Multi-Target Multi-Camera Drone Tracking

YUCHANG JIANG[1]

The use of drones has increased dramatically in recent years. To monitor different use cases of drones, a visual tracking system for multi-target multi-camera (MTMC) drone tracking tasks is required. Yet intensive studies have been conducted in the field of MTMC, most of them focus on pedestrian tracking or vehicle tracking, leaving drone tracking underexplored. To tackle this problem, we have proposed an approach to track multiple drones in a roughly synchronized static camera network with unknown camera poses. This setting simulates practical applications, such as monitoring drones in a public webcam network. First, the existing method developed for pedestrian tracking is evaluated on our drone data to generate single-camera tracking results. Then cross-camera matching is formulated as an optimization problem with the single-camera tracking result as input. The results of single-camera tracking show that we can apply methods developed for pedestrian tracking to drone tracking. Besides, the experiments of cross-camera association demonstrate the limits of visual features in the drone tracking scenario and prove the utility of geometry features extracted from drone trajectories.

1 Introduction

Due to the increasing popularity of drones in recent years, drone monitoring is eagerly in demand. UAV-related accidents have been reported more often recently because it is hard to observe a small drone and a missing alarm can lead to catastrophic results. Therefore, it is an essential and urgent task to develop a reliable drone tracking system. Furthermore, thanks to the widely-installed public camera networks, achieving drone surveillance with a multi-camera network is cost-effective, scalable, precise, and desired in real applications or commercial projects.

To perform drone surveillance in practical applications (e.g. drone monitoring in the airport), a high-level task, multi-drone multi-camera tracking (MDMCT), is required but needs further investigation. The common application domain of multi-target multi-camera tracking is pedestrian tracking. Its classic solution is a two-step approach: firstly tracking objects in each camera view independently and then associating objects across cameras to find their global identities. For the first step, we can reuse the single-camera tracker designed for pedestrian tracking in our drone tracking task. However, it is hard to directly utilize the commonly used features like appearance and geometry in the multi-camera association step in our drone tracking task. The primary challenges come from the small, fast-moving drones and the public camera network with unknown camera poses. The target objects, drones, are usually blurry and small in camera views, creating obstacle to gain enough information from appearance. Therefore, appearance features are of limited usefulness for drones. The second challenge is caused by unknown camera poses in camera networks. Although it is common to leverage geometric relationships to associate tracks across camera views, the use of geometric features requires camera poses, which are unknown in most practical cases (e.g. outdoor webcam data).

[1] ETH Zürich, Switzerland, E-Mail: yujiang@ethz.ch

This presented work focuses on building a tracking system for multiple drones in a camera network with overlapping views. To simulate real applications like drone surveillance in an airport, we consider the camera network consists of synchronized stationary cameras without spatial calibration. This work builds a two-step approach to track multiple drones in a synchronized camera network without knowing camera poses. As shown in Fig. 1, single-camera tracking (SCT) is performed independently on each camera view and then in the multi-camera tracking (MCT) stage, resulted single-camera tracks are fed into multi-camera matchers to figure out their global identities. In the single-camera stage, an established multi-object tracking (MOT) method from pedestrian tracking is applied to generate tracking results and a post-processing procedure is utilized to produce more robust long tracks. Then, both geometric and visual features are employed to match tracks across camera views in the multi-camera stage. In the end, this work generates tracking results for drones in a synchronized camera network.

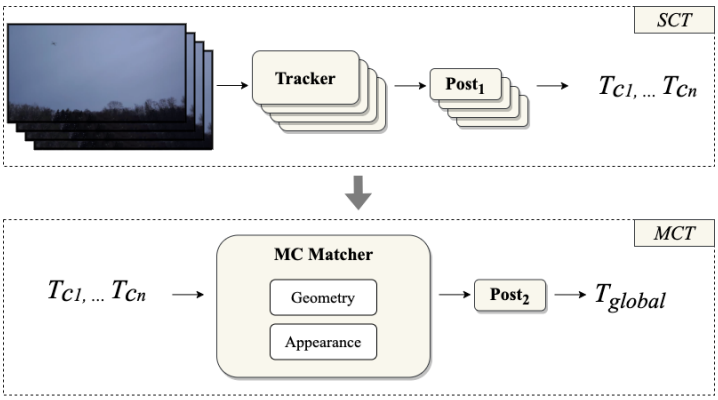

Fig. 1: Pipeline of the proposed method

2 Methodology

2.1 Data

Our dataset consists of video sequences from four synchronized static cameras. The four cameras are Sony a5100, Sony G, Samsung S10, and Sony nex5n. This challenging dataset contains three drones in a snow-covered background. Three drones are Pixhawk drone, DJI Phantom, and DJI Mavic. Annotations (ground truth tracks of drones) of acquired video sequences are labelled by human operators. For each camera, the whole video sequence is around 10 minutes. After synchronization, 16414 image frames for each camera are split into 1642 train images and 14772 test images. To observe different cases such as single-drone cases and multi-drone cases efficiently, the test set is further divided into six subsets and each subset has around 2461 frames. All train images are used to train the deep learning-based single-camera tracker, while all test images are utilized for evaluation. Several examples images are presented in Fig. 2 and Fig. 3, which shows the main difficulties of this dataset: homogeneous snow-covered background and small fast-moving objects. The homogeneous background adds the difficulty to extract distinct keypoints for camera calibration while small fast-moving objects complicates object detection and tracking.

Fig. 2: Examples of images containing large objects from four cameras

Fig. 3: Examples of images containing small objects from four cameras (different colors stand for different drones)

2.2 Generating Multi-Object Single-Camera Drone Tracks

The established tracker, Chained tracker (CTracker) (PENG et al. 2020) is utilized to track drones in each camera view. Since it raises the concept of node chaining to leverage temporal information of tracks, it is suitable to track fast-moving objects, drones. As indicated in Fig. 4, node chaining means treating a pair of consecutive frames as a node and defining a pair of successive nodes as a

node chaining. If the objects detected in the common frame (F_t) are highly overlapped based on Intersection-over-Union(IoU) score, the related detection in this node chaining (F_{t-1}, F_t, F_{t+1}) should belong to the same track.

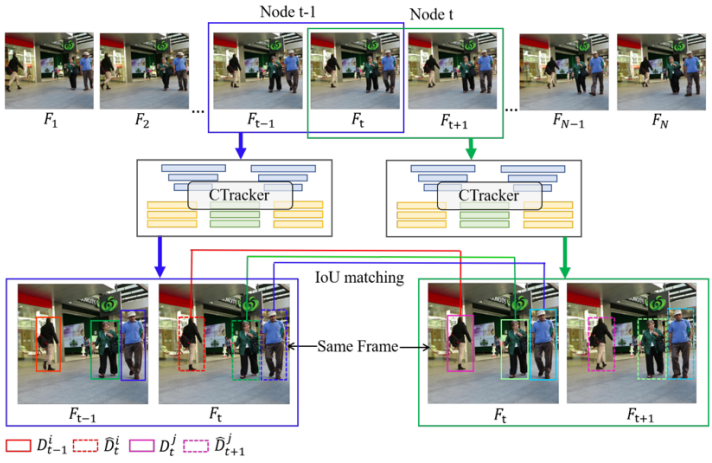

Fig. 4: Concept of node chaining (PENG et al. 2020)

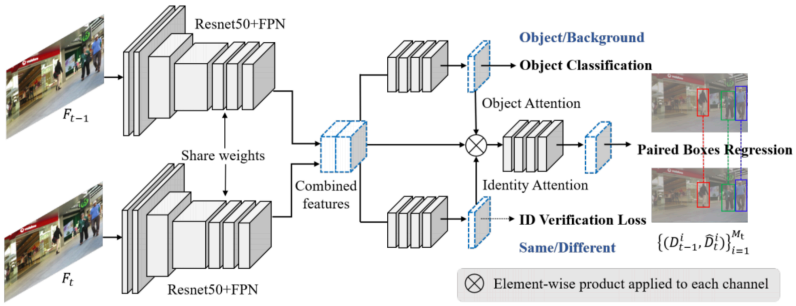

Fig. 5: Network architecture of CTracker (PENG et al. 2020)

More specifically, the detailed structure of CTracker is sketched in Fig. 5. It takes a pair of consecutive image frames as input and uses a pretrained neural network to extract features for each image. Then features from both images are concatenated and fed into two sub-models: a classification model to classify whether the detection contains the foreground and a re-identification (ReID) model to check if the detection from two frames belongs to the same object. After that, a joint attention mechanism is designed to extract useful information from two sub-models. Then a regression layer is employed to output a pair of bounding boxes (position of a bounding box in F_{t-1} and its corresponding position in F_t). During inference, the paired bounding boxes from a node chaining are used to compute the IoU score and decide whether they should be connected. The structure of CTracker and the concept of node chaining seem applicable to drone tracking tasks because the temporal information between consecutive frames is considered, which handles the fast movements of drones.

After applying the single-camera trackers, tracking results for each camera are obtained. However, these tracking results are usually a large number of short tracks, which will increase the computational burden for camera association in the MCT stage. Since connecting short tracks into

long trajectories before cross-camera association helps to reduce the computational burden, post-processing is performed to connect short tracks and remove noisy detection.

2.3 Generating Multi-Object Multi-Camera Drone Tracks

With single-camera tracking results, the multi-camera association aims to assign global identities to tracks. The MCT stage has two main components: a pairwise camera association method and a post-processing method. The pairwise camera association extracts features from a pair of cameras to match tracks from these two cameras. Then post-processing method is performed to merge pairwise matching results and resolve conflict to assign global identities to all tracks. Here we adopt a centralized way in the multi-camera tracking: one camera is defined as the reference camera and then the pairwise association is performed between the reference camera and each of the remaining cameras. Based on the matching results, global identities are assigned.

In the single-camera multi-object problem, optimization methods like Hungarian algorithm (KUHN 1955) are usually chosen to find the one object-to-one object (1V1) assignment and match objects in consecutive frames. As one object from the previous frame can match to at most one object in the current frame, this is a standard assignment problem. However, in the multi-camera scenario, associating tracks from two cameras is more complicated since a long track in camera A can be matched to multiple short tracks in camera B, making pairwise camera association an NP-hard problem. To fulfil this multiple track-to-multiple tracks (NVN) requirements approximately and simplify the optimization problem, we split the whole time domain (e.g. 2000 frames) into equal-sized sliding windows (e.g. 300 frames) and then solve the 1V1 assignment problem for each sliding window independently. The assumption is that the 1V1 relationship is approximately preserved in a short sliding window. With this assumption, we can extract features to compute the cost or affinity between tracks and then perform the classic Hungarian algorithm to solve the assignment problem in each sliding window. The pseudo-code of this algorithm is listed in Alg. 1. This algorithm takes two sets of tracks from two cameras as input and outputs the matches of tracks.

Input tracks from two cameras, T_{c1}, T_{c2}
Output matches of tracks from two cameras $M = \{(t_{c1}, t_{c2}, start, end) | t_{c1} \in T_{c1}, t_{c2} \in T_{c2}\}$

procedure MATCH PAIRWISE CAMERAS
  $M \leftarrow \emptyset$
  for each sliding window in time **do**
    $start \leftarrow start_{window}$ ▷ starting time of the current window
    $end \leftarrow end_{window}$ ▷ ending time of the current window
    $cost \leftarrow \infty$
    for each pair of t_i from $c1$, t_j from $c2$ inside current window **do**
      if t_i, t_j overlap in time **then**
        $cost(t_i, t_j) = \Psi_{affinity}(t_i, t_j)$
    Add dustbin node to $cost$
    $M' = Hungarian\ algorithm(cost)$
    $M = M \cup M'$

Alg. 1: Algorithm of pairwise camera association

In each sliding window, the track-to-track cost matrix $C \in \mathbb{R}^{n \times n}$ is formed (n is the total number of tracks in this sliding window) and each element $c_{i,j}$ contains the distance or affinity measure of two tracks, i and j. During the computation of the cost matrix, dustbin nodes are concatenated to

deal with objects that only appear in one camera view, motivated by the dustbin concept in (SARLIN et al. 2020). Then Hungarian algorithm is applied to the cost matrix to find tracks belonging to the same object.

To measure track affinity ($\Psi_{affinity}$), two types of features, appearance (Ψ_{app}) and geometry (Ψ_{geo}), are considered. The pretrained ReID component from CTracker extracts appearance features of detections in each track. The average appearance feature of each track is defined as the representative feature and the Euclidean distance between any two tracks are computed as appearance cost. Compared to visual features, geometric features are more complicated. To leverage geometric relationships, camera poses are usually required, which is unknown in our case. A natural solution is to extract keypoints and then compute fundamental matrix to obtain geometric relationships. However, in the practical scenario of drone tracking, the outdoor background of video sequences is usually sky or ground. Hence, it is hard to find robust and distinctive keypoints from the static background. Inspired by the previous work (LI et al. 2020), we consider the dynamic part, the detected drones, as keypoints. For a pair of tracks, center points of detection from temporal-overlapping frames are treated as corresponding points. Geometric relationships such as fundamental matrix and homography are obtained from these detections. There are two affinity measurements for geometric features: homography error (H) and epipolar line error (F). Homography of two tracks can be computed with the RANSAC method (FISCHLER & BOLLES 1981). Then points in image 2 can be transformed to image 1 to calculate Euclidean distance as affinity score. Similarly, the fundamental matrix of two tracks can be computed and then the epipolar line error can reflect the affinity between two tracks. To combine both features, the affinity score is defined as Eq. 1, which means the affinity score prefers appearance feature more if the object's bounding box is close to the defined size of 'large' drones, λ_{box}. This equation is based on the assumption that larger objects usually provide richer visual features while small objects can be considered as points to utilize geometry features.

$$\Psi_{affinity} = w_{app}\Psi_{app} + (1 - w_{app})\Psi_{geo}$$
$$w_{app} = min(\frac{box\ size}{\lambda_{box}}, 1) \quad (1)$$

After accomplishing pairwise camera association, a post-processing procedure is designed to combine these matching results across all cameras and resolve possible conflicts to build trajectories with global identities. In the end, each track is assigned with a new global identity and across-camera tracking is achieved.

3 Results and Discussion

For evaluation, IDF1 (identity F1 score) is adopted as the primary metric. It penalizes false negative and false positive predictions based on a bijective mapping between trajectories and ground truth tracks. In experiments, all test sets are evaluated and their average IDF1 values are summarized for both single-camera tracking and multi-camera tracking stage.

Fig. 6: Visual results of single-camera tracking (left: before post-processing, right: after post-processing)

In the single-camera tracking stage, it is noticed that CTracker tends to produce short tracks but these tracks can be connected into longer ones with a post-processing method. For example, from Fig. 6, several short tracks in the left image (before post-processing) are combined into a longer one in the right image (after post-processing), which shows the effect of post-processing and the result single-camera tracking. At the end of single-camera tracking, the IDF1 can achieve 65.27%.

With single-camera tracking results, different features are adopted to realize multi-camera association. In Tab. 1, results of using appearance feature only, geometry feature only, and the combination of both features are summarized. It proves the effectiveness of geometry features in the drone tracking application.

Tab. 1: Result of multi-camera trackers with different affinity measures

	Appearance	**Geometry (H)**	**Geometry (F)**	**App + Geo (H)**	**App + Geo (F)**
IDF1	40.55 %	43.29 %	43.50 %	50.04 %	43.45 %

4 Conclusion

The proposed pipeline can partially solve the multi-drone multi-camera tracking problem. It can generate single-camera tracks with the existing algorithm and then associate tracks with appearance and geometric features. The contributions of this work include: evaluate the state-of-the-art tracker on a drone dataset, explore possible features in multi-camera drone tracking scenarios, and leverage geometric features in a camera network without camera calibration. However, the proposed solution in the multi-camera association step approximates the 1-to-N association problem with temporal windows. Still, the information in all frames from all cameras remained underexploited, suggesting the possibilities to reformulate the problem exactly in order to fully use the information. Besides, more training data can improve the single-camera tracker's performance and the results of the multi-camera association.

5 Bibliography

FISCHLER, M.A. & BOLLES, R.C., 1981: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM, **24**(6), 381-395.

LI, J., MURRAY, J., ISMAILI, D., SCHINDLER, K.& ALBL, C., 2020: Reconstruction of 3D flight trajectories from ad-hoc camera networks. In 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 1621-1628.

KUHN, H.W., 1955: The Hungarian method for the assignment problem. Naval research logistics quarterly, **2**(1-2), 83-97.

PENG, J., WANG, C., WAN, F., WU, Y., WANG, Y., TAI, Y., WANG, C., LI, J., HUANG, F. & FU, Y., 2020: Chained-tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking. European Conference on Computer Vision, Springer, Cham, 145-161.

SARLIN, P.E., DETONE, D., MALISIEWICZ, T. & RABINOVICH, A., 2020: Superglue: Learning feature matching with graph neural networks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 4938-4947.