes across frames. Many associating methods have been utilized on tracking models. In [2, 29, 66, 45, 35], every detected bounding-box is treated as a node of graph, the associating task is equivalent to determining the edges where maximum flow [2, 61], or equivalently, minimum cost [45, 29, 66] are usually adopted as the principles. Recently, with the development of deep learning, appearance-based matching algorithms have been proposed [32, 50, 18]. By matching targets with similar appearances such as clothes and body types, models can associate them over long temporal distances. Re-ID techniques [33, 3, 54] are usually employed as an auxiliary in this matching framework.

Bridging the gap between detection and tracking Performances of image-based object detectors are limited when facing dense crowds and serious occlusions. Thus, some works try to utilize extra information such as motion [50] or temporal features learned by the track step to aid detection. One simple direction is to add bounding-boxes generated by the tracking step into the detection step [41, 10], but this does not affect the original detection process. In [67], the tracking step can efficiently improve the performance of detection by controlling the NMS process. [20] proposes a unified CNN structure to jointly perform detection and tracking tasks. By sharing features and conducting multi-task learning, it can further reduce the isolation between the two steps. The authors of [59] propose a joint detection and embedding framework where the detection and associating steps share same features. Despite these works' effort to bridge the gap between detection and tracking, they still treat them as two separate tasks and can not well utilize spatial-temporal information.

Tracking framework based on trajectories or tubes Tubes can successfully capture motion trails of targets,



a Definition of the Btube

b Generating Btubes from Track

Figure 2. Definition and generation of the Btube. **a**: A Btube can be seen as the combination of three bounding-boxes B_s, B_m, and B_e from different video frames. A Btube has 15 degrees of freedom, which can be determined by the spatial locations of the three bounding-boxes (4×3 degrees) and their temporal positions (3 degrees, t_s, t_m, and t_e). **b**: Btubes are generated from whole tracks. Left: For each bounding-box in a track, we treat it as the B_m of one Btube then look forward and backward to find its B_e and B_s in the track. Right: A longer Btube can capture more temporal features but the IoU between it and the track is lower (η is the IoU threshold), which leads to bad moving trails as the second row shows. Overlaps between the Btubes are used for linking them.

which are important for tracking. There are previous works that adopt tubes to conduct MOT or video detection [51] tasks. In [31, 30], a tubelet proposal module combining detection results into tubes is adopted to solve the video detection task. And [70] employs a single-object tracking method to capture subjects' trajectories. Although these works propose and utilize the concept of tubes, they still utilize external detection results and form tubes at the second step, instead of directly regressing them. Thus they are still TBD methods and the problems stated above are not solved.

3. The Proposed Tracking Model

We propose a new one-step end-to-end training MOT paradigm, the TubeTK. Compared with the TBD framework, this paradigm can better model spatial-temporal features and alleviate problems led by dense crowds and occlusions. In this section, we will introduce the entire pipeline in the following arrangement: 1) We first define the Btube which is a 3D extension of Bbox and introduce its generation method in Sec. 3.1. 2) In Sec. 3.2, we introduce the deep network adopted to predict Btubes from input videos. 3) Next, we interpret the training method tailored for Btubes in Sec. 3.3. 4) Finally, we propose the parameter-free post-processing method to link the predicted Btubes in Sec. 3.4.

3.1. From Bounding-Box to Bounding-Tube

Traditional image-based bounding-box (Bbox) which serves as the smallest enclosing box of a target can only indicate its spatial position, while for MOT, the pattern of targets' temporal positions and moving directions is of equal importance. Thus, we go down to consider how can we extend the bounding-box to simultaneously represent the temporal position and motion, with which, models can overcome occlusions shorter than the receptive field.

Btube definition Adopting a 3D Bbox to point out an object across frames is the simplest extension method, but obviously, this 3D Bbox is too sparse to precisely represent the target's moving trajectory. Inspired by the tubelet in video detection task [31, 30], we design a simplified version, called bounding-tube (Btube), for the dimension of original tubelets is too large to directly regress. A Btube can be uniquely identified in space and time by 15 coordinate values and it is generated by a method similar to the linear spline interpolation which splits a whole track into several overlapping Btubes.

As shown in Fig. 2 **a**, a Btube T is a decahedron composed of 3 Bboxes in different video frames, namely B_s, B_m, and B_e, which need 12 coordinate values to define. And 3 other values are used to point out their temporal positions. This setting allows the target to change its moving direction once in a short time. Moreover, its length-width ratio can change linearly, which makes the Btube more robust when facing pose and scale changes led by perspective. By interpolation between (B_s, B_m) and (B_m, B_e), we can restore all the bounding-boxes $\{B_s, B_{s+1}, ..., B_m, ..., B_{e-1}, B_e\}$ that constitute the Btube. Note that B_m does not have to be exactly at the midpoint of B_s and B_e. It may be closer to one of them. Btubes are designed to encode spatial and temporal information simultaneously. It can even reflect targets' moving trends which are important in MOT task. These specialties make Btubes contain much more useful semantics than traditional Bboxes.

Generating Btubes from tracks Btubes can only capture simple linear trajectories, thus we need to disassemble com-

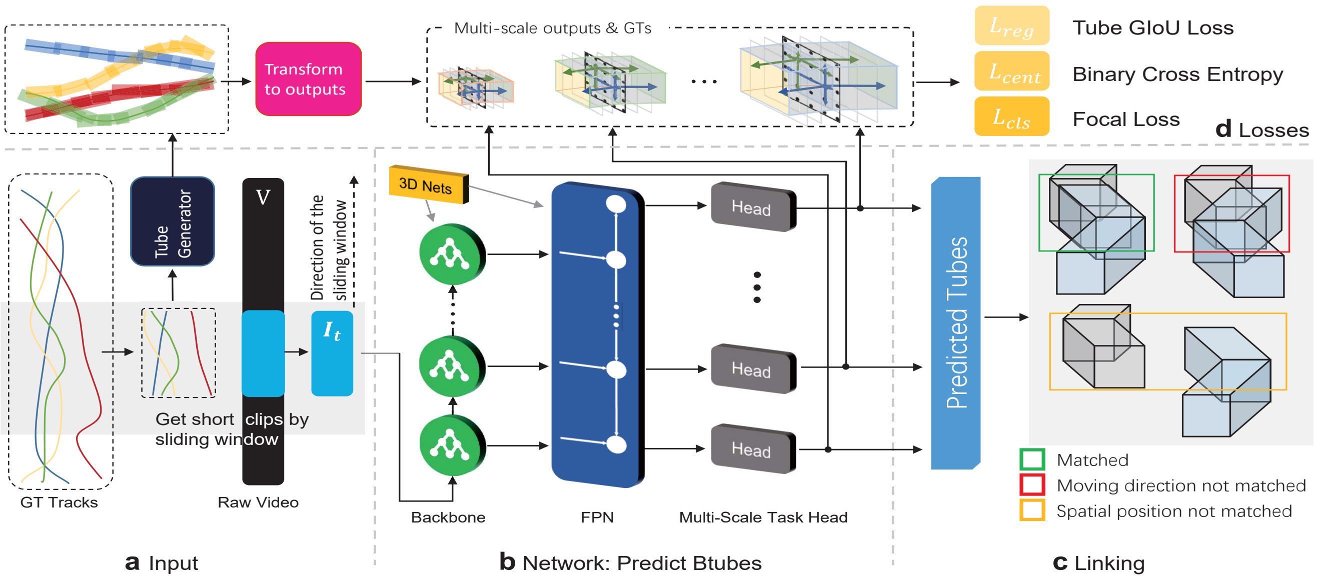


Figure 3. The pipeline of our TubeTK. **a**: Given a video V and the corresponding ground-truth tracks, we cut them into short clips in a sliding window manner to get inputs of the network. **b**: To model spatial-temporal information in video clips, we adopt 3D convolutional layers to build our network which consists of a backbone, an FPN, and a few multi-scale heads. Following FCOS [55], the multi-scale heads are responsible for targets with different scales respectively. The 3D network directly predicts Btubes. **c**: We link the predicted Btubes that have the same spatial positions and moving directions in the overlap part into whole tracks. **d**: In the training phase, the GT tracks are split into Btubes and then they are transformed into the same form of the network's output: target maps (see Fig. 4 for details). The target and predicted maps are fed into three loss functions to train the model: the Focal loss for classifying the foreground and background, BCE for giving out the center-ness, and GIoU loss for regressing Btubes.

plex target's tracks into short clips, in which motions can approximately be seen as linear and captured by our Btubes.

The disassembly process is shown in Fig. 2 **b**. We split a whole track into multiple overlapping Btubes by extending **EVERY** Bbox in it to a Btube. We treat each Bbox as the B_m of one Btube then look forward and backward in the track to find its corresponding B_e and B_s. We can extend Bboxes to longer Btubes for capturing more temporal information, but long Btubes generated by linear interpolation cannot well represent the complex moving trail (see Fig. 2). To balance this trade-off, we set each Btube to be the longest one which satisfies that the mean IoU between its interpolated bounding-boxes B and the ground-truth bounding-boxes $B^\star$ is no less than the threshold η:

$$\begin{aligned} \max \quad & e - s \\ \text{s.t.} \quad & \text{mean}(\{\text{IoU}(B_i, B_i^\star)\}) \geq \eta \\ & i \in \{s, s+1, ..., m, ..., e\} \end{aligned} \tag{1}$$

This principle allows to dynamically generate Btubes with different lengths. When the moving trajectory is monotonous, the Btubes will be longer to capture more temporal information. While when the motion varies sharply, it will generate shorter Btubes to better fit the trail.

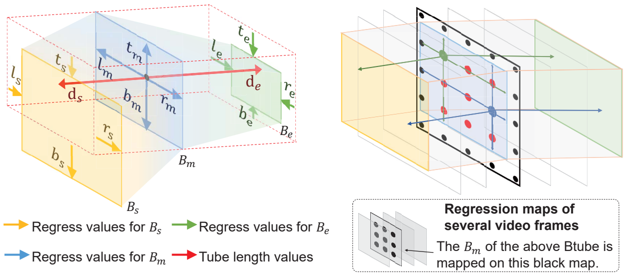
Overcoming the occlusion Btubes guide models to capture moving trends. Thus, when facing occlusions, these trends will assist in predicting the position of shortly invisible targets. Moreover, this specialty can reduce the ID switches at the crossover point of two tracks because two crossing tracks trend to have different moving directions.

3.2. Model Structure

With Btubes that encode the spatial-temporal position, we can handle the MOT task in one step learning without the help of external object detectors or handcrafted matching features. To fit Btubes, we adopt the 3D convolutional structure [28] to capture spatial-temporal features, which is widely used in the video action recognition task [6, 24, 19]. The whole pipeline is shown in Fig. 3.

Network structure The network consists of a backbone, an FPN [38], and a few multi-scale task heads.

Given a video $V \in \mathbb{R}^{T,H,W,C}$ to track, where T, H, W and $C = 3$ are frame number, height, width, and input channel respectively, we split it into short clips I_t as inputs. I_t starts from frame t and its length is l. As Btubes are usually short, the split clips can provide enough temporal information and reduce the computational complexity. Moreover, by adopting a sliding window scheme, the model can work in an online manner. The 3D-ResNet [25, 26] is applied as the backbone to extract the basic spatial-temporal feature groups $\{G^i\}$ with multiple scales. i denotes the level of the features which are generated by stage i of 3D-ResNet. Like the RetinaNet [39] and FCOS [55], a 3D version FPN in which the 2D-CNN layers are simply replaced by 3D-CNNs [28] then takes $\{G^i\}$ as input and outputs multi-



a Output of the network representing the Btube **b** Only the colored points regress that Btube

Figure 4. Regression method and the matchup between output maps and GT Btubes. **a**: The model is required to regress the relative temporal and spatial position to focus on moving patterns. **b**: Each Btube can be regressed by several points in the output map. The colored points on the black map are inside the Btube's B_m, so they are responsible for this Btube. Even through on the grey maps, there are some points also inside the Btube, they do not predict it because they are not on its B_m.

scale feature map groups $\{F^i\}$. This multi-scale setting can better capture targets with different scales. For each F^i, there is a task head composed of several CNN layers to output regressed Btubes and confidence scores. This fully 3D network processes temporal-spatial information simultaneously, making it possible to extract more efficient features.

Outputs Each task head generates three output maps: the confidence map, regression map, and center-ness map following FCOS [55]. The center-ness map is utilized as a weight mask applied on the confidence map in order to reduce confidence scores of off-center boxes. The sizes of these three maps are the same. Each point p in the map can be mapped back to the original input image. If the corresponding point of p in the original input image is inside the B_m of one Btube, then p will regress its position (see Fig. 4). With p the Btube position $\mathbf{r}$ can be regressed by 14 values: four for B_m $\{l_m, t_m, r_m, b_m\}$, four for B_s $\{l_s, t_s, r_s, b_s\}$, four for B_e $\{l_e, t_e, r_e, b_e\}$, and two for the tube length $\{d_s, d_e\}$. Their definitions are shown in Fig. 4. We utilize relative distances with respect to B_m, instead of absolute ones, to regress Btubes aiming to make the model focus on moving trails. The center-ness $\mathbf{c}$ which servers as the weighting coefficient of confidence score $\mathbf{s}$ is defined as:

$$\mathbf{c} = \sqrt{\frac{\min l_m, r_m}{\max l_m, r_m} \times \frac{\min t_m, b_m}{\max t_m, b_m} \times \frac{\min d_s, d_e}{\max d_s, d_e}} \qquad (2)$$

Although $\mathbf{c}$ can be calculated directly from the predicted $\mathbf{r}$, we adopt a head to regress it, and $\mathbf{c}^\star$ calculated based on GT $\mathbf{r}^\star$ by Eq. 2 is utilized as the ground-truth to train the head.

Following the FCOS [55], different task heads are responsible for detecting objects within a range of different sizes respectively, which can largely alleviate the ambiguity caused by one point p falling into multiple Btubes' B_m.

3.3. Training Method

Tube GIoU IoU is the most popular indicator to evaluate the quality of the predicted Bbox, and it is usually used as the loss function. GIoU [49] loss is an extension of IoU loss which solves the problem that there is no supervisory information when the predicted Bbox has no intersection with the ground truth. GIoU of Bbox is defined as:

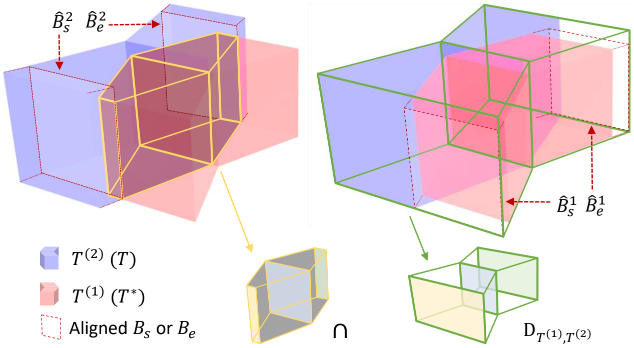


Figure 5. Visualization of the calculation process of Tube GIoU. The intersection and D_{T,T^*} of targets are also decahedrons, thus the volume of them can be calculated in the same way of Btubes.

$$\mathrm{GIoU}(B, B^\star) = \mathrm{IoU}(B, B^\star) - \frac{|D_{B,B^\star} \backslash (B \cup B^\star)|}{|D_{B,B^\star}|} \qquad (3)$$

where $D_{B,B^\star}$ is the smallest enclosing convex object of B and $B^\star$. We extend the definition of GIoU to make it compatible with Btubes. According to our regression method, B_m and $B_m^\star$ must be on the same video frame, which makes the calculation of BTube's volume, intersection $\bigcap$ and smallest tube enclosing object $D_{T,T^\star}$ straightforward. As shown in Fig. 5, we can treat each Btube as two square frustums sharing the same underside. Because B_m and $B_m^\star$ are on the same video frame, $\bigcap$ and $D_{T,T^\star}$ are also composed of two adjoining square frustums whose volumes are easy to calculate (Detail algorithm is shown in supplementary files). Tube GIoU and Tube IoU are the volume extended version of the original area ones.

Loss function For each point p in map M, we denote its confidence score, regression result, and center-ness as $\mathbf{s}_p$, $\mathbf{r}_p$, and $\mathbf{c}_p$. The training loss function can be formulated as:

$$\begin{aligned} L(\{\mathbf{s}_p\}, \{\mathbf{r}_p\}, \{\mathbf{c}_p\}) = & \frac{1}{N_{pos}} \sum_{p \in M} L_{cls}(\mathbf{s}_p, \mathbf{s}_p^\star) \\ & + \frac{\lambda}{N_{pos}} \sum_{p \in M} L_{reg}(\mathbf{r}_p, \mathbf{r}_p^\star) \\ & + \frac{\alpha}{N_{pos}} \sum_{p \in M} L_{cent}(\mathbf{c}_p, \mathbf{c}_p^\star) \end{aligned} \qquad (4)$$

where $\star$ denotes the corresponding ground truth. N_{pos} denotes the number of positive foreground samples. λ and α are the weight coefficients which are assigned as 1 in the experiments. L_{cls} is the focal loss proposed in [39], L_{cent} is the binary cross-entropy loss, and L_{reg} is the Tube GIoU loss which can be formulated as:

$$L_{reg}(\mathbf{r}_p, \mathbf{r}_p^\star) = \mathbb{I}_{\{\mathbf{s}_p^\star = 1\}}(1 - \mathrm{TGIoU}(\mathbf{r}_p, \mathbf{r}_p^\star)) \qquad (5)$$

where $\mathbb{I}_{\{\mathbf{s}_p^\star = 1\}}$ is the indicator function, being 1 if $\mathbf{s}_p^\star = 1$ and 0 otherwise. TGIoU is the Tube GIoU.

3.4. Linking the Bounding-Tubes

After getting predicted Btubes, we only need an IoU-based method without any trainable parameters to link them into whole tracks.

Tube NMS Before the linking principles, we will first introduce the NMS method tailored for Btubes. As Btubes are in 3D space, if we conduct a pure 3D NMS, the huge number of them will lead to large computational overhead. Thus, we simplify the 3D NMS into a modified 2D version. The NMS operation is only conducted among the Btubes whose B_m is on the same video frame. Traditional NMS eliminates targets that have large IoU. However, this method will break at least one track when two or more tracks intersect. Due to the temporal information encoded in Btubes, we can utilize B_s and B_e to perceive the moving direction of targets. Often the directions of intersecting tracks are different, thus the IoU of their B_s, B_m, and B_e will not all be large. In the original NMS algorithm, it will suppress one of two Btubes with IoU larger than the threshold γ, while in the Tube NMS, we set two thresholds γ_1 and γ_2, and for two Btubes $T^{(1)}$ and $T^{(2)}$, suppression is conduct when $\text{IoU}(B_m^{(1)}, B_m^{(2)}) > \gamma_1$ & $\text{IoU}(B_{s'}^{(1)}, B_{s'}^{(2)}) > \gamma_2$ & $\text{IoU}(B_{e'}^{(1)}, B_{e'}^{(2)}) > \gamma_2$, where $s' = \max(s^{(1)}, s^{(2)})$, $e' = \min(e^{(1)}, e^{(2)})$ and $B_{s'}$ is generated by interpolation.

Linking principles After the Tube NMS pre-processing, we need to link all the rest Btubes into whole tracks. The linking method is pretty simple which is only an IoU-based greedy algorithm without any learnable parameters or assisting techniques like appearance matching or Re-ID.

Due to the overlap of Btubes in the temporal dimension, we can focus on it to calculate the frame-based IoU for linking. Given a track $K_{(s_1,e_1)}$ starting from frame s_1 and ending at frame e_1, and a Btube $T_{(s_2,e_2)}$, we first find the overlap part: $O_{(s_3,e_3)}$ where $s_3 = \max(s_1, s_2)$ and $e_3 = \min(e_1, e_2)$. If $s_3 > e_3$, K and T have no overlap and do not need to link. When they are overlapping, we calculate the matching score $\mathcal{M}$ as:

$$\mathcal{M}(K, T) = [\sum_{f \in O} \text{IoU}(K_f, T_f)]/|O| \tag{6}$$

where K_f and T_f denote the (interpolated) bounding-boxes at frame f in K and T. $|O|$ is the number of frames in O. If $\mathcal{M}$ is larger than the linking threshold β, we link them by adding the interpolated bounding-boxes of T onto K. It should be noted that in the overlap part, we average the bounding-boxes from T and K to reduce the deviation caused by the linear interpolation. The linking function can be formulated as:

$$\begin{aligned} K^{new} &= \text{Link}(K_{(s_1,e_1)}, T_{(s_2,e_2)}) \\ &= K_{(s_1,s_3)} + \text{Avg}(K_{(s_3,e_3)}, T_{(s_3,e_3)}) + T_{(e_3,e_2)} \end{aligned} \tag{7}$$

where we assume that $e_1 < e_2$, and $+$ denotes jointing two Btubes (or tracks) without overlap.

To avoid ID switch at intersection of two tracks, we also take moving directions into account. The moving direction vector (MDV) of a Btube (or track) starts from the center of its B_s and ends at B_e's center. We hope the track and Btube with similar directions can be more likely to link. Thus, we compute the angle θ between the MDV of $T_{(s_3,e_3)}$ and $K_{(s_3,e_3)}$ and take $\cos\theta$ as a weighted coefficient masked on $\mathcal{M}$ to adjust the matching score. The final matching score utilized to link is $\mathcal{M}' = \mathcal{M} * (1 + \phi * \cos\theta)$, where $\phi > 0$ is a hyper-parameter. If the direction vectors of the track and Btube form an acute angle, $\cos\theta > 0$ and their matching score $\mathcal{M}'$ will be enlarged, otherwise reduced.

The overall linking method is an online greedy algorithm, which is shown in Alg. 1.

Algorithm 1 Greedy Linking Algorithm

```
Input: Predicted Btubes $\{T_i | i \in \{1, 2, ..., N_T\}\}$
Output: Final tracks $\{K_i | i \in \{1, 2, ..., N_K\}\}$
Grouping $\{T_i\}$ to $\{H_1, H_2, ..., H_L\}$, where $L$ is the total length of the video and
    $H_t = \{T_i^{H_t} | B_m \text{of } T_i^{H_t}$ is at frame $t$ & $i \in \{1, 2, ..., N_T\}\}$.
Utilizing $H_1$ to initialize $\{K_i\}$.
for $t = 2; t \leq L; t++$ do
    Calculating $M'$ between $\{K_i\}$ and $H_t$ to form the matching score matrix $S$, where $S_{i,j} = \mathcal{M}'(K_i, T_j^{H_t})$
    Linking the track-tube pairs starting from the largest $S_{i,j}$ in $S$ by Eq. 7 until all the rest $S_{i,j} < \beta$ . Each linking operation will update $\{K_i\}$.
    The remaining Btubes after linking are added to $\{K_i\}$ as new tracks.
end for
```

4. Experiments

Datasets and evaluation metrics We evaluate our TubeTK model on three MOT Benchmarks [44, 34], namely 2D-MOT2015 (MOT15), MOT16, and MOT17. These benchmarks consist of videos with many occlusions, which makes them really challenging. They are widely used in the field of multi-object tracking and can objectively evaluate models' performances. MOT15 contains 11 train and 11 test videos, while MOT16 and MOT17 contain the same videos, including 7 train and 7 test videos. These three benchmarks provide public detection results (detected by DPM [21], Faster R-CNN [48], and SDP [63]) for fair comparison among TBD frameworks. However, because our TubeTK conducts MOT in one-step, we do not adopt any external detection results. Without detection results generated by sophisticated detection models trained on large datasets, we need more videos to train the 3D network. Thus, we adopt a synthetic dataset JTA [12] which is directly generated from the video game *Grand Theft Auto V* developed by *Rockstar North*. There are 256 video sequences in JTA, enough to pre-train our 3D network. Following the MOT Challenge [44], we adopt the CLEAR MOT metrics [4], and other measures proposed in [60].

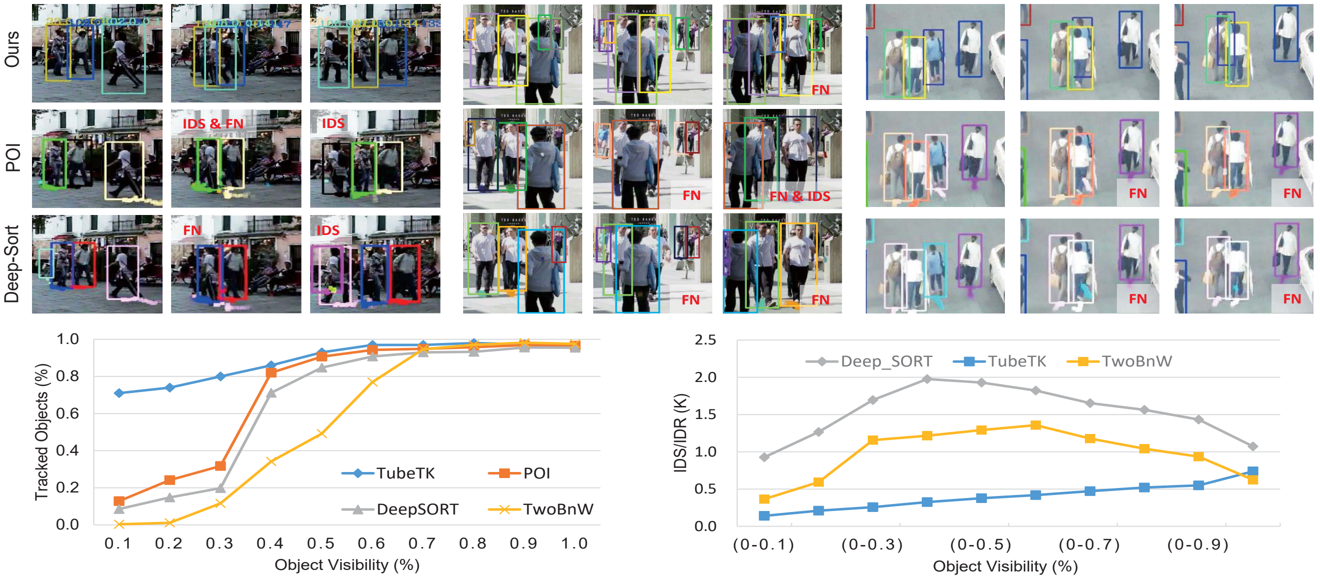


Figure 6. Analysis of the performances in occlusion situations. The examples (from test set of MOT16) in the top row show that our TubeTK can effectively reduce the ID switches and false negatives caused by the occlusion. The bottom analysis is conducted on the training set of MOT-16 dataset. We first illustrate the tracked ratio with respect to visibility. The results reveal that our TubeTK performs much better on highly occluded targets than other models. Then, we illustrate the values of IDS/IDR, the conclusion still holds.

Implementation The hyper-parameters we adopt in the experiments are shown in the following table.

η	l	img size	β	ϕ	γ_1	γ_2
0.8	8	896×1152	0.4	0.2	0.5	0.4

For each clip I_t we randomly sample a spatial crop from it or its horizontal flip, with the per-pixel mean subtracted. HSL jitter is adopted as color augmentation. The details of the network structure follow FCOS [55] (see supplementary file for detail). We only replace the 2D CNN layers with the 3D version and modify the last layer in the task head to output tracking results. We initialize the weights as [55] and train them on JTA from scratch. We utilize SGD with a mini-batch size of 32. The learning rate starts from 10^{-3} and is divided by 5 when error plateaus. TubeTK is trained for 150K iterations on JTA and 25K on benchmarks.The weight decay and momentum factors are 10^{-5} and 0.9.

Ablation study The ablation study is conducted on MOT17 training set (without pre-training on JTA). Tab. 1 demonstrates the great potential of the proposed model. We find that shorter clips ($l = 4$) encoding less temporal information lead to bad performance, which reveals that extending the bounding-box to Btube is effective. Moreover, if we fix the length of all the Btubes to 8 (the length of input clips), the performance drops significantly. Fixing length makes the Btubes deviate from the ground-truth trajectory, leading to much more FNs. This demonstrates that setting the length of Btubes dynamically can better capture the moving trails. The other comparisons show the importance of the Tube GIoU loss and Tube NMS. The Original NMS kills many highly occluded Btubes, causing more FN and IDS, and Tube GIoU loss guides the model to regress the Btube's length more accurately than Tube IoU loss (less FN and FP). TubeTK has much more IDS than Tracktor [3] because our FN is much lower and more tracked results potentially lead to more IDS. From IDF1 we can tell that TubeTK tracks better. Note that we refrain from a cross-validation following [3] as our TubeTK is trained on local clips and never accesses to the tracking ground truth data.

Table 1. Ablation study on the training set of MOT17. D&T and Tracktor adopt public detections generated by Faster R-CNN [23]. POI adopts private detection results and is tested on MOT16.

Model	MOTA↑	IDF1↑	MT↑	ML↓	FP↓	FN↓	IDS↓
D&T [20]	50.1	24.9	23.1	27.1	3561	52481	2715
Tracktor++[3]	61.9	64.7	35.3	21.4	323	42454	326
POI [65]	65.2	-	37.3	14.7	3497	34241	716
TubeTK shorter clips	60.3	60.7	44.3	25.5	3446	40139	968
TubeTK fixed tube len	74.3	68.5	62.5	8.6	7468	19452	1184
TubeTK IoU Loss	70.5	63.7	67.8	6.4	13247	18148	1734
TubeTK original NMS	75.3	70.1	84.6	6.2	11256	13421	2995
TubeTK	76.9	70.0	84.7	3.1	11541	11801	2687

Benchmark evaluation Tab. 2 presents the results of our TubeTK and other state-of-the-art (SOTA) models which adopt public or private external detection results (detailed results are shown in supplementary files). We only compare with the officially published and peer-reviewed online models in the MOT Challenge benchmark*. As we show, although TubeTK does not adopt any external detection results, it achieves new SOTA results on MOT17 (3.0 MOTA

*MOT challenge leaderboard: https://motchallenge.net

Table 2. Results of the online state-of-the-art models on MOT15, 16, 17 datasets. “Detr” denotes the source of the detection results. Our model does not adopt external detection results (w/o). RAN and CNNMTT utilize the ones provided by POI [65].

	Model	Detr	MOTA↑	IDF1↑	MT↑	ML↓	FP↓	FN↓	IDS↓
MOT17	Ours	w/o	**63.0**	58.6	31.2	19.9	27060	177483	4137
	SCNet	Priv	60.0	54.4	**34.4**	**16.2**	72230	**145851**	7611
	LSST17 [22]	Pub	54.7	**62.3**	20.4	40.1	26091	228434	**1243**
	Tracktor [3]	Pub	53.5	52.3	19.5	36.3	**12201**	248047	2072
	JBNOT [27]	Pub	52.6	50.8	19.7	35.8	31572	232659	3050
	FAMNet [9]	Pub	52.0	48.7	19.1	33.4	14138	253616	3072
MOT16	Ours	POI	**66.9**	62.2	39.0	**16.1**	11544	**47502**	1236
	Ours	w/o	64.0	59.4	33.5	19.4	10962	53626	1117
	POI [65]	POI	66.1	65.1	34.0	20.8	5061	55914	805
	CNNMTT [43]	POI	65.2	62.2	32.4	21.3	6578	55896	946
	TAP [69]	Priv	64.8	**73.5**	38.5	21.6	12980	50635	571
	RAN [18]	POI	63.0	63.8	**39.9**	22.1	13663	53248	**482**
	SORT [5]	Priv	59.8	53.8	25.4	22.7	8698	63245	1423
	Tracktor [3]	Pub	54.5	52.5	19.0	36.9	**3280**	79149	682
MOT15	Ours	w/o	**58.4**	53.1	39.3	18.0	5756	18961	854
	RAN [18]	POI	56.5	**61.3**	**45.1**	**14.6**	9386	**16921**	428
	NOMT [8]	Priv	55.5	59.1	39.0	25.8	5594	21322	**427**
	APRCNN [7]	Priv	53.0	52.2	29.1	20.2	**5159**	22984	708
	CDADDAL [1]	Priv	51.3	54.1	36.3	22.2	7110	22271	544
	Tracktor [3]	Pub	44.1	46.7	18.0	26.2	6477	26577	1318

improvements) and MOT15 (1.9 MOTA improvements). On MOT16, it achieves much better performance than other SOTAs that rely on publicly available detections (64.0 vs. 54.5). Moreover, TubeTK performs competitively with the SOTA models adopting POI [65] detection bounding-boxes and appearance features (POI-D-F)[†] on MOT16. It should be noted that the authors of POI-D-F utilize 2 extra tracking datasets, many self-collected surveillance data (10× frames than MOT16) to train the Faster-RCNN detector, and 4 extra Re-ID datasets to extract the appearance features. Thus, we cannot get the same generalization ability as the POI-D-F with synthetic JTA data. To demonstrate the potential of TubeTK, we also provide the results adopting the POI detection (without the appearance features, details in supplementary files) and in this setting our TubeTK achieves the new state-of-the-art on MOT16 (66.9 vs. 66.1). On these three benchmarks, due to the great resistibility to occlusions, our model has fewer FN, under the condition that the number of FP is relatively acceptable. Although TubeTK can handle occlusions better, its IDS is relatively higher because we do not adopt feature matching mechanisms to maintain global consistency. The situation of IDS in occlusion parts is further discussed in Sec. 5.

5. Discussion

Overcoming the occlusion With Btubes, our model can learn and encode the moving trend of targets, leading to more robust performances when facing severe occlusions. We show the qualitative and quantitative analysis in Fig. 6.

Table 3. Experiments on linking robustness. We only test on the GT tracks of a single video MOT17-02. “cn” and “sn” denote the center position and bounding-box scale noises. In each grid, the values are “MOTA” “IDF1”, “MT”, and “ML” in order.

sn cn	0.00	0.05	0.10	0.15	0.20	0.25
0.00	97.2 91.5 59 0	95.2 91.3 58 0	95.0 91.2 58 0	94.3 91.4 58 0	93.6 91.4 54 0	92.4 86.2 53 1
0.05	96.1 91.5 58 0	95.2 90.8 57 0	95.1 91.5 58 0	95.9 91.3 58 0	94.9 91.9 58 0	96.5 89.6 59 0
0.15	94.2 89.1 56 0	94.4 91.8 54 2	95.1 89.4 56 1	96.3 91.3 56 1	94.3 87.4 56 0	94.0 91.3 55 0
0.25	91.6 84.7 54 2	91.1 81.9 55 2	92.8 83.1 54 3	87.8 82.8 54 2	88.5 83.4 53 2	86.4 79.9 54 2

Form the top part of Fig. 6, we show that TubeTK can keep tracking with much less FN or IDS when the target is totally shielded by other targets. In the bottom part, we provide the tracked ratio and number of IDS (regularized by ID recall) with respect to targets' visibility on the training set of MOT16. When the visibility is low, TubeTK performs much better than other TBD models.

Robustness of Btubes for linking The final linking process has no learnable parameters, thus the linking performances depend heavily on the accuracy of regressed Btubes. To verify the robustness, we perform the linking algorithm on GT Btubes with noise jitter. The jitter is conducted on Btubes' center position and spatial-temporal scale. 0.25 jitter on center position or scale means the position or scale shift up to 25% of the Btube's size. The results on MOT17-02, a video with many crossovers, are shown in Tab. 3. We can find that even with large jitter up to 25%, the linking results are still great enough (MOTA $>$ 86, IDF1 $>$ 79), which reveals that the linking algorithm is robust and does not need rigorously accurate Btubes to finish the tracking.

6. Conclusion

In this paper, we proposed an end-to-end one-step training model TubeTK for MOT task. It utilizes Btubes to encode target's temporal-spatial position and local moving trail. This makes the model independent of external detection results and has enormous potential to overcome occlusions. We conducted extensive experiments to evaluate the proposed model. On the mainstream benchmarks, our model achieves the new state-of-the-art performances compared with other online models, even if they adopt private detection results. Comprehensive analyses were presented to further validate the robustness of TubeTK.

7. Acknowledgements

This work is supported in part by the National Key R&D Program of China, No. 2017YFA0700800, National Natural Science Foundation of China under Grants 61772332 and Shanghai Qi Zhi Institute.

[†] https://drive.google.com/open?id=0B5ACiy41McAHMjczS2p0dFg3emM

References

[1] Seung-Hwan Bae and Kuk-Jin Yoon. Confidence-based data association and discriminative deep appearance learning for robust online multi-object tracking. *TPAMI*, 40(3):595–610, 2017. 1, 8

[2] Jerome Berclaz, Francois Fleuret, Engin Turetken, and Pascal Fua. Multiple object tracking using k-shortest paths optimization. *TPAMI*, 33(9):1806–1819, 2011. 2

[3] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixe. Tracking without bells and whistles. *arXiv preprint arXiv:1903.05625*, 2019. 1, 2, 7, 8

[4] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *Journal on Image and Video Processing*, 2008:1, 2008. 6

[5] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *ICIP*, pages 3464–3468. IEEE, 2016. 8

[6] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, pages 6299–6308, 2017. 2, 4

[7] Long Chen, Haizhou Ai, Chong Shang, Zijie Zhuang, and Bo Bai. Online multi-object tracking with convolutional neural networks. In *ICIP*, pages 645–649. IEEE, 2017. 8

[8] Wongun Choi. Near-online multi-target tracking with aggregated local flow descriptor. In *ICCV*, pages 3029–3037, 2015. 8

[9] Peng Chu and Haibin Ling. Famnet: Joint learning of feature, affinity and multi-dimensional assignment for online multiple object tracking. *arXiv preprint arXiv:1904.04989*, 2019. 1, 2, 8

[10] Qi Chu, Wanli Ouyang, Hongsheng Li, Xiaogang Wang, Bin Liu, and Nenghai Yu. Online multi-object tracking using cnn-based single object tracker with spatial-temporal attention mechanism. In *ICCV*, pages 4836–4845, 2017. 2

[11] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Object detection with keypoint triplets. *arXiv preprint arXiv:1904.08189*, 2019. 2

[12] Matteo Fabbri, Fabio Lanzi, Simone Calderara, Andrea Palazzi, Roberto Vezzani, and Rita Cucchiara. Learning to detect and track visible and occluded body joints in a virtual world. In *ECCV*, 2018. 6

[13] Loïc Fagot-Bouquet, Romaric Audigier, Yoann Dhome, and Frédéric Lerasle. Improving multi-frame data association with sparse representations for robust near-online multi-object tracking. In *ECCV*, pages 774–790. Springer, 2016. 1

[14] Hao-Shu Fang, Jinkun Cao, Yu-Wing Tai, and Cewu Lu. Pairwise body-part attention for recognizing human-object interactions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 51–67, 2018. 2

[15] Hao-Shu Fang, Jianhua Sun, Runzhong Wang, Minghao Gou, Yong-Lu Li, and Cewu Lu. Instaboost: Boosting instance segmentation via probability map guided copy-pasting. In *ICCV*, pages 682–691, 2019. 2

[16] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. Rmpe: Regional multi-person pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2334–2343, 2017. 2

[17] Hao-Shu Fang, Yuanlu Xu, Wenguan Wang, Xiaobai Liu, and Song-Chun Zhu. Learning pose grammar to encode human body configuration for 3d pose estimation. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 2

[18] Kuan Fang, Yu Xiang, Xiaocheng Li, and Silvio Savarese. Recurrent autoregressive networks for online multi-object tracking. In *IEEE Winter Conference on Applications of Computer Vision*, pages 466–475. IEEE, 2018. 1, 2, 8

[19] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. *arXiv preprint arXiv:1812.03982*, 2018. 4

[20] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Detect to track and track to detect. In *ICCV*, 2017. 1, 2, 7

[21] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *TPAMI*, 32(9):1627–1645, 2009. 6

[22] Weitao Feng, Zhihao Hu, Wei Wu, Junjie Yan, and Wanli Ouyang. Multi-object tracking with multiple cues and switcher-aware classification. *arXiv preprint arXiv:1901.06129*, 2019. 8

[23] Ross Girshick. Fast r-cnn. In *ICCV*, pages 1440–1448, 2015. 7

[24] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Learning spatio-temporal features with 3d residual networks for action recognition. In *ICCV*, pages 3154–3160, 2017. 4

[25] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *CVPR*, pages 6546–6555, 2018. 4

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 4

[27] Roberto Henschel, Yunzhe Zou, and Bodo Rosenhahn. Multiple people tracking using body and joint detections. In *CVPRW*, pages 0–0, 2019. 8

[28] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *TPAMI*, 35(1):221–231, 2012. 4

[29] Hao Jiang, Sidney Fels, and James J Little. A linear programming approach for multiple object tracking. In *CVPR*, pages 1–8. IEEE, 2007. 2

[30] Kai Kang, Hongsheng Li, Junjie Yan, Xingyu Zeng, Bin Yang, Tong Xiao, Cong Zhang, Zhe Wang, Ruohui Wang, Xiaogang Wang, et al. T-cnn: Tubelets with convolutional neural networks for object detection from videos. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(10):2896–2907, 2017. 3

[31] Kai Kang, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Object detection from video tubelets with convolutional neural networks. In *CVPR*, pages 817–825, 2016. 3

[32] Chanho Kim, Fuxin Li, Arridhana Ciptadi, and James M Rehg. Multiple hypothesis tracking revisited. In *CVPR*, pages 4696–4704, 2015. 2

[33] Cheng-Hao Kuo and Ram Nevatia. How does person identity recognition help multi-person tracking? In *CVPR*, pages 1217–1224. IEEE, 2011. 2

[34] Laura Leal-Taixé, Anton Milan, Ian Reid, Stefan Roth, and Konrad Schindler. Motchallenge 2015: Towards a benchmark for multi-target tracking. *arXiv preprint arXiv:1504.01942*, 2015. 2, 6

[35] Philip Lenz, Andreas Geiger, and Raquel Urtasun. Followme: Efficient online min-cost flow tracking with bounded memory and computation. In *CVPR*, pages 4364–4372, 2015. 2

[36] Yong-Lu Li, Liang Xu, Xijie Huang, Xinpeng Liu, Ze Ma, Mingyang Chen, Shiyi Wang, Hao-Shu Fang, and Cewu Lu. Hake: Human activity knowledge engine. *arXiv preprint arXiv:1904.06539*, 2019. 2

[37] Yong-Lu Li, Siyuan Zhou, Xijie Huang, Liang Xu, Ze Ma, Hao-Shu Fang, Yan-Feng Wang, and Cewu Lu. Transferable interactiveness prior for human-object interaction detection. *CVPR*, 2019. 2

[38] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 2117–2125, 2017. 2, 4

[39] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pages 2980–2988, 2017. 4, 5

[40] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *ECCV*, pages 21–37. Springer, 2016. 2

[41] Chen Long, Ai Haizhou, Zhuang Zijie, and Shang Chong. Real-time multiple people tracking with deeply learned candidate selection and person re-identification. In *ICME*, volume 5, page 8, 2018. 2

[42] Cewu Lu, Hao Su, Yonglu Li, Yongyi Lu, Li Yi, Chi-Keung Tang, and Leonidas J Guibas. Beyond holistic object recognition: Enriching image understanding with part states. In *CVPR*, 2018. 2

[43] Nima Mahmoudi, Seyed Mohammad Ahadi, and Mohammad Rahmati. Multi-target tracking using cnn-based features: Cnnmtt. *Multimedia Tools and Applications*, 78(6):7077–7096, 2019. 8

[44] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. Mot16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*, 2016. 2, 6

[45] Hamed Pirsiavash, Deva Ramanan, and Charless C Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *CVPR*, pages 1201–1208. IEEE, 2011. 2

[46] Siyuan Qi, Wenguan Wang, Baoxiong Jia, Jianbing Shen, and Song-Chun Zhu. Learning human-object interactions by graph parsing neural networks. In *ECCV*, 2018. 2

[47] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, pages 779–788, 2016. 2

[48] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, pages 91–99, 2015. 6

[49] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *CVPR*, pages 658–666, 2019. 5

[50] Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Tracking the untrackable: Learning to track multiple cues with long-term dependencies. In *ICCV*, pages 300–311, 2017. 2

[51] Dian Shao, Yu Xiong, Yue Zhao, Qingqiu Huang, Yu Qiao, and Dahua Lin. Find and focus: Retrieve and localize video events with natural language queries. In *ECCV*, pages 200–216, 2018. 3

[52] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NeurIPS*, pages 568–576, 2014. 2

[53] ShiJie Sun, Naveed Akhtar, HuanSheng Song, Ajmal S Mian, and Mubarak Shah. Deep affinity network for multiple object tracking. *TPAMI*, 2019. 2

[54] Siyu Tang, Mykhaylo Andriluka, Bjoern Andres, and Bernt Schiele. Multiple people tracking by lifted multicut and person re-identification. In *CVPR*, pages 3539–3548, 2017. 2

[55] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. *arXiv preprint arXiv:1904.01355*, 2019. 2, 4, 5, 7

[56] Ruben Villegas, Jimei Yang, Seunghoon Hong, Xunyu Lin, and Honglak Lee. Decomposing motion and content for natural video sequence prediction. *ICLR*, 2017. 2

[57] Paul Voigtlaender, Michael Krause, Aljosa Osep, Jonathon Luiten, Berin Balachandar Gnana Sekar, Andreas Geiger, and Bastian Leibe. Mots: Multi-object tracking and segmentation. In *CVPR*, pages 7942–7951, 2019. 2

[58] Wenguan Wang, Zhijie Zhang, Siyuan Qi, Jianbing Shen, Yanwei Pang, and Ling Shao. Learning compositional neural information fusion for human parsing. In *ICCV*, 2019. 2

[59] Zhongdao Wang, Liang Zheng, Yixuan Liu, and Shengjin Wang. Towards real-time multi-object tracking. In *arXiv preprint arXiv:1909.12605*, 2019. 2

[60] Bo Wu and Ram Nevatia. Tracking of multiple, partially occluded humans based on static body part detection. In *CVPR*, volume 1, pages 951–958. IEEE, 2006. 6

[61] Yuliang Xiu, Jiefeng Li, Haoyu Wang, Yinghong Fang, and Cewu Lu. Pose Flow: Efficient online pose tracking. In *BMVC*, 2018. 2

[62] Wenqiang Xu, Yonglu Li, and Cewu Lu. Srda: Generating instance segmentation annotation via scanning, reasoning and domain adaptation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 120–136, 2018. 2

[63] Fan Yang, Wongun Choi, and Yuanqing Lin. Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers. In *CVPR*, pages 2129–2137, 2016. 6

[64] Yang You, Yujing Lou, Qi Liu, Yu-Wing Tai, Weiming Wang, Lizhuang Ma, and Cewu Lu. Prin: Pointwise rotation-invariant network. *arXiv preprint arXiv:1811.09361*, 2018. 2

[65] Fengwei Yu, Wenbo Li, Quanquan Li, Yu Liu, Xiaohua Shi, and Junjie Yan. Poi: Multiple object tracking with high performance detection and appearance feature. In *ECCV*, pages 36–42. Springer, 2016. 1, 7, 8

[66] Li Zhang, Yuan Li, and Ramakant Nevatia. Global data association for multi-object tracking using network flows. In *CVPR*, pages 1–8. IEEE, 2008. 2

[67] Zheng Zhang, Dazhi Cheng, Xizhou Zhu, Stephen Lin, and Jifeng Dai. Integrated object detection and tracking with tracklet-conditioned detection. *arXiv preprint arXiv:1811.11167*, 2018. 1, 2

[68] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *ECCV*, pages 803–818, 2018. 2

[69] Zongwei Zhou, Junliang Xing, Mengdan Zhang, and Weiming Hu. Online multi-target tracking with tensor-based high-order graph matching. In *ICPR*, pages 1809–1814. IEEE, 2018. 8

[70] Ji Zhu, Hua Yang, Nian Liu, Minyoung Kim, Wenjun Zhang, and Ming-Hsuan Yang. Online multi-object tracking with dual matching attention networks. In *ECCV*, pages 366–382, 2018. 3