# Multi-Object Tracking Hierarchically in Visual Data Taken From Drones

Siyang Pan[1], Zhihang Tong[1], Yanyun Zhao[1,2], Zhicheng Zhao[1,2], Fei Su[1,2], Bojin Zhuang[3]
[1]Beijing University of Posts and Telecommunications
[2]Beijing Key Laboratory of Network System and Network Culture, Beijing University of Posts and Telecommunications
[3]Ping An Technology Co.,Ltd
{pansiyang, tongzh, zyy, zhaozc, sufei}@bupt.edu.cn
zhuangbojin232@pingan.com.cn

## Abstract

*Visual understanding tasks on the drone platform have gained considerable attention recently due to the rapid development of drones. In this paper, we present a hierarchical multi-target tracker (HMTT) for visual data taken from drones. Our approach is specifically directed against sequences shot from drone's view with several stages hierarchically performed. The detector detects objects taken from different viewing angles and the detections are filtered to ensure the correctness. Moreover, we propose a method to locate the frames in the case of camera's fast move by two-norm of the homography matrix. Based on that, performance on Multi-Object Tracking is improved with the involvement of Single Object Tracking and a re-identification subnet. Our method participated in the Multi-Object Tracking Challenge (Task 4) of VisDrone2019 benchmark and achieved state-of-the-art performance.*

## 1. Introduction

In the wake of drones (or UAVs) equipped with cameras, analysis on captured sequences continues to receive much attention in recent years. As of today, drones have been widely employed in diversified fields, including agriculture, meteorology, aerial photography and surveillance. Multi-Object Tracking (MOT), aiming to recover the trajectories of objects across frames, plays an important role in autonomous drone systems. However, the vast majority of existing MOT algorithms are hardly optimal for sequences captured by drones as a result of perspective variations, which makes developing an innovation method specifically catering for drones urgent and challenging.

Tracking-by-detection is a widely used pipeline in multiple object tracking thanks to the rapid progress in object detection. Object trajectories are generated by performing data association on detection basis. Some approaches [13, 29, 31, 33] have made excellent progress with the aid of appearance and motion models. However, nearly all the steps of these algorithms are based on the accuracy of detection result, which can be hard to improve due to occlusions and background clutters. In the meantime, unlike surveillance cameras which are fixed, drones capture visual data whilst moving and same kind of objects might exhibit different features in diverse drone views. These issues make it difficult to perform precise estimate on object location as well as high quality of trajectory recovery.

In this paper, we provide a hierarchical multi-target tracker (HMTT) based on detection for drone vision. Hierarchical operating steps significantly enhance the particularity and exactitude. During the stage of determining object position, we first divide each classification into two categories depending upon the shooting angles (right above and inclined top) to get accurate detection kinds. Then, we perform detection result filtering by generating tracklets to remove unreliable bounding boxes, for such detections can seriously reduce the performance of data association in subsequent processing. Based on the reliable detections, we restart the association stage with Single Object Tracking (SOT) and Kalman filtering [12] additionally employed to fill the missed detection gaps. Meanwhile, to maintain stable tracking when the camera suddenly moves, we extract SIFT (Scale Invariant Feature Transform) points [19] to assist data association and single target tracking across consecutive frames. Each trajectory's appearance feature is used to calculate its distance from other ones, which is regarded as the measure of coalescence.

The main contributions of this article are summarized as follows:

- For video data captured by drones, we propose a hierarchical MOT method incorporating SOT and Kalman filtering, with an object re-identification (ReID) net-

work facilitating to polish the result.
- We propose a modified object detection method adapting to different drone's shooting angles, which further filters out unreliable detections by generating tracklets to avoid their adverse impact on follow-up tracking.
- We propose a novel method to determine when the camera moves abruptly by using two-norm of the homography matrix.

## 2. Related works

### 2.1. Drone-based visual data understanding

Though computer vision has been brought closer to drones, the lack of publicly available large-scale drone-based benchmarks or datasets somewhat hinders the further development in drone-based visual data understanding. There are merely a small number of datasets related. With drone platforms, [11] presents a dataset for car counting and [16, 21, 28] respectively collect video sequences for object tracking. Moreover, VisDrone2018 dataset [37] focuses on core problems in computer vision fields and the challenge workshop, Vision Meets Drone Video Object Detection and Tracking (VisDrone-VDT2018) [38], proposed plentiful methods which pushed the boundary of automatic understanding of drone-based visual data.

### 2.2. Object detection

Object detection is one of the fundamental problems in computer vision. Two-stage detectors generate a set of region candidates and classify each using a network. RCNN [10] and Fast-RCNN [9] rely on low level region proposal methods while Faster-RCNN [27] generates region proposal by introducing a region proposal network (RPN). On the other hand, one-stage detectors are able to achieve higher computational efficiency with region proposal generation stage completely dropped. YOLO [26] directly predicts detections with fewer anchor boxes, namely a grid of input image. SSD [18] places anchor boxes densely over feature maps. Recently, other than these anchor-based one-stage approaches, keypoint estimation for object detection gradually ascends the stage. CornerNet [14] employs two bounding box corners as keypoints while ExtremeNet [36] detects four extreme points (top-most, leftmost, bottom-most, right-most) and one center point of objects. CenterNet [35] simply presents per object by a single center point with other properties thereafter regressed from image features at that location. Given that center points are easier to detect, we choose CenterNet for detection in the multi-object tracking challenge due to its high accuracy.

### 2.3. Multi-object Tracking

Plenty of recent MOT methods tend to deal with the task based on the tracking-by-detection paradigm. IOU Tracker [3] relies on no other than intersection-over-union (IOU) of detections and SORT [2] performs Kalman filtering [12] and data association using Hungarian algorithm [22]. Deep SORT [31] combines motion and appearance information to provide greater accuracy in association metric.

On the other hand, tremendous progress has been made in SOT field recently and Siamese networks especially gain considerable attention due to their balanced accuracy and speed. SiamRPN [15] achieves end-to-end representation learning regarding tracking as a one-shot local detection task and DaSiamRPN [39] learns distractor-aware features for explicit distractors suppression. Therefore, SOT trackers have been put into use of MOT tasks. V-IOU Tracker [4] makes improvements on basis of IOU Tracker [3] recurring to visual tracking to continue a track in the absence of detections. Analogously, SAC [6] incorporates a SOT tracker into tracking schemes to cope with missing detections.

However, we find that there is rarely a good way to cope with the small number of false detections as well as camera's fast motion. Therefore we propose a new method to improve MOT performance with detection filtering and frame monitoring added.

## 3. Proposed method

The proposed method, HMTT, consists of four parts: (1) object detection and filtering; (2) frame monitoring; (3) tracking; (4) trajectory connection. Figure 1 shows the framework of our approach and each stage included will be elaborated in this section.

### 3.1. Object detection and filtering

We firstly use a CenterNet [35] network with hourglass [23] backbone to perform object detection. Unfortunately, for same kind of objects under disparate vision angles, features may be extracted with large difference, so it is difficult to obtain detection results robustly, as shown in Figure 2. We observe that drone's shooting angles roughly fall into two categories as RA (right above) and IT (inclined top). Consequently, we separate each object category into two varieties, namely RA and IT, to avoid confusing the network. A low score threshold is set here to reserve as many correct results as possible in spite of FPs (False Positives).

To be specific, we ulteriorly label each bounding box on the train and validation set manually given their vision angles. For instance, a bounding box labeled as a bus would be specifically classified as a RA-bus or an IT-bus. This further manual annotation is easy to finish due to the almost constant shooting angle in a sequence. Also, we deal with the detection results on the test set regarding sequence as the basic unit. Right after obtaining the bounding box results including categories shot from RA and IT, we calculate RORA (the ratio measuring the amount of objects shot

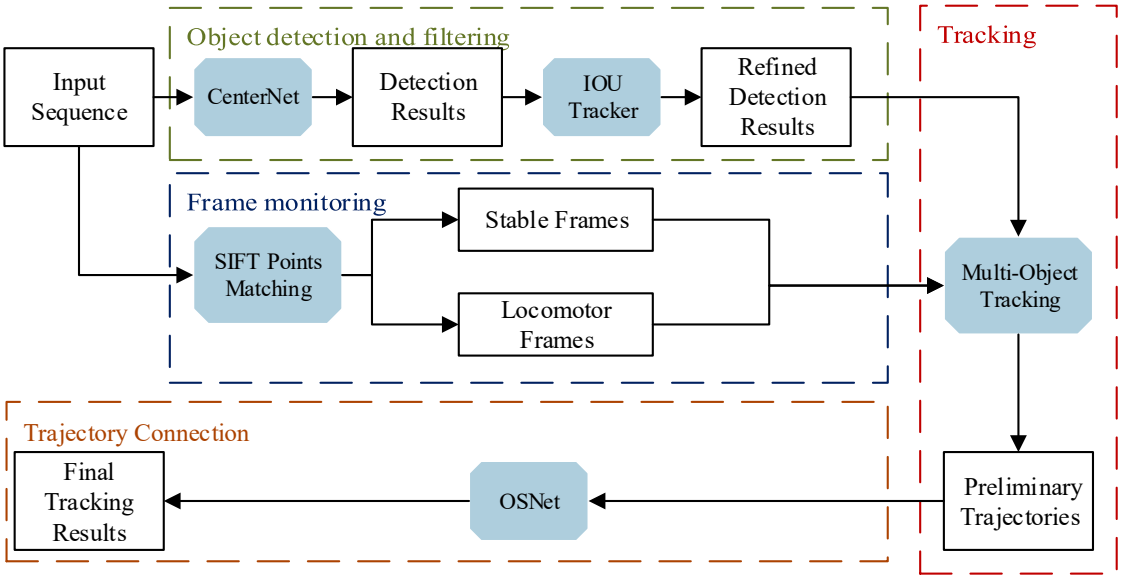


Figure 1. Overview of our method, which processes input sequences hierarchically at the four different stages.

Figure 2. Same kind of objects under disparate vision angles. First row: sample patches shot from IT (inclined top). Second row: sample patches of corresponding categories shot from RA (right above).

from right above) in a sequence by:

$$RORA = \frac{N(RA)}{N(RA) + N(IT)}, \tag{1}$$

where $N(RA)$ means the number of detected objects shot from right above in the sequence and similarly for $N(IT)$. In the light of RORA, we strictly define the sequence's shooting angle as RA-shot if the ratio is rather high or IT-shot if it is extremely low. Sequences whose RORA locates in the middle of the thresholds (mid-shot) will be coped with in a loose way because of their changeable or awkward shooting angle. We simply throw away IT results in RA-shot sequences and discard RA results in IT-shot ones. As for mid-shot sequences, we perform the reintegration to restore the number of categories.

However, false detections are inevitable, so we need to filter out unreliable ones so that true detections can be tracked with no deviations. For the purpose of moving out FPs, the detection results are then fed into a tracker to generate tracklets. Following IOU Tracker [3], we assume true detections of an object in consecutive frames own a high overlap ratio that of IOU, which is calculated by:

$$IOU(a, b) = \frac{Area(a) \bigcap Area(b)}{Area(a) \bigcup Area(b)}. \tag{2}$$

A tracklet extends if the last detection in the previous frame associates with a new detection according to their IOU. The distance matrix $M$ is constructed by IOU distance:

$$m_{i,j} = 1 - IOU(T_i, D_j), \tag{3}$$

where $T_i$ denotes the last bounding box of the $i^{th}$ tracklet and $D_j$ represents the $j^{th}$ detection box, and the distance between them is denoted as $m_{i,j}$. With the aid of Hungarian algorithm [22] and the distance matrix $M$, we find the optimal pairs of tracklets and detections. New tracklets start with detections not assigned to the existing ones and the existing tracklets end without any detection assigned. Then we get rid of the detections neither in any of the tracklets nor with high enough detection score. As shown in Figure 3, nearly all the left detections are reliable.

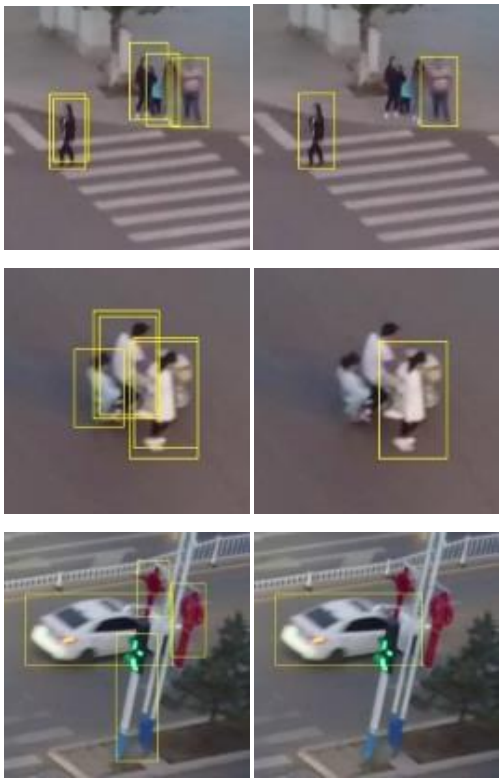

Figure 3. Sample detection filtering outputs using IOU Tracker. Left column: detection results of CenterNet. Right column: refined detection results after filtering.

### 3.2. Frame monitoring

The purpose of frame monitoring is to determine if there is a sudden change in drone motion in order to correct the tracking strategy. Drones generally shoot videos while flying. Slow flights turn out to be innocuous to tracking, but camera's sudden move does not. It has serious effects upon the accuracy of association based on IOU at the following tracking stage. As a result, we perform frame monitoring as pre-processing in order to locate the frames affected by camera's fast motion.

Following [19], we extract SIFT points in every frame of a sequence and attempt to match them across consecutive images by k-NN (k-nearest neighbors algorithm). To be specific, for one key point in a frame, the algorithm is used to find the two nearest points in the consecutive picture. If the ratio of the closest distance to the next closest is less enough, we will retain the closest one and consider the pair as a good match. Hence come a quantity of well-matched points across consecutive frames. By the use of RANSAC [7] algorithm, four pairs are singled out to figure out the homography matrix, which can transform an image from one view to another through perspective transformation calculated by:

$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = H \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix}, \quad (4)$$

where $x_1$, $y_1$ and $x_2$, $y_2$ respectively represent the homogeneous coordinates of corresponding points, and $H$ denotes the homography matrix. To measure the intensity of camera's movements, we use two-norm of the homography matrix as criterion and then pick out the locomotor frames out of stable ones for adaptive operations at the next stage. The two-norm is computed using Equation 5.

$$\|H\|_2 = \sqrt{\lambda_{max}(H^T H)}, \quad (5)$$

where $\lambda_{max}(\cdot)$ represents the maximal eigenvalue of the matrix product. In this way, we convert mutation degree of the multi-dimensional motion of camera into a simple scalar representation, which is more suitable for algorithm discrimination. As shown in Figure 4, SIFT image alignment using Equation 4 performs well in locomotor frames.

### 3.3. Tracking

Detection filtering makes sure the reliability of bounding box results, but it is inevitable to miss some positive ones. So we present an algorithm attempting to fill the gaps while associating the detection results. Based on the frame monitoring stage, we divide the algorithm flow into two branches for stable frames and locomotor ones. Furthermore, Kalman filter [12] and a DaSiamRPN tracker [39] are used for higher association validity and better tracking continuity.

Algorithm 1 shows the tracking procedure of single stable frame and we explain the whole tracking stage in the following steps.

**Step 1.** Initially, the sets of trajectories including $T_a$ of active tracks, $T_t$ of tentative tracks and $T_f$ of finished tracks are all empty. The algorithm begins with the first frame in a sequence.

**Step 2.** For a stable frame, execute the Step 3. For a locomotor one, execute the Step 4 to 5.

**Step 3.** We send $T_a$, $T_t$, $T_f$ and the detection result $D$ of current frame into Algorithm 1 for single frame tracking procedure, where $KF(\cdot)$ means Kalman filtering and $SOT(\cdot)$ denotes the single object tracking with DaSiamRPN.

**Step 4.** Some fine adjustments for Algorithm 1 are performed. We work out the homography matrix between the

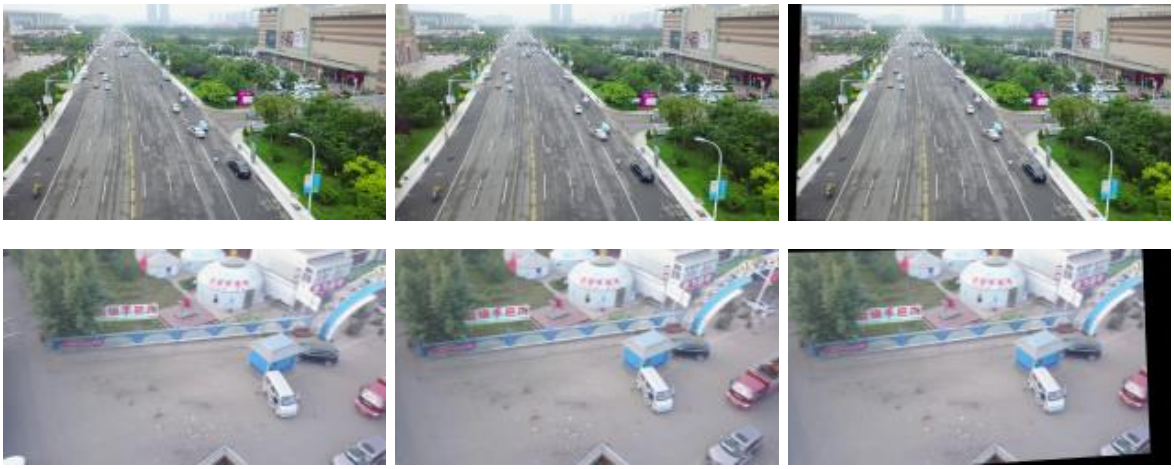

Figure 4. Illustration of cross-frame perspective transformation based on SIFT points matching. The first two columns show the samples of consecutive frames and the last one shows the previous frame after transformation using the homography matrix.

previous frame and the current one, then use it to calculate the transformed locations of bounding boxes in the last frame, which are also the last boxes of tracks in $T_a$ and $T_t$. Then they replace the Kalman filtering predictions in line 2 and 3 to do the Hungarian algorithm. Meanwhile, the SOT result, $b_{SOT}$, is based on the transformed box location and SIFT image alignment.

**Step 5.** In line 16, we no longer perform Kalman filtering and directly determine whether $t_R$ can be sent into $T_f$ according to its length, just following line 28 to 30. Moreover, we get rid of the Kalman filtering operation in line 20 and for the tracks processed in line 20 to 34, we directly bring them to an end casting away the last bounding boxes generated by $KF$ and execute line 28 to 30.

**Step 6.** Go back to Step 2 to process the next frame, unless all the frames in the sequence have been processed. We treat all the tracks in $T_f$ and long enough ones in $T_a$ and $T_t$ (the last bounding boxes of tracks in $T_t$ generated by $KF$ are cast away) as results.

**Step 7.** We extend all the result tracks forward using DaSiamRPN until any of the following condition is satisfied: (1) tracker tracking to the first frame; (2) over half area of the tracking box hanging out of frame; (3) tracking result scoring lower than $\delta_{fSOT}$; (4) tracking result overlapping too much (more than $\delta_{fIOU}$ calculated by IOU) with some certain bounding box in another track; (5) IOU of tracking results between consecutive frames lower than $\delta_{adIOU}$.

For algorithm 1, note that in line 2 and 3, we take advantage of the predictions from line 1 to do the Hungarian algorithm where distance matrix $M$ constructed by IOU distance serves as the criterion to find optimal pairs of tracks and detections, exactly as the detection filtering stage. The first pair enclosed by brackets denotes the successfully assigned detection and track while the following two mean the left ones. In addition, $score(\cdot)$ in line 13 is calculated by the output score of SOT tracker. This algorithm aims to associate highly correct detections while filling gaps with SOT results and use Kalman filtering to mitigate occlusions.

We change the algorithm for locomotor frames because Kalman filtering is no longer valid in this case. Also, both IOU distance and the SOT tracker are based on location information which becomes trustless. So all the moves about Kalman filtering are deleted or replaced. We specifically apply the image alignment for association and SOT to reduce the impact of camera's rapid movements.

Also, as each of the tracks begins with a specific detection, we compensate for its missing front part with SOT in Step 7.

### 3.4. Trajectory connection

Notwithstanding that Kalman filtering alleviates the problems brought by occlusions, probability mass still spreads out in state space in the case of long term object disappearance. In addition, both irregular movements and initial entries have an impact on Kalman filtering forecast. So we introduce a ReID network to get same object's trajectories merged. OSNet [34] is applied here because of its capacity of learning omni-scale feature representations.

The ReID features extracted with OSNet represent bounding boxes rather than trajectories. Thus we propose an exhaustive algorithm that feature distances of every pair of bounding boxes from two trajectories are calculated and the minimum one determines whether they need to be merged. Specifically, we use Euclidean distance of features

**Algorithm 1:** Tracking procedure of single stable frame

**Input:**
The sets of trajectories: $T_a$ of active tracks, $T_t$ of tentative tracks and $T_f$ of finished tracks;
The set of bounding boxes $D$ detected in current frame;
**Output:**
The sets of updated tracks $T_a, T_t, T_f$;

All the tracks in $T_a$ and $T_t$ get predictions with the help of Kalman filtering;
$[D_H, T_H]$, $D_R, T_R$ = Hungarian($D, T_a$);
$[D'_H, T'_H]$, $D_L, T_L$ = Hungarian($D_R, T_t$);
$T_a, T_t, T_f = \phi$;
**for** each $[d_H, t_H]$ in $[D_H, T_H] \bigcup [D'_H, T'_H]$ **do**
  $t_H \leftarrow d_H, T_a \leftarrow t_H$;
**end**
**for** each $t_R$ in $T_R$ **do**
  $b_{SOT}$ = SOT($t_R$);
  $d_{best} = d_j$ where $max(IOU(d_j, b_{SOT}))$, $d_j \in D_L$;
  **if** $IOU(d_{best}, b_{SOT}) \geqslant \delta_{IOU}$ **then**
    $t_R \leftarrow d_{best}$, remove $d_{best}$ from $D_L$, $T_a \leftarrow t_R$;
  **else if** $score(b_{SOT}) \geqslant \delta_{SOT}$ **then**
    $t_R \leftarrow b_{SOT}, T_a \leftarrow t_R$;
  **else**
    $t_R \leftarrow KF(t_R), T_t \leftarrow t_R$;
  **end**
**end**
**for** each $t_L$ in $T_L$ **do**
  $b_{kf} = KF(t_L)$;
  $d_{best} = d_j$ where $max(IOU(d_j, b_{kf}))$, $d_j \in D_L$;
  **if** $IOU(d_{best}, b_{kf}) \geqslant \delta_{IOU}$ **then**
    $t_L \leftarrow d_{best}$, remove $d_{best}$ from $D_L$, $T_a \leftarrow t_L$;
  **else**
    $t_L \leftarrow b_{kf}$;
    **if** all the last 20 tracking boxes of $t_L$ are generated by $KF$ **then**
      remove the 20 boxes from $t_L$;
      **if** $len(t_L) \geqslant L_{min}$ **then**
        $T_f \leftarrow t_L$;
      **end**
    **else**
      $T_t \leftarrow t_L$;
    **end**
  **end**
**end**
**for** each $d_j$ in $D_L$ **do**
  start a new track $t$ with $d_j$;
  $T_a \leftarrow t$;
**end**

as measurement so the distance between two trajectories is defined as:

$$d(x, y) = \min Eu(x_i, y_j), \tag{6}$$

where $x_i$ is the $i^{th}$ bounding box of one trajectory $x$ and $y_j$ is the $j^{th}$ bounding box of the other trajectory $y$. $Eu(\cdot, \cdot)$ denotes the Euclidean distance of the two boxes' ReID features.

For the trajectories which need to be merged, we connect them with linear interpolation if they are close enough on the timeline. Otherwise we simply label them with a same identity.

# 4. Experiment

In this section, we first introduce the benchmark dataset and experiment details. Then the evaluation of the workshop challenge is introduced with our results presented.

## 4.1. Dataset

The VisDrone2019 dataset comprises 288 video clips formed by 261,908 frames and 10,209 static images. These frames are manually annotated with more than 2.6 million bounding boxes of targets of frequent interests. To evaluate our algorithm, we use the Multi-Object Tracking Challenge (Task 4) dataset which provides 56 video sequences for training, 7 sequences for validation and 16 sequences for workshop competition testing. Ten object categories of interest including pedestrian, person, car, van, bus, truck, motor, bicycle, awning-tricycle, and tricycle are annotated while the competition solely considers five object categories in multi-object tracking evaluation, *i.e.*, car, bus, truck, pedestrian, and van. Some rarely occurring special vehicles labeled as others are ignored.

## 4.2. Implementation details

**Object detection and filtering.** Though people is excluded from object categories of interest, it is called back in our work due to its high similarity with pedestrian. We finetune the CenterNet [35] detector pretrained on MS COCO [17] dataset and firstly set the detection score threshold as 0.3. The RORA thresholds for differentiating shooting angles of sequences are as strict as 0.1 and 0.9. No additional processing is required for people and pedestrian results due to their close distance between each other in location under some circumstances. However, we need to deal with vehicle results to eliminate some redundant boxes. NMS (non-maximum suppression) is used for RA-shot sequences while we process vehicle results in IT-shot and mid-shot sequences according to NIOU (Nest-IOU), defined as follows:

$$NIOU(a, b) = \frac{Area(a) \bigcap Area(b)}{min(Area(a), Area(b))}. \tag{7}$$

If the NIOU of two detection boxes exceeds 0.7, we merely remain the enclosing rectangle of them. It successfully

avoids the situation where some part of an object is simultaneously considered as a new whole one. With multi-scale evaluation, the performance on small objects gets promoted. In the filtering phase, all sequences are processed with the IOU distance threshold of 0.5 while using the Hungarian algorithm. Tracklets in RA-shot sequences and IT-shot sequences are generated strictly with every detection scoring over 0.45 and the maximum should be higher than 0.5. Every tracklet should last 6 frames at least. Tracklets in mid-shot sequences are eased up with every detection scoring over 0.35 and the maximum over 0.4. The length requirement has also been lowered to 2 frames. Moreover, tracklets containing more than two detections classified as people are entirely thrown away and we pick up back detections scoring over 0.68 even if they have been dropped during tracklets generation.

**Frame monitoring.** We screen out locomotor frames of a sequence after getting the homography matrix set. The mean and standard deviation of the matrices' two-norms quantifies the overall movement of camera and we define the threshold using Equation 8:

$$thresh = \mu + 2\sigma, \tag{8}$$

where $\mu$ and $\sigma$ indicate the mean and the standard deviation. Frames with homography matrix whose two-norm exceeds $thresh$ are defined as locomotor frames and the other frames are deemed as stable ones.

**Tracking.** A number of parameters need to be set at this stage. Starting with the IOU distance threshold for Hungarian algorithm, we set it loosely as 0.8 with confidence in the filtered detection results. We pick up back unassigned detections with IOU threshold $\delta_{IOU}$ as 0.35. For SOT part, we get the pretrained DaSiamRPN [39] model on OTB dataset [32] and use it in our experiment. The score determining whether to add the SOT box to the track is computed as:

$$score_i = \begin{cases} score_{SOT} & \text{if } \nexists score_{i-1} \\ score_{SOT} * score_{i-1} & \text{if } \exists score_{i-1} \end{cases}, \tag{9}$$

where $score_{SOT}$ denotes the SOT score in current frame. If the last bounding box of the trajectory is also generated by SOT, we multiply $score_{SOT}$ by the score in previous frame. Otherwise we simply use the current SOT score. We set 0.998 as the threshold $\delta_{SOT}$ in stable frames and 0.4 in locomotor ones. Only if $score_i$ exceeds the threshold, can the SOT box be attached. Meanwhile, we set $L_{min}$ as 3 to filter out badly short tracks. For the final forward tracking session, thresholds for the stopping condition include 0.1 as $\delta_{adIOU}$, 0.5 as $\delta_{fIOU}$, and 0.998 as $\delta_{fSOT}$.

**Trajectory connection.** In virtue of OSNet [34], Two different models with same structure are trained for pedestrian and vehicles respectively. Trajectories are merged supposing their Euclidean distance less than 20 for pedestrian and 30 for vehicles. We connect them with linear interpolation if they are apart from each other no more than 40 frames.

### 4.3. Evaluation and results

The protocol in [24] is used to evaluate the tracking performance. A track is considered correct if the IOU overlap with ground truth is larger than a threshold. Three thresholds in evaluation, *i.e.*, 0.25, 0.50, and 0.75 are employed here. The performance of an algorithm is evaluated by averaging the mean average precision (mAP) across object classes over different thresholds.

We evaluate the performance of our approach on the VisDrone2019-MOT test-challenge dataset and obtain the results as shown in Table 1. Our HMTT attains the highest mAP comparing to the six baseline methods and Ctrack[38], the MOT-track winner of VisDrone-VDT2018 Challenge. Meanwhile, among a total of 12 teams participating in the MOT task this year, our proposed method ranks the fourth place. Some final visualizations are shown in Figure 5.

## 5. Conclusion

In this work, we propose a novel approach for Multi-Object Tracking in visual data taken from drones. To adapt perspective variations, we improve the object detector by fine classification. Detection filtering is performed to get rid of FPs which lead to object shifting and tracking failing. Besides, we screen out the frames affected by the fast motion of drones using two-norm of the homography matrix for separated treatment. With the help of single object tracking and re-identification, the Multi-Object Tracking performance gets promoted significantly. The proposed method performs well on the VisDrone2019 Multi-Object Tracking test-challenge dataset.

## 6. Acknowledgements

This work is supported by Chinese National Natural Science Foundation (U1931202, 61532018).

## References

[1] Seung-Hwan Bae and Kuk-Jin Yoon. 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*, pages 1218–1225, 2014.

[2] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3464–3468. IEEE, 2016.

[3] Erik Bochinski, Volker Eiselein, and Thomas Sikora. High-speed tracking-by-detection without using image information. In *2017 14th IEEE International Conference on Ad-*

Figure 5. A tracking example of the proposed method. The tracks are identified by the color of bounding boxes.

| Method | AP | AP@0.25 | AP@0.5 | AP@0.75 | AP car | AP bus | AP truck | AP ped | AP van |
|---|---|---|---|---|---|---|---|---|---|
| IHTLS[5] | 4.72 | 8.60 | 4.34 | 1.22 | 12.07 | 2.38 | 5.82 | 1.94 | 1.40 |
| $H^2T$[30] | 4.93 | 8.93 | 4.73 | 1.12 | 12.90 | 5.99 | 2.27 | 2.18 | 1.29 |
| CEM[20] | 5.70 | 9.22 | 4.89 | 2.99 | 6.51 | 10.58 | 8.33 | 0.70 | 2.38 |
| TBD[8] | 5.92 | 10.77 | 5.00 | 1.99 | 12.75 | 6.55 | 5.90 | 2.62 | 1.79 |
| GOG[25] | 6.16 | 11.03 | 5.30 | 2.14 | 17.05 | 1.80 | 5.67 | 3.70 | 2.55 |
| CMOT[1] | 14.22 | 22.11 | 14.58 | 5.98 | 27.72 | 17.95 | 7.79 | 9.95 | 7.71 |
| Ctrack[38] | 16.12 | 22.40 | 16.26 | 9.70 | 27.74 | 28.45 | 8.15 | 7.95 | 8.31 |
| **HMTT(ours)** | **28.67** | **39.05** | **27.88** | **19.08** | **44.35** | **30.56** | **18.75** | **26.49** | **23.19** |

Table 1. Comparison with baseline methods and Ctrack on the VisDrone2019-MOT test-challenge dataset. Ctrack is the MOT-track winner of VisDrone-VDT2018 Challenge and all the six baseline methods are submitted by the VisDrone Team.


*vanced Video and Signal Based Surveillance (AVSS)*, pages 1–6. IEEE, 2017.

[4] Erik Bochinski, Tobias Senst, and Thomas Sikora. Extending iou based multi-object tracking by visual information. In *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6. IEEE, 2018.

[5] Caglayan Dicle, Octavia I Camps, and Mario Sznaier. The way they move: Tracking multiple targets with similar appearance. In *Proceedings of the IEEE international conference on computer vision*, pages 2304–2311, 2013.

[6] 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.

[7] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 6(6):381–395, 1981.

[8] Andreas Geiger, Martin Lauer, Christian Wojek, Christoph Stiller, and Raquel Urtasun. 3d traffic scene understanding from movable platforms. *IEEE transactions on pattern analysis and machine intelligence*, 36(5):1012–1025, 2013.

[9] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[10] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[11] Meng Ru Hsieh, Yen Liang Lin, and Winston H. Hsu. Drone-based object counting by spatially regularized regional proposal network. 2017.

[12] Kalman and R. E. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering Transactions*, 82(1):35–45, 1960.

[13] Tino Kutschbach, Erik Bochinski, Volker Eiselein, and Thomas Sikora. Sequential sensor fusion combining probability hypothesis density and kernelized correlation filters for multi-object tracking in video data. In *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–5. IEEE, 2017.

[14] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 734–750, 2018.

[15] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8971–8980, 2018.

[16] Siyi Li and Dit-Yan Yeung. Visual object tracking for unmanned aerial vehicles: A benchmark and new motion models. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[17] Tsung Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. 2014.

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

[19] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[20] Anton Milan, Stefan Roth, and Konrad Schindler. Continuous energy minimization for multitarget tracking. *IEEE transactions on pattern analysis and machine intelligence*, 36(1):58–72, 2013.

[21] Matthias Mueller, Neil Smith, and Bernard Ghanem. A benchmark and simulator for uav tracking. *Far East Journal of Mathematical Sciences*, 2(2):445–461, 2016.

[22] James Munkres. Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics*, 5(1):32–38, 1957.

[23] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European conference on computer vision*, pages 483–499. Springer, 2016.

[24] E Park, W Liu, O Russakovsky, J Deng, FF Li, and A Berg. Large scale visual recognition challenge 2017, 2017.

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

[26] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[27] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[28] Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In *European Conference on Computer Vision*, 2016.

[29] Zheng Tang and Jenq-Neng Hwang. Moana: An online learned adaptive appearance model for robust multiple object tracking in 3d. *IEEE Access*, 7:31934–31945, 2019.

[30] Longyin Wen, Wenbo Li, Junjie Yan, Zhen Lei, Dong Yi, and Stan Z Li. Multiple target tracking based on undirected hierarchical relation hypergraph. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1282–1289, 2014.

[31] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3645–3649. IEEE, 2017.

[32] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Object tracking benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1834–1848, 2015.

[33] Shengping Zhang, Xiangyuan Lan, Hongxun Yao, Huiyu Zhou, Dacheng Tao, and Xuelong Li. A biologically inspired appearance model for robust visual tracking. *IEEE transactions on neural networks and learning systems*, 28(10):2357–2370, 2016.

[34] Kaiyang Zhou, Yongxin Yang, Andrea Cavallaro, and Tao Xiang. Omni-scale feature learning for person re-identification. *arXiv preprint arXiv:1905.00953*, 2019.

[35] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019.

[36] Xingyi Zhou, Jiacheng Zhuo, and Philipp Krahenbuhl. Bottom-up object detection by grouping extreme and center points. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 850–859, 2019.

[37] Pengfei Zhu, Longyin Wen, Xiao Bian, Haibin Ling, and Qinghua Hu. Vision meets drones: A challenge. *arXiv preprint arXiv:1804.07437*, 2018.

[38] Pengfei Zhu, Longyin Wen, Dawei Du, Xiao Bian, Haibin Ling, Qinghua Hu, Haotian Wu, Qinqin Nie, Hao Cheng, Chenfeng Liu, et al. Visdrone-vdt2018: The vision meets drone video detection and tracking challenge results. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018.

[39] Zheng Zhu, Qiang Wang, Bo Li, Wei Wu, Junjie Yan, and Weiming Hu. Distractor-aware siamese networks for visual object tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 101–117, 2018.