

Eye in the Sky: Drone-Based Object Tracking and 3D Localization

Haotian Zhang*

haotiz@uw.edu

University of Washington
Seattle, Washington

Zhichao Lei

zl68@uw.edu

University of Washington
Seattle, Washington

Gaoang Wang

gaoang@uw.edu

University of Washington
Seattle, Washington

Jenq-Neng Hwang

hwang@uw.edu

University of Washington
Seattle, Washington

ABSTRACT

Drones, or general UAVs, equipped with a single camera have been widely deployed to a broad range of applications, such as aerial photography, fast goods delivery and most importantly, surveillance. Despite the great progress achieved in computer vision algorithms, these algorithms are not usually optimized for dealing with images or video sequences acquired by drones, due to various challenges such as occlusion, fast camera motion and pose variation. In this paper, a drone-based multi-object tracking and 3D localization scheme is proposed based on the deep learning based object detection. We first combine a multi-object tracking method called TrackletNet Tracker (TNT) which utilizes temporal and appearance information to track detected objects located on the ground for UAV applications. Then, we are also able to localize the tracked ground objects based on the group plane estimated from the Multi-View Stereo technique. The system deployed on the drone can not only detect and track the objects in a scene, but can also localize their 3D coordinates in meters with respect to the drone camera. The experiments have proved our tracker can reliably handle most of the detected objects captured by drones and achieve favorable 3D localization performance when compared with the state-of-the-art methods.

CCS CONCEPTS

- Security and privacy; • Computing methodologies → Camera calibration; Epipolar geometry; Tracking; Object detection; Neural networks;

KEYWORDS

drone, multi-object tracking, 3D localization, Ground Plane

ACM Reference Format:

Haotian Zhang, Gaoang Wang, Zhichao Lei, and Jenq-Neng Hwang. 2019. Eye in the Sky: Drone-Based Object Tracking and 3D Localization. In *Proceedings of the 27th ACM International Conference on Multimedia (MM '19)*, October 21–25, 2019, Nice, France. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3343031.3350933>

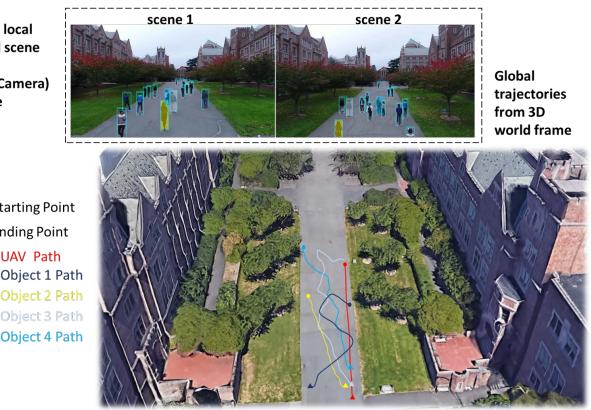


Figure 1: Demonstration of the UAV tracking and localization system. Some local visual scene in UAV (camera) frame and global trajectories in 3D world frame is shown. Each path of the object in the recorded frames begins from the start point towards the endpoint and different colors represent the different object.

'19), October 21–25, 2019, Nice, France. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3343031.3350933>

1 INTRODUCTION

Machine vision systems, such as monocular video cameras, and algorithms represent essential tools for several applications involving the use of unmanned aerial vehicles (UAVs). These techniques are frequently used to extract information about the surrounding scenes for several civilian/military applications like human surveillance, expedition guidance, and 3D mapping. Indeed, UAVs have the potential to dramatically increase the availability and usefulness of an aircraft as information-gathering platforms.

In addition to video cameras, multi-UAV missions have exploited various relative sensing systems for information gathering, such as Radio-Frequency (RF)-based ranging [27], LIDAR-based ranging[17], etc. The main advantages relevant to vision systems are: (1) no additional sensors are needed; (2) visual cameras are extremely small,

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

MM '19, October 21–25, 2019, Nice, France

© 2019 Association for Computing Machinery.

ACM ISBN 978-1-4503-6889-6/19/10...\$15.00

<https://doi.org/10.1145/3343031.3350933>

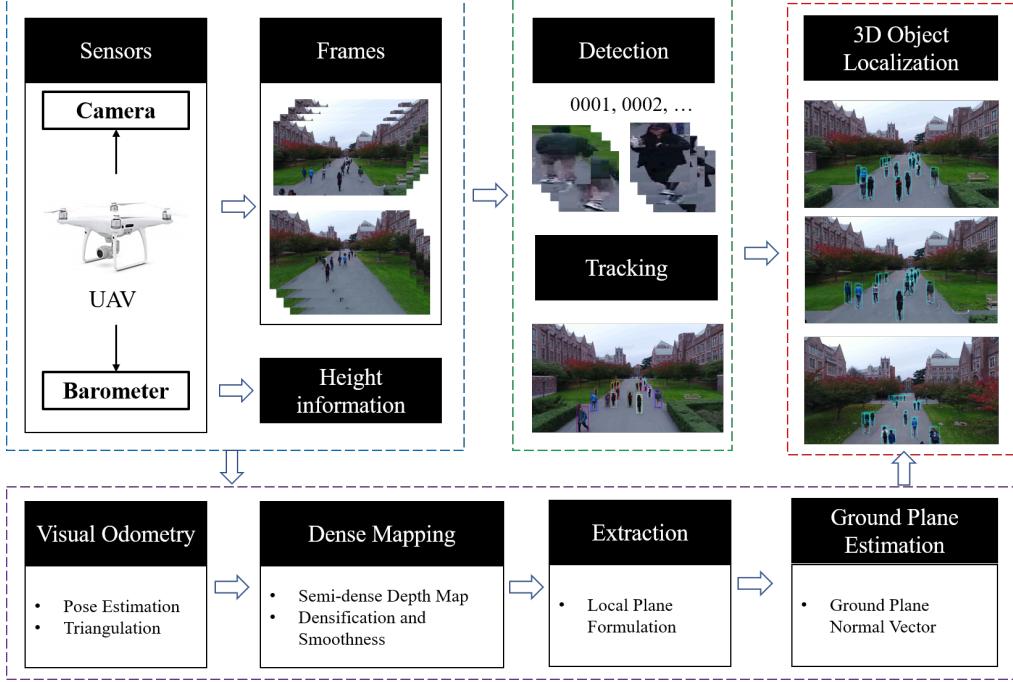


Figure 2: The flow chart of our proposed system, which integrates object detection, multi-object tracking and 3D localization.

light and inexpensive with respect to the other sensors; (3) cameras can provide accurate line-of-sight information, which is often required in specific applications.

The novelties of the proposed system and the advantages are detailed below:

- **Accurate object detection:** The proposed system detects objects of interest based on the modified RetinaNet [16], which provides a better prior for tracking by detection [11] method compared with other state-of-the-art detectors.
- **Multi-object tracking:** A robust TrackletNet Tracker (TNT) for multiple object tracking (MOT), which takes into account both discriminative CNN appearance features and rich temporal information, is incorporated to reduce the impact from unreliable or missing detections and generate smooth and accurate trajectories of moving objects.
- **Visual odometry and ground plane estimation:** We use the effective semi-direct visual odometry (SVO) [9] to get the camera pose between views. The ground plane is then estimated from dense mapping based on the multi-view stereo (MVS) [35] method. It minimizes photometric errors across frames and uses a regularization term to smoothen depth map in low-textured region.
- **3D object localization:** Based on the self-calibrated drone camera parameters, available camera height and estimated ground plane, the detected and tracked objects can be back-projected to 3D world coordinates from 2D image plane. The distance between objects and drones can thus be obtained.

The rest of the paper is organized as follows: Section 2 provides an overview of related works, with a focus on drone-based vision

techniques. Furthermore, the originality and the advantages of each proposed module are motivated and addressed. Sections 3 presents the practical contexts in which every part of the proposed tracking and 3D ground object localization system are developed. Section 4 provides detailed implementation details and extensive experiments results to show the accuracy and robustness of our systems, followed by the conclusions in Section 5.

2 RELATED WORK

The key enabling technologies required in the cognitive task of our proposed drone-based system mainly include object of interest detection, multiple object tracking, detected object 3D localization and overall system integration. In this section, we will present a review of related works on each of modules and open issues ahead.

Object of Interest Detection. Most of surveillance drones fly with low-altitude, so that the ground objects to be detected are within the range of views. Existing vision approaches for object detection are classified into two categories: (1) direct and feature-based methods, and (2) deep learning methods. The latter methods usually achieve higher performance and now become the state-of-the-art techniques. Among which, Faster R-CNN [29], SSD [18], YOLO [28] and RetinaNet [16] are the most popular deep learning detectors used by researchers. However, due to the critical challenges such as fast camera motions, occlusion and relative motion between camera and targets that can cause a significant and high-dynamic variation of sudden appearance changes, the above mentioned deep learning detectors may not be optimal for such scenarios.

Multiple Object Tracking. Most of the recent multi-object tracking (MOT) methods are based on tracking-by-detection schemes[32]. Given detection results, we are able to associate detections across frames and locate objects in 2D even when unreliable detections and occlusions occur. Common tracking frameworks, such as the Graph Model proposed in [22], try to solve the problem by minimizing the total energy loss. However, using graph models for representation requires the nodes (detections) to be conditionally independent, which is usually not the case. Some other frameworks, such as Tracking by Feature Fusion [31, 38, 44], usually jointly fuse classical features (HOG, color histogram and LBP) as appearance features and locations/speed of 2D bounding boxes from detections as temporal features, nonetheless, it is still hard and quite heuristic to determine each weight for feature fusion. Other approaches, like End-to-End deep learning based Tracking [8, 12, 13], can sometimes be successful but require huge amount of labelled training data. It is usually not the case for drones since it is very laborious for human labelling of tiny objects in the drone videos.

Ground Object Localization. There are two definitions for 3D localization in our paper: (1) Self-localized (self-calibrated) of the camera extrinsic parameters to get its own world positions; (2) 3D-localized of detected target objects and get the distance from camera to objects;

To achieve the first goal, simultaneous localization and mapping (SLAM) technology is introduced. ORB-SLAM [23] is a symbolic framework for computing camera trajectory in real-time by extracting and tracking feature points across video frames and reconstruct sparse point cloud using camera geometry. Foster et al. [9] propose a semi-direct monocular visual odometry (SVO), which uses pixel brightness to estimate pose, resulting in the ability to maintain pixel-level based precision in high-frame-rate video, and can generate denser map compared to the ORB-SLAM. To accomplish goal of the second definition, as we all know, it is impossible for an object from a single image to obtain the distance of the object to camera. Knoppe et al. [10] propose a system for a drone carrying a stereo camera to get ground surface scanning data. Karol et al. [20] use a drone-mounted LIDAR to do the ground plane estimation. Nonetheless, the use of additional cameras and advanced sensors could generate problems for a drone such as the increased payload, etc. Thus, the obvious solution is to carry a single camera.

Traditional computer vision techniques also indicate that a ground object can be accurately 3D-localized if the camera pose, its height and ground plane patch beneath the object is known.

Overall System Integration. There are few works have been done for high-level drone-based surveillance systems. Surya et al. [24] propose an autonomous drone surveillance system that can detect individuals engaged in violent activities. Singh et al. [36] use a feature pyramid network (FPN) as the object detector and a ScatterNet Hybrid Deep Learning (SHDL) Networks to estimate the pose of each detected human. However, both works are still very much in its early stage and all their techniques have been demonstrated only based on 2D coordinates. In real world applications, a much better way for a drone to achieve surveillance aim is to infer where are the ground targets (distances) in 3D world space (meters) and how they will move in the future, so that some actions can be

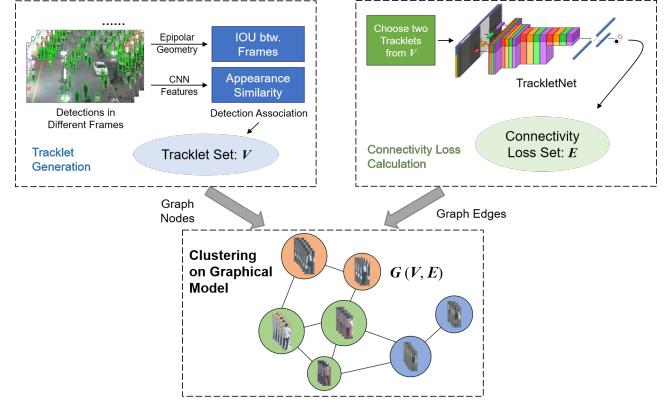


Figure 3: The TNT framework for multi-object tracking. Given the detections in different frames, detection association is computed to generate Tracklets for the Vertex Set V . After that, every two tracklets are put into the TrackletNet to measure the degree of connectivity, which form the similarity on the Edge Set E . A graph model G can be derived from V and E . Finally, the tracklets with the same ID are grouped into one cluster using the graph partition/clustering approach.

predicted according to targets' locations, movements, and speed, etc.

3 PROPOSED TRACKING AND LOCALIZATION SYSTEM

3.1 TrackletNet based MOT Tracker

As shown in Figure 2, our proposed drone-based multiple object tracking (MOT) and 3D localization system only require one single monocular video camera, which can systematically and dynamically calibrate its own extrinsic parameters in order to achieve the self-localization. The ground plane in the view is then estimated by a multi-view stereo[35] method to infer 3D coordinate transformation of the image pixels. Based on the calibrated camera parameters and estimated ground plane, the detected and tracked objects of interests (pedestrians, cars) on the ground can be 3D localized in either the image or the world coordinates.

We adopt TrackletNet Tracker (TNT) [42] in our UAV applications. The tracking system is based on a tracklet graph-based model, as shown in Figure 3, which has three key components, 1) tracklet generation, 2) connectivity measure, and 3) graph-based clustering. Given the detection results in each frame, each tracklets to be treated as a node in the graph is generated based on the intersection-over-union (IOU) compensated by the epipolar geometry constraint due to camera motion and the appearance similarity between two adjacent frames. Between every two tracklets, the connectivity is measured as the edge weight in the graph model, where the connectivity represents the likelihood of the two tracklets being from the same object. To calculate the connectivity, a multi-scale TrackletNet is built as a classifier, which can combine both temporal and spatial features in the likelihood estimation. Clustering [39] is then

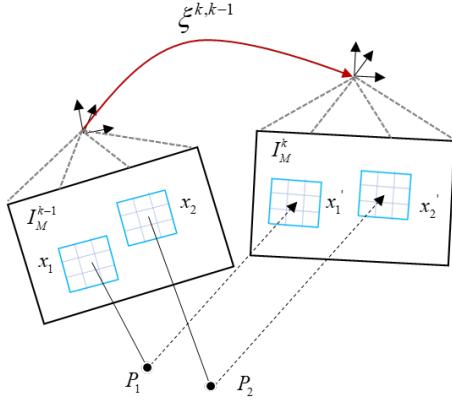


Figure 4: Changing the relative pose $\xi^{k,k-1}$ between the current and the previous frame implicitly moves the position of the reprojected points in the new image. Sparse image alignment seeks to find $\xi^{k,k-1}$ that minimizes the photometric difference between image blocks corresponding to the same 3D point (blue blocks) corresponding to the same 3D point (P_1, P_2).

conducted to minimize the total cost on the graph. After clustering, the tracklets from the same ID can be merged into one group.

The reason we use TNT as our tracking method is due to its robustness dealing with erroneous detections caused by occlusions and missing detections. More specifically, 1) The TrackletNet focuses on the continuity of the embedded features along the time. In other words, the convolution kernels only capture the dependency along time. 2) The network integrates object Re-ID, temporal and spatial dependency as one unified framework. Based on the tracking results from TNT, we know the continuous trajectory of each object ID across frames. This information will be used in the object 3D localization to be discussed in the subsequent subsection.

3.2 Semi-Direct Visual Odometry

To self-calibrate the drone camera, i.e., to estimate the extrinsic camera parameters frame-by-frame, we use a monocular semi-direct visual odometry (SVO) algorithm [9, 34], which directly operates on the raw intensity image instead of using extracted features at any stage of the algorithm. As shown in Figure 4, we represent the image as function $I : \Omega \rightarrow \mathbb{R}$. Similarly, we represent the inverse depth map and inverse depth variance as functions $D : \Omega_D \rightarrow \mathbb{R}^+$ and $V : \Omega_D \rightarrow \mathbb{R}^+$, where Ω_D contains all the pixels which should have a valid depth hypothesis. Note that D and V separately denote the mean and variance of the *inverse depth*, which is assumed as a Gaussian-distributed depth. The depth values of extracted SIFT [19] feature points are initialized with random depth values and large variance for the first frame. Assume the camera moves slowly and in parallel to the image plane, the SVO will quickly converge to a valid map. The pose of a new frame is then estimated using direct image alignment, more specifically, given the current map $\{I_M, D_M, V_M\}$, the relative pose $\xi \in SE(3)$ of a new frame I is obtained by directly

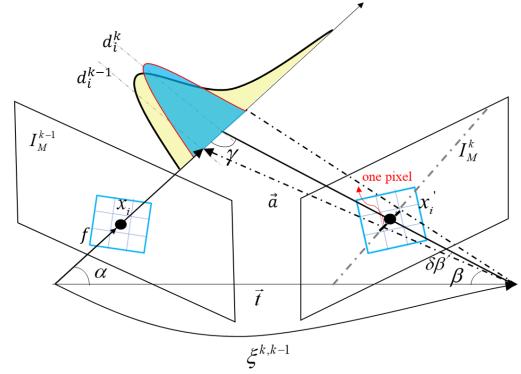


Figure 5: Probabilistic depth estimate d_i^k for feature i in the reference frame I_M^{k-1} . The point at the true depth projects to similar image regions in both images (blue squares). The point of highest correlation lies always on the epipolar line in the new image.

minimizing the photometric error.

$$E(\xi) := \sum_{x \in \Omega_{D_M}} \|I_M(x) - I(\omega(x, D_M(x), \xi))\|_\delta, \quad (1)$$

where $\omega : \Omega_{D_M} \times \mathbb{R} \times SE(3) \rightarrow \Omega$ projects a point from reference image plane to the new frame, and $\|\cdot\|_\delta$ is the Huber norm to account for outliers.

In order to make the approach more robust, we propose to aggregate the photometric cost in a small pixel block centered at the feature pixel and approximate the neighboring pixels as those estimated for the SIFT feature points. The minimization is computed using standard nonlinear least squares algorithms, such as Levenberg–Marquardt (LM).

3.3 Depth Map from Multi-View Stereo

Based on the sparse depth map values estimated from SVO, we further formulate the dense depth calculation as a Gaussian estimation problem [26] so as to estimate the depth map values surrounding the initialized SIFT points based on multiple frames of a monocular video. As discussed in Section 3.2, the relative pose between subsequent frames and the depth at semi-direct feature locations are estimated from SVO. Each observation gives a depth measurement by triangulating from the reference view and the last acquired view. The depth of a pixel block can be continuously updated on the basis of the current observation. Finally, densification and smoothness on the resulting depth map based on multiple observations is achieved.

More specifically, for a set of previous keyframes as well as every subsequent frame with known relative camera pose, a block matching epipolar search is performed to search for the highest correlation. Several metrics to describe the similarities can be introduced to form the block matching problem, such as Sum of Absolute Distance (SAD) [15], Sum of Squared Distance (SSD) [43], and Normalized Cross Correlation (NCC) [19], among which NCC has been commonly used as a metric to evaluate the degree of similarity between two compared pixel blocks. The main advantage of the NCC is that it is less sensitive to linear changes in the amplitude of

illumination in two compared pixel blocks. In our case, the block matching between the block centered at x_i in frame I_M^{k-1} and that of x_i' in frame I_M^k can be given as,

$$S(x_i, x_i') = \frac{\sum_{m,n} x_i(m,n) x_i'(m,n)}{\sqrt{\sum_{m,n} x_i(m,n)^2 x_i'(m,n)^2}} \quad (2)$$

where (m, n) indicates each pixel inside the corresponding block. If the resulting value is close to 1, which means two pixel blocks between two consecutive frames are very likely to be the same. The problem might occur if the epipolar search is long or the block becomes non-textured, we are very likely to encounter a non-convex distribution for correlation score, resulting in a very unreliable and non-smooth depth map. However, we always know that this is a one-to-one problem, therefore the depth filter is thus introduced for further processing.

We model the depth filter based on a Gaussian distribution, which is the depth d (D) (normally distributed around the true depth). Hence, the probability of depth measurement d_i^k for each block i at frame k is modeled as:

$$p(d_i^k) \sim N(d_i^k | \mu_i, \sigma_i^2) \quad (3)$$

where μ_i represents the mean and σ_i^2 represents variance of the performance of Gaussian distribution of depth measurement, whose parameters could be estimated in a maximum likelihood framework using Expectation Maximization. Since each observation gives a depth measurement by triangulating from reference view and the last acquired view, given the consecutively multiple independent observations $\{d^k, \text{for } k = 1, 2, \dots, N\}$, the depth estimation can be continuously refined by Bayesian propagation, i.e.,

$$p(\mu, \sigma^2 | d^1, \dots, d^N) \propto p(\mu, \sigma^2) \prod_k p(d^k | \mu, \sigma^2) \quad (4)$$

where $p(\mu, \sigma^2)$ is our prior on depth. The μ^k and σ^k can be iteratively obtained from relative positions of the camera at frame $k-1$ and k . According to Figure 5, let \vec{t} be the translation component of relative pose ξ and f be the camera focal length, $\|\vec{d}^{k-1}\|$ and $\|\vec{a}\|$ ($\gg f$) are the depth regarding to image frame I_M^{k-1} and I_M^k , which are obtained from triangulation, then:

$$\alpha = \arccos \left(\frac{\vec{d}^{k-1} \cdot \vec{t}}{\|\vec{d}^{k-1}\| \|\vec{t}\|} \right) \quad (5)$$

$$\beta = \arccos \left(\frac{\vec{a} \cdot (-\vec{t})}{\|\vec{a}\| \|\vec{t}\|} \right) \quad (6)$$

Let $\delta\beta$ be the angle spinning for one pixel:

$$\delta\beta = \arctan \frac{1}{f} \quad (7)$$

$$\gamma = \pi - \alpha - (\beta + \delta\beta) \quad (8)$$

Applying the law of sines, we can recover the norm of the updated \vec{d}^k :

$$\|\vec{d}^k\| = \|t\| \frac{\sin(\beta + \delta\beta)}{\sin \gamma} \quad (9)$$

Hence, the μ^k and σ^k can be represented as

$$\begin{aligned} \mu^k &= \frac{1}{2} (\|\vec{d}^{k-1}\| + \|\vec{d}^k\|) \\ \sigma^k &= \|\vec{d}^k\| - \|\vec{d}^{k-1}\| \end{aligned} \quad (10)$$

By using Eq. 4, the estimates of μ^k and σ^k will eventually converge to the correct value and the depth is updated on the basis of the current observation.

For densification, we extend PatchMatch Stereo [4] to Multiview form. We keep the camera poses from SVO and epipolar search for best depth value for each local block. Search and updating for the best value for each block is time-consuming, however, PatchMatch uses a belief propagation to accelerate the updating process. For each block, we look for the depth value with least photometric error and propagate it to the other neighboring pixels using bilinear interpolation.

3.4 3D Object Localization via Ground Plane Estimation

As shown in Figure 6, the camera height above the ground h_{cam} is defined as the distance from the principle center to the ground plane. For a common geometry representation of the ground plane, the ground plane is defined as ground height h_{cam} and the unit normal vector $n = (n_1, n_2, n_3)^T$. There exists a pitch angle θ between the drone and ground plane. For any 3D point $(x, y, z)^T$ on the ground plane, we have $h_{cam} = y \cos \theta - z \sin \theta$.

Assume we obtain the depth map from 4 and there are multiple objects on the ground, we use the multiple average depth values \bar{z} surrounding the bottom center points of each bounding boxes of multiple detected objects to form a local plane. Once such a plane is obtained, we can get the unit normal vector $n = (n_1, n_2, n_3)^T$ by using Cramer's rule [14]:

$$\begin{aligned} n_1 &= \sum y \bar{z} \times \sum xy - \sum x \bar{z} \times \sum yy \\ n_2 &= \sum xy \times \sum x \bar{z} - \sum xx \times \sum y \bar{z} \\ n_3 &= \sum xx \times \sum yy - \sum xy \times \sum xy \end{aligned} \quad (11)$$

3D Object Localization. Accurate estimation of both ground height and orientation is crucial for 3D object localization [37]. Let K be the camera intrinsic calibration matrix. The bottom center of a 2D bounding box, $b = (x, y, 1)^T$ in homogeneous coordinates, can now be back-projected to 3D through the ground plane $\{n^T, h_{cam}\}$.

$$c = \pi_G^{-1}(b) = \frac{h_{cam} K^{-1} b}{n^T K^{-1} b} \quad (12)$$

4 EXPERIMENTS

4.1 Datasets

Two datasets are used to evaluate our performance for each stage.

VisDrone-2018. VisDrone benchmark dataset [45] was proposed at ECCV 2018 workshop. The benchmark datasets consist of 263 video clips with 179,264 frames, captured by various drone-mounted cameras. Objects of interests frequently appear in the image are pedestrians, cars, buses, etc. Tasks involved in this dataset, such as

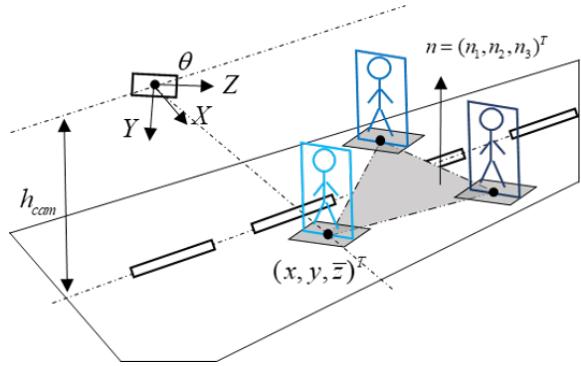


Figure 6: Coordinate system definitions for 3D object localization. The ground plane is defined as a $\{n^T, h_{cam}\}$. \bar{z} is the average depth of the surrounding area.

object detection and multi-object tracking, are extremely challenging due to issues such as occlusion, large scale, and pose variation and fast motions. The dataset is used to evaluate our performance for detection and tracking modules.

Our own-recorded dataset. We chose the commercial UAV DJI Phantom 4 as a platform for the data acquisition. The video frames were captured by the equipped monocular camera, which guarantees high-quality video/image acquisition during speed movement with its wide-angle fixed focal length, and a shooting screen without distortion. The barometer module on the drone is used to measure the flight attitude for the monocular visual odometry scale correction. Our own datasets also cover different environments, including campus, grass land, basketball field, etc. The target objects positions are recorded using hand-hold GPS device. We then human-labeled the positions by refining them into multiple grids. Finally, all trajectories are smoothed and can be regarded as the ground truth.

4.2 Implementation Details

Object Detection. Our trained detector was based on the RetinaNet50 Detector [16, 46]. We changed the anchor size to detect smaller objects. For the same reason, we added a CONV layer in FPN’s P3 and P4, where the higher-level features are added to the lower-level features. We also used the multi-scale training techniques and the Soft-NMS [6] algorithm in post processing. The detector was pretrained on MOT16 [21] and fine-tuned on VisDrone2018-DET datasets. We split the training datasets from VisDrone-2018-DET into 6,000 frames for training and 1,048 frames for testing. We evaluated our detection performance for only pedestrians, cars and buses after 20,000 epochs. The mAP for each class reached 86.2%, 97.8%, 95.5% respectively.

Multi-Object Tracking. Similar to the training of the detector, we also pre-trained the multi-scale TrackletNet on MOT16 datasets, and then fine-tuned the model on VisDrone2018-MOT datasets. The VisDrone2018-MOT contains 56 video sequences for training (24,201 frames in total), and 33 sequences for testing. To generate better tracklets, the IOU_threshold is set to 0.3 due to the drone’s fast camera motions. The time window is set to 64 and batch size is



Figure 7: Tracking results on the test sequences in our recorded campus datasets and the VisDrone-MOT benchmark.

set to 32. The Adam optimizer with an initial learning rate of 1e-3 and is decreased by 10 times for every 2,000 iterations.

Intrinsic Camera. The camera matrix K is assumed to be known for every testing sequence. As we will show in the experiments, an approximation [7] of focal length f .

$$K = \begin{bmatrix} f & 0 & w/2 \\ 0 & f & h/2 \\ 0 & 0 & 1 \end{bmatrix} \text{ with } f = \frac{w}{2} \arctan\left(\frac{90}{180} \frac{\pi}{2}\right) \quad (13)$$

under an image size of $w \times h$, assuming a horizontal field of view of 90 degrees, is sufficiently accurate for drone-equipped cameras.

Ground Plane Estimation. As mentioned above, the camera pose estimation is based on semi-direct VO. The implementation is measured by an average drift in pose of 0.0045 meters per second for an average depth of 1 meter. We also estimated depth using sliding window approach by setting the window interval $N = 30$ frames. The area of ground patch beneath the object is chose to be $(a, \frac{1}{3}b)$, where a is the width of the bounding box and b is the height.

4.3 Experimental Performance

Multi-Object Tracking on VisDrone2018-MOT datasets. We provide our qualitative results on VisDrone2018-MOT benchmark datasets by comparing with other state-of-the art methods, which are shown in Table 1. Note that the benchmark datasets can evaluate performance on one of two different evaluation tasks, donated by without prior detection and with prior detection. As mentioned above, our method is based on tracking-by-detection, so the final performance is evaluated on provided Faster-RCNN detection results. Figure 7 shows some examples of tracking results on both VisDrone dataset and our recorded datasets.

V_IoU [5] is also a tracking by detection method, they assumed that the detections of an object in consecutive frames have an unmistakably high overlap IoU which is commonly the case when sufficiently high frame rates. However, their method is just a simple IoU tracker without incorporating the appearance information. TrackCG [41] proposed a novel approach by aggregating temporal events within target groups and integrating a graph-modeling based stitching procedure to handle the multi-object tracking problems.

Tracker	MOTA \uparrow	IDF1 \uparrow	MT \uparrow	ML \downarrow	FP \downarrow	FN \downarrow	IDsw. \downarrow
V_IoU [5]	40.2	56.1	297	514	11,838	74,027	265
TrackCG [40]	42.6	58.0	323	395	14,722	68,060	779
GOG_EOC [25]	36.9	46.5	205	589	5,445	86,399	754
SCTrack [1]	35.8	45.1	211	550	7,298	85,623	798
Ctrack [41]	30.8	51.9	369	375	36,930	62,819	1,376
FRMOT [29]	33.1	50.8	254	463	21,736	74,953	1,043
GOG [25]	38.4	45.1	244	496	10,179	78,724	1,114
CMOT [2]	31.5	51.3	282	435	26,851	72,382	789
Ours	48.6	58.1	281	478	5,349	76,402	468

Table 1: Tracking performance on the VisDrone2018-MOT test set compared to state-of-the-art. Best in bold, second best in blue.

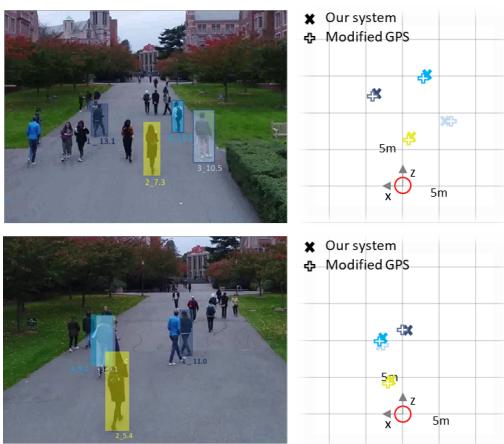


Figure 8: Output of our localization system. The left panel shows input 2D bounding boxes, its object id given by tracking and the estimated distance. The right panel shows the top view of the ground truth object localization from modified GPS results, compared to our 3D object localization given by our system.

Yet, the graphical model is used for representation and requires the nodes (detections) be conditionally independent, which is usually not the case. Our method takes advantage of both appearance feature and temporal information into a unified framework based on an undirected graph model. By comparing the tracking performance, it can be seen that we achieved the first place on MOTA [3, 21], IDF1 [30], and FP (false positive). Among these, IDF1 scores can effectively reflects how long of an object has been correctly tracked and MOT score computes the tracking accuracy. For other metrics like ID Switch, we are also among the top rankings.

3D Localization Performance. The output of our system is shown in Figure 8. The 3D localization performance was evaluated under our captured sequences. As the drone flies to a higher altitude, or the object is farther away, the distance towards the object becomes less accurate. Some examples of 3D localization results are shown in Figure 9.

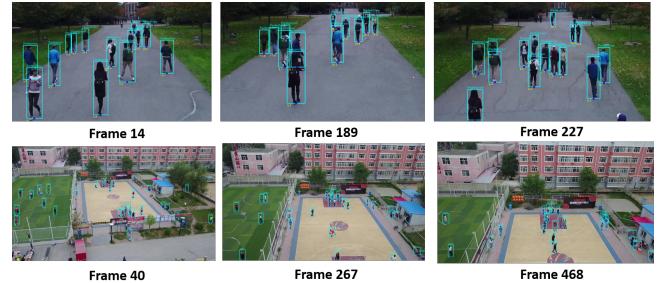


Figure 9: Sampled localization results. The distance between objects and the drone is displayed in yellow and white beneath the bounding boxes (Zoom in for better visualization).



Figure 10: An example showing occlusion handling in testing sequence (basketball field). The trajectory of the person with a purple bounding box (ID: 32) recovers after fully occlusion.

4.4 Ablation Study

Occlusion Handling. Better occlusion handling can help improving the 3D object localization performance. When an object is occluded, the detection is very likely to be unreliable or missing, which can generate wrong 2D bounding boxes or even no bounding boxes at all. The TNT tracker can handle the partial and full occlusions for a long duration. In Figure 10 of the basketball sequence, the person with a purple bounding box is fully occluded by a billboard from frame 12, but its trajectory is recovered after it appears again at frame 40.

Table 2: Mean localization error(standard deviation in parenthesis) in meters.

Approach	Scene	Overall (m)	<=10m	<=25m	>25m
Det+Flat_Ground_Asmo	Campus	3.84(± 1.67)	4.05(± 1.42)	4.76(± 2.06)	N/A
	Grass land	3.96(± 1.74)	2.41(± 1.32)	3.98(± 2.01)	N/A
	Basketball field	6.74(± 3.15)	6.04(± 2.78)	8.66(± 3.18)	12.30(± 3.84)
Det+Our_Ground_Est	Campus	2.22(± 1.12)	2.04(± 0.78)	2.61(± 1.47)	N/A
	Grass land	2.27(± 1.16)	1.15(± 0.77)	1.98(± 1.43)	N/A
	Basketball field	3.21(± 1.84)	2.49(± 1.66)	4.47(± 2.12)	6.71(± 2.33)
Det+Trk+Our_Ground_Est	Campus	0.49(± 0.31)	0.47(± 0.08)	1.21(± 0.54)	N/A
	Grass land	0.78(± 0.31)	0.21(± 0.08)	0.94(± 0.35)	N/A
	Basketball field	2.07(± 1.46)	1.97(± 1.22)	2.42(± 1.74)	3.87(± 1.95)



Figure 11: Typical issues (e.g. view of truncation, incorrect ground plane estimation and motion blur) that affect 3D localization performance.

Ground Plane Estimation and Tracking. To demonstrate the effectiveness of each of our modules, we show the object localization performance with different methods in Table 2. Det+Flat_Ground_Asmo denotes performing detection only and assuming a flat ground plane, i.e., unit normal vector of $[0, -1, 0]^T$. Det+Our_Ground_Est uses our ground plane estimation method in 3.3. Note that the localization performance is especially improved for far objects, since small errors in ground plane can have a large impact on error over longer distances. Finally, in Det+Trk+Our_Ground_Est, the tracking method is added for comparison. In TNT, the unweighted moving average algorithm is applied to adjust the size of the bounding box when unreliable detection occurs. If the detection score is below threshold (0.2), the size of the bounding box is then determined by the past k frames. Let $\{s_{i,t}\}$, where $i \in \{1, 2, 3, 4\}$ be four corner points of the target bounding box in the t -th frame and $\{x_{i,t}\}$ be the detection outputs. The recursive formula of the unweighted moving average is

$$s_{i,t} = s_{i,t-1} + \frac{x_{i,t} - x_{i,t-k}}{k} \quad (14)$$

It is observed that the error decreases further, since the localization can now be estimated on more reliable detection bounding boxes with the help of tracking.

Failure Modes. We illustrate some failure cases in Figure 11, which includes field of view truncation that cause the bottom center of the bounding box no longer being the actual footpoint of the object. Failures can also occur due to incorrect ground plane estimation, and the abrupt camera motion with blurring.

5 CONCLUSION AND FUTURE STUDY

In this work, we have presented a novel framework for drone-based tracking and 3D object localization system. It combines CNN-based object detection, multi-object tracking, ground plane estimation and finally, 3D localization of the ground targets. Both the tracking performance and 3D localization performance are compared with either the state-of-the-art or ground truth. The robustness of our system is shown to handle most of the cases by drone, including occlusion handling and camera fast motions.

However, our work does have a few limitations. Although we demonstrate the fast camera motions may not affect the performance of tracking, it may affect the ground plane estimation. When performing the epipolar search, it is not able to obtain the depth if the camera performs pure rotation, which is usually the case for the drone. A possible solution is to take the monocular depth map by CNN into considerations [33]. Since we are able to get 3D positions of each objects from proposed system, our future work also explores the 3D tracking so the trajectory will be much smoother compared to 2D. By adding some constraints into 3D trajectories, we believe the system will become more robust and effective.

REFERENCES

- [1] Noor M Al-Shakarji, Guna Seetharaman, Filiz Bunyak, and Kannappan Palaniappan. 2017. Robust multi-object tracking with semantic color correlation. In *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 1–7.
- [2] Seung-Hwan Bae and Kuk-Jin Yoon. 2014. Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1218–1225.
- [3] Keni Bernardin and Rainer Stiefelhagen. 2008. Evaluating multiple object tracking performance: the CLEAR MOT metrics. *Journal on Image and Video Processing* 2008 (2008), 1.
- [4] Michael Bleyer, Christoph Rhemann, and Carsten Rother. 2011. PatchMatch Stereo-Stereo Matching with Slanted Support Windows.. In *Bmvc*, Vol. 11. 1–11.
- [5] Eryk Bochinski, Tobias Senst, and Thomas Sikora. 2018. Extending IOU based multi-object tracking by visual information. *AVSS*. IEEE (2018).
- [6] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. 2017. Soft-NMS—Improving Object Detection With One Line of Code. In *Proceedings of the IEEE International Conference on Computer Vision*. 5561–5569.
- [7] Ralf Dragomir and Luc Van Gool. 2014. Ground plane estimation using a hidden markov model. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 4026–4033.
- [8] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. 2017. Detect to track and track to detect. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3038–3046.
- [9] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. 2014. SVO: Fast semi-direct monocular visual odometry. In *2014 IEEE international conference on robotics and automation*. 121–126.

- and automation (ICRA)*. IEEE, 15–22.
- [10] Eija Honkavaara, Heikki Saari, Jere Kaivosoja, Ilkka Pöölönen, Teemu Hakala, Paula Litkey, Jussi Mäkinen, and Liisa Pesonen. 2013. Processing and assessment of spectrometric, stereoscopic imagery collected using a lightweight UAV spectral camera for precision agriculture. *Remote Sensing* 5, 10 (2013), 5006–5039.
 - [11] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas. 2012. Tracking-learning-detection. *IEEE transactions on pattern analysis and machine intelligence* 34, 7 (2012), 1409–1422.
 - [12] Kai Kang, Hongsheng Li, Junjie Yan, Xingyu Zeng, Bin Yang, Tong Xiao, Cong Zhang, Zhe Wang, Ruohui Wang, Xiaogang Wang, et al. 2017. T-cnn: Tubelets with convolutional neural networks for object detection from videos. *IEEE Transactions on Circuits and Systems for Video Technology* (2017).
 - [13] Kai Kang, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. 2016. Object detection from video tubelets with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 817–825.
 - [14] II Kyrchei. 2010. CramerâZs rule for some quaternion matrix equations. *Appl. Math. Comput.* 217, 5 (2010), 2024–2030.
 - [15] Victor Lempitsky and Andrew Zisserman. 2010. Learning to count objects in images. In *Advances in neural information processing systems*. 1324–1332.
 - [16] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*. 2980–2988.
 - [17] Yi Lin, Juha Hyyppä, and Anttoni Jaakkola. 2011. Mini-UAV-borne LIDAR for fine-scale mapping. *IEEE Geoscience and Remote Sensing Letters* 8, 3 (2011), 426–430.
 - [18] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. 2016. Ssd: Single shot multibox detector. In *European conference on computer vision*. Springer, 21–37.
 - [19] David G Lowe. 2004. Distinctive image features from scale-invariant keypoints. *International journal of computer vision* 60, 2 (2004), 91–110.
 - [20] Karol Miadlicki, Mirosław Pajor, and Mateusz Sakow. 2017. Real-time ground filtration method for a loader crane environment monitoring system using sparse LIDAR data. In *2017 IEEE International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*. IEEE, 207–212.
 - [21] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. 2016. MOT16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831* (2016).
 - [22] Anton Milan, Konrad Schindler, and Stefan Roth. 2016. Multi-target tracking by discrete-continuous energy minimization. *IEEE transactions on pattern analysis and machine intelligence* 38, 10 (2016), 2054–2068.
 - [23] Raul Mur-Artal and Juan D Tardós. 2017. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics* 33, 5 (2017), 1255–1262.
 - [24] Surya Pennetra, Fatima Minhuj, Amarjot Singh, and SN Omkar. 2014. Autonomous UAV for suspicious action detection using pictorial human pose estimation and classification. *ELCVIA: electronic letters on computer vision and image analysis* 13, 1 (2014), 18–32.
 - [25] Hamed Pirsiavash, Deva Ramanan, and Charless C Fowlkes. 2011. Globally-optimal greedy algorithms for tracking a variable number of objects. In *CVPR 2011*. IEEE, 1201–1208.
 - [26] Matia Pizzoli, Christian Forster, and Davide Scaramuzza. 2014. REMODE: Probabilistic, monocular dense reconstruction in real time. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2609–2616.
 - [27] G Pupillo, G Naldi, G Bianchi, A Mattana, J Monari, F Perini, M Poloni, M Schiaffino, P Bolli, A Lingua, et al. 2015. Medicina array demonstrator: calibration and radiation pattern characterization using a UAV-mounted radio-frequency source. *Experimental Astronomy* 39, 2 (2015), 405–421.
 - [28] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhad. 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 779–788.
 - [29] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*. 91–99.
 - [30] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. 2016. Performance measures and a data set for multi-target, multi-camera tracking. In *European Conference on Computer Vision*. Springer, 17–35.
 - [31] Ergys Ristani and Carlo Tomasi. 2018. Features for Multi-Target Multi-Camera Tracking and Re-Identification. *arXiv preprint arXiv:1803.10859* (2018).
 - [32] Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. 2017. Tracking the untrackable: Learning to track multiple cues with long-term dependencies. *arXiv preprint arXiv:1701.01909* 4, 5 (2017), 6.
 - [33] Ashutosh Saxena, Sung H Chung, and Andrew Y Ng. 2006. Learning depth from single monocular images. In *Advances in neural information processing systems*. 1161–1168.
 - [34] Thomas Schöps, Jakob Engel, and Daniel Cremers. 2014. Semi-dense visual odometry for AR on a smartphone. In *2014 IEEE international symposium on mixed and augmented reality (ISMAR)*. IEEE, 145–150.
 - [35] Steven M Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. 2006. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, Vol. 1. IEEE, 519–528.
 - [36] Amarjot Singh, Devendra Patil, and SN Omkar. 2018. Eye in the Sky: Real-time Drone Surveillance System (DSS) for Violent Individuals Identification using ScatterNet Hybrid Deep Learning Network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 1629–1637.
 - [37] Shiyu Song and Manmohan Chandraker. 2015. Joint SFM and detection cues for monocular 3D localization in road scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3734–3742.
 - [38] Siyu Tang, Mykhaylo Andriluka, Bjoern Andres, and Bernt Schiele. 2017. Multiple people tracking by lifted multicut and person reidentification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3539–3548.
 - [39] Zheng Tang, Gaogang Wang, Hao Xiao, Aotian Zheng, and Jenq-Neng Hwang. 2018. Single-camera and inter-camera vehicle tracking and 3D speed estimation based on fusion of visual and semantic features. In *CVPR Workshop (CVPRW) on the AI City Challenge*.
 - [40] Wei Tian and Martin Lauer. 2015. Fast cyclist detection by cascaded detector and geometric constraint. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems*. IEEE, 1286–1291.
 - [41] Wei Tian and Martin Lauer. 2017. Joint tracking with event grouping and temporal constraints. In *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 1–5.
 - [42] Gaogang Wang, Yizhou Wang, Haotian Zhang, Renshu Gu, and Jenq-Neng Hwang. 2018. Exploit the Connectivity: Multi-Object Tracking with TrackletNet. *arXiv preprint arXiv:1811.07258* (2018).
 - [43] Kilian Q Weinberger, John Blitzer, and Lawrence K Saul. 2006. Distance metric learning for large margin nearest neighbor classification. In *Advances in neural information processing systems*. 1473–1480.
 - [44] Zhimeng Zhang, Jianan Wu, Xuan Zhang, and Chi Zhang. 2017. Multi-Target, Multi-Camera Tracking by Hierarchical Clustering: Recent Progress on DukeMTMC Project. *arXiv preprint arXiv:1712.09531* (2017).
 - [45] Pengfei Zhu, Longyin Wen, Xiao Bian, Haibin Ling, and Qinghua Hu. 2018. Vision meets drones: a challenge. *arXiv preprint arXiv:1804.07437* (2018).
 - [46] Pengfei Zhu, Longyin Wen, Dawei Du, Xiao Bian, Haibin Ling, Qinghua Hu, Haotian Wu, Qinjin Nie, Hao Cheng, Chenfeng Liu, et al. 2018. VisDrone-VDT2018: The vision meets drone video detection and tracking challenge results. In *European Conference on Computer Vision*. Springer, 496–518.