3D TRAJECTORY RECONSTRUCTION OF DRONES USING A SINGLE CAMERA

Seo-Bin Hwang[1] *Han-Young Kim*[2] *Chae-Yeon Heo*[2]
You-Kyoung Na[2] Yeong-Jun Cho*

[1*]Artificial Intelligenc Convergence, Chonnam National University, Gwangju, Korea
[2]Artificial Intelligence, Chonnam National University, Gwangju, Korea

ABSTRACT

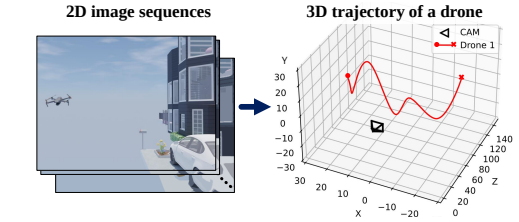
Drones have been widely employed in various fields, but the number of drones being used illegally and for hazardous purposes has recently increased. To prevent illegal drones, in this work, we propose a novel framework for reconstructing three-dimensional (3D) drone trajectories using a single camera. By leveraging calibrated cameras, we exploit the relationship between 2D and 3D spaces. We automatically track the drones in 2D images using a drone tracker and estimate their 2D rotations. By combining the estimated 2D drone positions with their actual length and camera parameters, we geometrically infer the 3D drone trajectories. To address the lack of public drone datasets, we also create synthetic 2D and 3D drone datasets. The experimental results show that the proposed methods accurately reconstruct drone trajectories in 3D space and demonstrate the potential of our framework for single-camera-based surveillance systems.

***Index Terms*—** Drone, Trajectory reconstruction, Single camera, Surveillance system

1. INTRODUCTION

Recently, drones have been widely employed in various fields, including security, surveillance, and disaster response [1]. However, drones without permission for illegal and hazardous purposes have also increased. For instance, drones equipped with recording devices, such as cameras and storage, can occur security risks. To address these issues, an anti-drone method that can automatically distinguish illegal drones can be employed. To this end, estimating three-dimensional (3D) trajectories of drones becomes crucial because their locations and moving patterns provide significant information (e.g., their behavior and legality).

Several approaches [2, 3] have been studied to perform a 3D drone trajectory estimation using various sensors equipped on the object, including global positioning systems (GPS), inertial measurement units (IMU), and cameras. However, in the context of detecting illegal drones, these methods are unsuitable because it is not possible to access the data stored in the drones. Therefore, we focus on trajectory reconstruction approaches that analyze externally acquired data rather than



(a) An example of the 3D trajectory reconstruction result.

(b) Challenges in 3D trajectory reconstruction of drones.

Fig. 1: Example and challenges of 3D trajectory reconstruction of drones using a single camera.

using internal drone data. To reconstruct 3D trajectories of objects, numerous studies [4, 5, 6] have proposed methods that use various sensors, including lidar, radar, and RGB-D cameras. These methods aim to fuse sensor data to detect 3D objects and accurately determine their positions. However, using multiple sensors can be expensive and impractical.

In this work, we focus on a surveillance system using a single static camera, such as closed-circuit television (CCTV), because it is cost-effective and common in urban areas compared to other multi sensor-based systems. Based on these advantages, we propose methods for 3D drone trajectory reconstruction using a single camera(Fig. 1 (a)). Recently, several methods [7, 8] have attempted to reconstruct the 3D trajectory of objects using a single camera. However, these methods highly rely on prior knowledge, such as the ground plane and calibration grids. Unlike common objects, reconstructing the drone trajectory is challenging, as shown in Fig. 1 (b). First, drone image scenes have limited prior knowledge (e.g., other object structures, buildings, and ground planes) because drones fly at high altitudes. Second, drones can rotate freely compared to other common objects. Third, a drone is not a common object, but a novel object [9] that has not been extensively explored in detection applications. For example, the

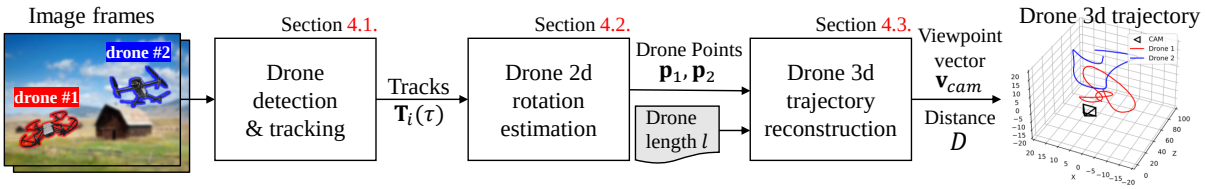


Fig. 2: Overall framework for 3D drone trajectory reconstruction.

third figure in Fig. 1 (b) shows object categories provided in MSCOCO [10] dataset, but images for drones are missing.

To overcome the limitations, we propose a new framework for 3D drone trajectory reconstruction using a single camera. We employ calibrated cameras to exploit the relationship between 2D and 3D spaces. A drone tracker in our framework automatically tracks each drone in the 2D image, and we find further 2D drone rotations in Sec. 3.2. Using the estimated 2D drone position and its actual length information from the drone speculation database, we infer the 3D drone positions as proposed in Sec. 3.3. Moreover, we created both 2D and 3D synthetic drone datasets including various scenarios to address the lack of drone datasets in Sec. 4. In addition, we validated our approach in various environments by generating real drone data and estimating the trajectories.

Using our datasets, we validated the effectiveness of the proposed methods in Sec. 5. Despite using a single camera view, our methods accurately reconstructed the 3D trajectories of multiple drones. The proposed 2D rotation estimation method effectively improved the trajectory reconstruction performance. In addition, due to our 2D drone image datasets, the performance of the drone detector was improved, resulting in improved trajectory reconstruction performance as well. The results strongly affirm the capability of our framework to accurately reconstruct drone trajectories in 3D space and demonstrate the potential for applying the framework in single-camera surveillance systems.

The main contributions of this work are summarized as follows. First, this attempt is the first to reconstruct the 3D drone trajectory using a single camera without any known patterns. Second, we propose new methods to overcome the challenges in 3D drone trajectory reconstruction. Third, we provide new 2D and 3D synthetic and real-world datasets of drones. We hope that this study provides meaningful guidelines to readers who aim to implement a surveillance system for drones in industrial and academic fields.

2. RELATED WORKS

Many studies have been proposed to localize an object in 3D space [4, 5, 6] using multiple 3D sensors, such as lidar, radar, and RGB-D cameras. For example, Chen *et al.* [11] utilized deep neural networks to reconstruct 3D trajectories using a spectrum-sensing dataset. Further, Nabati *et al.* [5] fused the radar and camera sensors for robust and accurate 3D multi-object tracking. Many methods based on multiple cameras have been proposed. For example, Rozantsev *et al.* [12] estimated a six-degrees-of-freedom (6-DOF) trajectory of the flying drone using multiple ground cameras. In addition, Li *et al.* [13] proposed the 3D reconstruction of drone trajectory using multiple unsynchronized cameras with unknown extrinsic camera parameters. The multiple camera systems can be applied to other applications, such as 3D human tracking [14, 15] and 3D ball tracking [16, 17] in many sport scenes. However, using such sensors or multiple cameras is expensive and impractical for standard surveillance systems (e.g., CCTV).

Geometrically, more than two camera views are required to reconstruct a 3D object trajectory. Despite the challenge, several methods have attempted to estimate the 3D positions of target objects using a single camera. Rougier *et al.* [18] estimated the 3D trajectory of the human head based on the assumption of the known human height and a calibrated camera. Similarly, Chen *et al.* [7] employed prior knowledge regarding a basketball court to reconstruct the 3D trajectory of a basketball. Recently, Srinivasan *et al.* [8] proposed a 3D reconstruction of bird flight trajectories with a known calibration grid. In this way, these methods applied structural cues of the scenes (e.g., ground plane, human heights, court structure, and known grid patterns). In contrast, learning-based methods [19, 20] that directly estimate depth using a single camera have also been proposed. However, in the case of drone trajectory reconstruction, there is limited prior knowledge available, such as scene and object structures, as shown in Fig. 1. Therefore, previous approaches that heavily rely on prior knowledge are ineffective for drone trajectory reconstruction.

3. PROPOSED METHODS

The overall framework for 3D drone trajectory estimation is illustrated in Fig. 2. It consists of three primary parts: a drone tracker, a drone 2D rotation estimator, and 3D drone trajectory re-constructor. The framework takes consecutive 2D image frames as input and generates 3D drone trajectories in a 3D camera coordinate system as outputs. First, the drone tracker detects and tracks the drones to estimate their locations, sizes, and identities in 2D image coordinates (Sec. 3.1). Next, we further estimate the 2D rotations of the tracked drones using the principal component analysis (PCA) [21] in Sec. 3.2. Finally, the 3D drone trajectory estimator determines the distance between the camera and drones by leveraging their relationships and the actual widths and lengths of the drones.

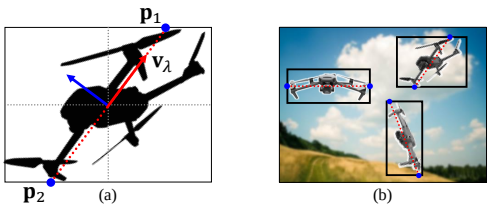


Fig. 3: (a) The black area is the foreground drone area. Red arrow is the eigenvect $\mathbf{v}_\lambda$. $\mathbf{p}_1, \mathbf{p}_2$ are the intersection points between $(\bar{x}_i, \bar{y}_i, \bar{w}_i, \bar{h}_i)$, and $\mathbf{v}_\lambda$. (b) Examples of 2D drone rotation estimation results.

3.1. Drone Detection and Tracking

To estimate the 3D drone trajectory, we first estimated the 2D drone trajectories given the image frames. To this end, we can utilize real-time multiple object trackers [22, 23, 24]. Recently, many trackers have adopted a tracking-by-detection strategy that associates the detection responses with building the object trajectory. Therefore, they essentially include object detectors such as the you only look once (YOLO) series [25, 26] or the faster region-based convolutional neural network (Faster R-CNN) [27] in their frameworks. The object detection results are represented by $\mathbf{D}_n(t)$, where n represents the index of detection responses, and t denotes the frame index. Each detection consists of five-dimensional vectors $\mathbf{D}_n = (x_n, y_n, w_n, h_n, c_n)$, where (x, y) indicates the center position, (w, h) denotes the width and height, and c represents the drone class. After applying multiple object tracking, the track is denoted by $\mathbf{T}_i(\tau) = (\bar{x}_i, \bar{y}_i, \bar{w}_i, \bar{h}_i, c_i)$, where i represents an index of the track, τ denotes the track lifetime, $(\bar{x}, \bar{y}, \bar{w}, \bar{h})$ denotes the ith track area estimated by the tracker, and c_i indicates the tracked drone class.

Our framework can apply to any kind of drone tracker and detector. However, a large drone image dataset is required to train a drone detector. Unfortunately, the drone is considered novel objects [9], which have not been extensively explored in drone detection applications. Many public datasets for object detection, such as ImageNet [28] and MSCOCO [10], do not provide drone images. Although several studies [29, 30] have a drone image dataset, they provide only a single drone class. Considering this problem, we built a new 2D drone image dataset in Sec. 4.1.

3.2. Drone 2D Rotation Estimation

Drones can freely rotate in the air compared to other common objects, so estimating the rotation of drones is important than other common objects. However, the bounding box (bbox) of a drone only offers a simple rectangular position (x, y, w, h) that cannot precisely depict the drone rotation. To find the drone rotation in the rectangular bbox, we applied the PCA [21]. Based on the tracking results, the estimated drone position is defined by $(\bar{x}_i, \bar{y}_i, \bar{w}_i, \bar{h}_i)$. Then, a set of pixel coordinates belonging to the foreground drone area is defined as follows:

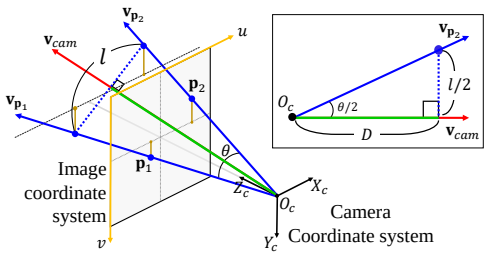


Fig. 4: Geometric relationship between camera and image coordinate systems. The green line is the distance D between the center of drone and camera. Best viewed in color.

$$\mathcal{C} = \{(x_k, y_k) \,|1 \leq k \leq K\}, \tag{1}$$

where K is the total number of pixels. We omit the track index i for convenience. To automatically find the foreground area of the drone, we utilized U2-net [31]. Then, we calculate the covariance matrix of $\mathcal{C}$ as

$$\mathbf{\Sigma} = \frac{1}{K} \begin{pmatrix} \sum(x_k - m_x)^2 & \sum(x_k - m_x)(y_k - m_y) \\ \sum(x_k - m_x)(y_k - m_y) & \sum(y_k - m_y)^2 \end{pmatrix}, \tag{2}$$

where m_x, m_y are means of x and y values, respectively.

Based on the PCA, two eigenvalues and eigenvectors are calculated from the covariance matrix $\mathbf{\Sigma}$, and we selected the eigenvector $\mathbf{v}_\lambda$ corresponding to a larger eigenvalue λ. The eigenvector reflects the drone rotation, as shown in Fig. 3 (a). Based on the vector, we find two points $(\mathbf{p}_1, \mathbf{p}_2)$ of intersection between the eigenvector and drone position $(\bar{x}_i, \bar{y}_i, \bar{w}_i, \bar{h}_i)$. To find the intersection points, we defined a line that passes through the drone center point $(\bar{x}, \bar{y})$, with its direction determined by the eigenvector $\mathbf{v}_\lambda$. As a result, we can approximately estimate the 2D drone rotations regardless of their poses, as shown in Fig. 3 (b). This aids in enhancing the overall accuracy of the 3D drone trajectory reconstruction.

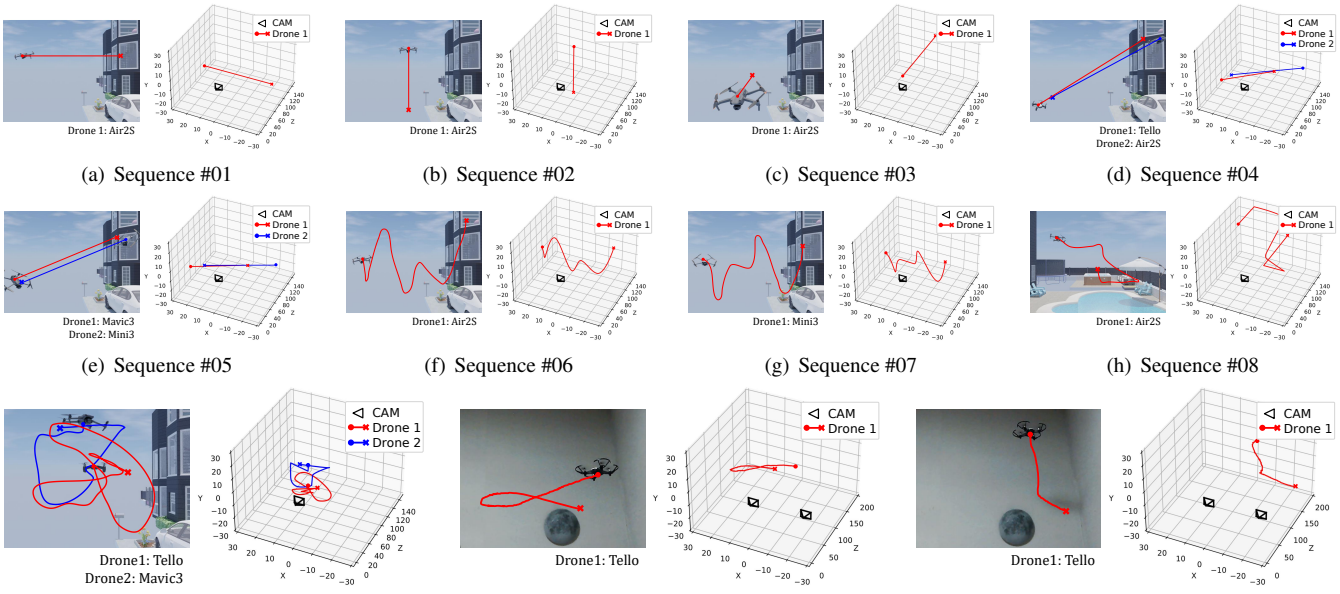
3.3. Drone 3D trajectory reconstruction

To reconstruct the 3D trajectory of the drone, in this section, we exploit the prior knowledge of each drone such as its specification information. We assume that the principle line $\overline{\mathbf{p}_1\mathbf{p}_2}$ of the drone corresponds to the the longest side of the drone. In general, the width length of the drone is longer than its other sides e.g., height and depth. Based on this assumption, we retrieve an actual drone's width length l from the drone specification database when the drone detector predicts the class of the drone as c. Then, we have two end points $(\mathbf{p}_1, \mathbf{p}_2)$ of the drone in 2d image, and its real 3D length l.

Using those cues (i.e., $\mathbf{p}_1, \mathbf{p}_2, l$) of the drone, we can estimate 3D position of the drone. To this end, we calibrate the camera to find the relationship between 2d image coordinate (u, v) and 3D world coordinate systems (X, Y, Z) as

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K}[\mathbf{R}|\mathbf{t}] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \tag{3}$$

where $\mathbf{K}^{3\times3}$ matrix denotes camera intrinsic parameters, $\mathbf{R}^{3\times3}, \mathbf{t}^{3\times1}$ denote camera extrinsic parameters.



(a) Sequence #01 (b) Sequence #02 (c) Sequence #03 (d) Sequence #04

(e) Sequence #05 (f) Sequence #06 (g) Sequence #07 (h) Sequence #08

(i) Sequence #09 (j) Sequence #10 (real-world) (k) Sequence #11 (real-world)

Fig. 5: Examples of image frames and the corresponding 3D drone trajectories of the 3D drone synthetic dataset. The • and × symbols denote the start and end of the drone trajectory, respectively. Scenario#1 – #9 are the synthetic and Scenario#10 – #11 are the real-world dataset.

By calculating $[\mathbf{R}|\mathbf{t}] \cdot [X, Y, Z, 1]^\top$, the world coordinate system is transformed into a camera coordinate system as

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}. \tag{4}$$

In the camera coordinate system, the origin $O_c = [0, 0, 0]$ represents the center of the camera. Since the $\mathbf{K}$ matrix is invertible, we can have a linear transformation of the 2D image coordinate to the 3D camera coordinate system by

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \mathbf{K}^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}. \tag{5}$$

According to Eq. 5, we can transform the points $(\mathbf{p}_1, \mathbf{p}_2)$ in the 2d image coordinates system into 3D vectors in the camera coordinate system by calculating

$$\mathbf{v}_{\mathbf{p}_1} = \mathbf{K}^{-1} \begin{bmatrix} \mathbf{p}_1 \\ 1 \end{bmatrix}, \quad \mathbf{v}_{\mathbf{p}_2} = \mathbf{K}^{-1} \begin{bmatrix} \mathbf{p}_2 \\ 1 \end{bmatrix}. \tag{6}$$

The vectors $\mathbf{v}_{\mathbf{p}_1}$ and $\mathbf{v}_{\mathbf{p}_2}$ are heading to the points $(\mathbf{p}_1, \mathbf{p}_2)$ from the origin (O_c). Additionally, a camera viewpoint vector to the center of the drone is defined by $\mathbf{v}_{cam} = \frac{\mathbf{v}_{\mathbf{p}_1} + \mathbf{v}_{\mathbf{p}_2}}{2}$. Fig. 4 visualizes both image and camera coordinates with the 2D image points $(\mathbf{p}_1, \mathbf{p}_2)$ and vectors $(\mathbf{v}_{\mathbf{p}_1}, \mathbf{v}_{\mathbf{p}_2}, \mathbf{v}_{cam})$.

Assuming that the vector $\overrightarrow{\mathbf{v}_{\mathbf{p}_1}\mathbf{v}_{\mathbf{p}_2}}$ is orthogonal to a camera-view point vector $\mathbf{v}_{cam}$, we can calculate the distance between the drone and camera center using the following steps. First, we compute a vector angle θ between $\mathbf{v}_{\mathbf{p}_1}$ and $\mathbf{v}_{\mathbf{p}_2}$ as follows:

$$\theta = cos^{-1} \frac{\mathbf{v}_{\mathbf{p}_1} \cdot \mathbf{v}_{\mathbf{p}_2}}{\|\mathbf{v}_{\mathbf{p}_1}\| \, \|\mathbf{v}_{\mathbf{p}_2}\|}. \tag{7}$$

The distance between two vectors is known as l, which is an actual drone length. Based on the θ and l values, we can calculate the distance between the camera and drone by

$$D = \frac{l}{2 \tan \frac{\theta}{2}}. \tag{8}$$

Then, we can determine the 3D position of the drone according to the distance D and camera-view point vector $\mathbf{v}_{cam}$. The 3D drone position estimation can be conducted across multiple image frames to build the 3D trajectory. However, a reconstructed 3D trajectory using a single camera may contain noise and error. To mitigate these problems, we applied an average filter to the initial trajectory. By smoothing out variations in the trajectory, it helps the average filter enhance the accuracy of the 3D drone position estimation.

4. DATASETS

All datasets in this section are available at https://will_be_available.

4.1. 2Drone(on+aug): 2D Drone Image dataset

Public object detection datasets (e.g., ImageNet [28] and MSCOCO [10]) do not provide drone images for training the detector. Several studies [29, 30] have built drone image datasets, but they only provide a single drone class. Considering these problems, we created a new 2D drone image dataset called 2Drone to train drone detectors. The dataset provides multi-class drone models (Air2S, Mavic3, Mini3Pro, and Tello) with diverse poses and backgrounds.

We first collected 1,107 real drone images of the four models available online and called this drone image set

Sequences	Seq. #01	Seq. #02	Seq. #03	Seq. #04	Seq. #05	Seq. #06	Seq. #07	Seq. #08	Seq. #09	Seq. #10	Seq. #11
Drones	Air2S	Air2S	Air2S	Air2S Tello	Mavic3 Mini3Pro	Air2S	Mini3Pro	Air2S	Tello Mavic3	Tello	Tello
Frames(30)	180	180	180	180	180	300	300	810	300	120	120
Translation	x	y	z	x, y, z	x, y, z	x, y, z	x, y, z	x, y, z	x, y, z	x, y, z	x, y, z
Rotation	None	None	None	None	None	None	None	✓	✓	✓	✓
Motion	Linear	Linear	Linear	Linear	Linear	Non-linear	Non-linear	Non-linear	Non-linear	Non-linear	Non-linear

Table 1: Properties of scenarios in the proposed 3D drone dataset. Translation denotes the direction of the drone movements. Rotation denotes the changes in the drone pose angle. Motion (linear) and motion (nonlinear) indicate the drone movements with constant and variable velocity, respectively.

Training Dataset	MOTA(↑)	FN(↓)	FP(↓)	IDs(↓)
`2Drone(on)`	75.3	693	**58**	5
`2Drone(aug)`	80.0	393	214	6
`2Drone(on+aug)`	**81.0**	**343**	236	**3**

Table 2: Multi-drone tracking results on `Syn3Drone` dataset according to different training datasets for the drone detector.

`2Drone(on)`. However, the volume of the dataset is insufficient to train a robust and accurate drone detector. Another work [32] synthesized objects and background images to make an object detection dataset; therefore, we synthesized drone source images with background images to augment more 2D drone images in the dataset. For this purpose, the total number of collected images is as follows: 25 for Air2S, 32 for Mavic3, 39 for Mini3Pro, 41 for Tello, and 100 for the background.

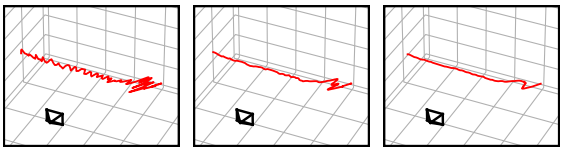
Then, we randomly selected drone and background source images and mixed them. The total number of images is 11,434, and the numbers of each drone in the images are as follows: Airs2S: 2,085, Mavic3: 2,668, Mini3Pro: 3,245, and Tello: 3,436. The dataset is called `2Drone(aug)`. To sum up, our dataset named `2Drone(on+aug)` consists of `2Drone(on)` and `2Drone(aug)` and involves 12,541 images and their ground-truth bounding boxes.

4.2. `Syn3Drone`: Synthetic 3D Drone Dataset

To validate the proposed methods, we require a 3D drone trajectory dataset (`Syn3Drone`) that provides 2D image frames and ground-truth 3D drone positions. As we know, there is no public dataset for 3D drone tracking. In this work, we built a new synthetic 3D drone trajectory dataset using an open-source 3D computer graphics software tool (blender). We collected 3D models of the four drones (Airs2S, Mavic3, Mini3Pro, and Tello) and constructed two 3D background scenes. Then, we rendered nine different drone scenarios as depicted in Fig. 5. The properties of sequences in the dataset are summarized in Tab. 1. We set the drones considering their actual sizes. The drone widths, depths, and heights in millimeters are as follows: Air2S (W:253.0, D:183.0, H:77.0), Mavic3 (W:347.5, D:283.0, H:107.7), Mini3Pro (W:245.0, D:171.0, H:62.0), and Tello (W:176.3, D:98.0, H:41.0).

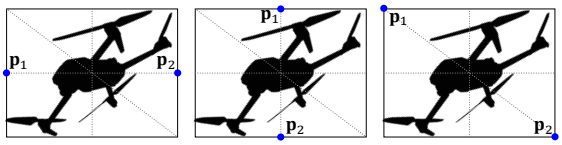
4.3. `Real3Drone`: Real-world 3D Drone Dataset

To validate our proposed framework in real-world scenarios, we generated real-drone data using the Tello drone model. To this end, we set calibrated two different cameras (i.e., stereo camera) and calculated the 3D positions of drones by bipolar geometry.



(a) Initial estimation (b) Window size: 5 (c) Window size: 10

Fig. 6: Trajectory smoothing results according to window sizes.



(a) Width-based (b) Height-based (c) Diagonal-based

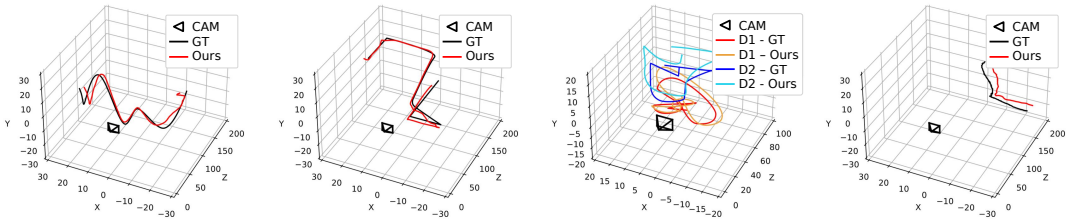
Fig. 7: Determining drone 2D points $\mathbf{p}_1, \mathbf{p}_2$.

5. EXPERIMENTAL RESULTS

5.1. Settings and Evaluation Metrics

For the multi-object tracker, we used byteTrack [23], which can perform real-time tracking. As an object detector for the tracker, we trained a YOLOv5x [33] using NVIDIA RTX 3070ti and 3090 (two cards) with the following parameter settings: 100 epochs, a batch size of 4, and a learning rate of 0.0001. Using the proposed dataset in Sec. 4.1, we evaluated the drone detector with various training & testing scenarios.

To evaluate the accuracy of multi-object trackers, we measured the multi-object tracking accuracy (MOTA) [34] defined by $MOTA = 1 - \frac{\sum_t FN_t + FP_t + IDs_t}{\sum_t g_t}$, where FN_t, FP_t, IDs_t, and g_t are the number of misses (false negatives), false positives, identity switches, and objects, respectively. To evaluate the performance of 3D trajectory estimation, we measured two commonly used metrics, the mean absolute error (MAE) and root mean squared error (RMSE). The MAE calculates the absolute differences between the predicted and actual values and measures an average summation of the differences. It provides an average measure of the absolute deviation between the predicted and actual values and is not sensitive to outliers because it treats all differences equally [35]. The RMSE also calculates the differences between the predicted and actual values, but it further squares the differences for the average of



(a) Sequence #06 (b) Sequence #08 (c) Sequence #09 (d) Sequence #11 (real)

Fig. 8: Qualitative results of the three-dimensional (3D) drone trajectory reconstruction based on the proposed methods.

Sequences	Metric	Width	Height	Diagonal	Ours
#01	MAE	4.2	32.38	**3.84**	4.53
	RMSE	5.05	48.38	**4.24**	5.63
#02	MAE	1.34	17.23	4.96	**1.23**
	RMSE	1.76	27.33	6.94	**1.72**
#03	MAE	**1.15**	27.78	5.05	2.16
	RMSE	**1.58**	48.49	9.04	3.8
#04	MAE	5	23.08	**2.38**	5.1
	RMSE	8.07	33.02	**3.8**	8.26
#05	MAE	23.54	31.86	**20.27**	25.57
	RMSE	27.51	38.02	**22.71**	35.05
#06	MAE	**2.75**	25.31	4.23	3.03
	RMSE	**3.73**	37.74	5.26	4.24
#07	MAE	5.34	11.13	8.6	**5**
	RMSE	6.89	18.7	11.14	**6.45**
#08	MAE	3.35	39.3	8.11	**2.91**
	RMSE	5.24	71.53	12.8	**4.28**
#09	MAE	9.78	38.94	7.16	**4.1**
	RMSE	13.79	58.78	10.05	**5.85**
#10	MAE	11.9	53.42	16.43	**11.21**
	RMSE	13.51	76.24	18.49	**13**
#11	MAE	12.91	63.92	15.34	**12.74**
	RMSE	15.6	89.19	18.9	**15.31**
Average	MAE	6.09	32.5	8	**5.66**
	RMSE	8.03	51.97	10.68	**7.93**

Table 3: Performance comparison of 3D trajectory reconstructions. This result is cm scale.

the squared differences. Small values indicate high performance for both metrics.

5.2. Effects of 2D drone detectors

For the multi-drone tracker, we trained various drone detectors according to our drone image dataset `2Drone`. We then tested the trackers on 2D synthetic sequences in `Syn3Drone` dataset as summarized in Tab. 2. The detector trained with `2Drone(on)` often failed to detect drone, due to the lack of training dataset. On the other hand, the detector trained by `2Drone(on+aug)` improved drone detection rate significantly and helped the tracker operate properly. In addition, it reduced the drone ID switch cases. These results support that effectiveness of our datasets in training a reliable drone detector and tracker. We anticipate further enhancements of the tracker by fine-tuning it using additional test domain images.

5.3. 3D Trajectory Reconstruction Results

To validate the proposed methods, we employed the `Syn3Drone` and `Real3Drone` datasets. During the experiments, we used ground-truth 2D drone bounding boxes to evaluate the effectiveness of our 3D trajectory reconstruction methods, independent of the performance of the drone detectors. Initially, the reconstructed 3D trajectories of drones may contain errors due to limited information from a single camera setting. To mitigate reconstruction errors, we applied a moving average filter on the initial 3D trajectory. Fig. 6 shows the trajectory smoothing results of Seq. #01 compared to the initial estimation. While a larger window size can provide a smoother trajectory, we empirically observed that a window size of 5 yields the best 3D trajectory reconstruction performance.

To prove the effect of the proposed 2D drone rotation estimation in Fig. 3, we compared three simple approaches for determining 2D drone points $\mathbf{p}_1, \mathbf{p}_2$ as shown in Fig. 7. Tab. 3 presents the 3D trajectory reconstruction performance of the different approaches and our method. The proposed method performed the best average reconstruction in terms of both evaluation metrics (MAE and RMSE). Moreover, the proposed exhibited superior performance in the most complex scenarios (Seq. #08 and #09), including drone rotation and nonlinear motion. Above all, it showed good performance not only for synthetic input but also for real-world sequences (Seq. #10 and #11). With this, our methods can be used application in the real-world. We observed the qualitative trajectory reconstruction results of our methods in Fig. 8. Compared to the ground-truth trajectories, our methods performed reliable and reasonable reconstruction. The results indicate that the proposed method efficiently handles complex scenarios, and demonstrates the potential for applying our methods to standard surveillance systems using a single static camera.

6. CONCLUSIONS

In this work, we proposed a novel framework for reconstructing 3D drone trajectories using a single static camera. To this end, we exploited calibration to leverage the relationship between 2D and 3D spaces and tracked the drones in 2D images based on the drone tracker. For the tracker, we augmented the 2D drone image dataset and trained an accurate 2D drone detector. Furthermore, we proposed a 2D drone rotation estimation method.

By combining the 2D drone rotation information with its actual length, we geometrically inferred the 3D drone trajectories in the camera coordinate system. The experimental results revealed that the proposed methods could perform reliable 3D drone trajectory reconstruction and demonstrated the potential for applying our framework in common surveillance systems using a single static camera.

7. REFERENCES

[1] Tech42, "[rapid growth of the global drone service industry] market outlook, growth drivers and limiting factors," 2021. 1

[2] Nemra Abdelkrim, Nabil Aouf, Antonios Tsourdos, and Brian White, "Robust nonlinear filtering for ins/gps uav localization," in *2008 16th Mediterranean Conference on Control and Automation*. IEEE, 2008, pp. 695–702. 1

[3] Hamid Didari Khamseh Motlagh, Faraz Lotfi, Hamid D Taghirad, and Saeed Bakhshi Germi, "Position estimation for drones based on visual slam and imu in gps-denied environment," in *2019 7th International Conference on Robotics and Mechatronics (ICRoM)*. IEEE, 2019, pp. 120–124. 1

[4] Nikolaos Kyriazis and Antonis Argyros, "Scalable 3d tracking of multiple interacting objects," in *CVPR*, 2014, pp. 3430–3437. 1, 2

[5] Ramin Nabati, Landon Harris, and Hairong Qi, "Cftrack: Center-based radar and camera fusion for 3d multi-object tracking," in *IV Workshops*. IEEE, 2021, pp. 243–248. 1, 2

[6] Meng Zhang, Zhiyu Pan, Jianjiang Feng, and Jie Zhou, "3d multi-object detection and tracking with sparse stationary lidar," in *CVPR*. Springer, 2021, pp. 16–28. 1, 2

[7] Hua-Tsung Chen et al., "Physics-based ball tracking and 3d trajectory reconstruction with applications to shooting location estimation in basketball video," *Journal of Visual Communication and Image Representation*, vol. 20, no. 3, pp. 204–216, 2009. 1, 2

[8] Mandyam V Srinivasan, Hong D Vo, and Ingo Schiffner, "3d reconstruction of bird flight trajectories using a single video camera," *Plos one*, vol. 17, no. 8, pp. e0271618, 2022. 1, 2

[9] Jessica S Horst and Michael C Hout, "The novel object and unusual name (noun) database: A collection of novel images for use in experimental research," *Behavior research methods*, vol. 48, pp. 1393–1409, 2016. 1, 3

[10] Tsung-Yi Lin et al., "Microsoft coco: Common objects in context," in *ECCV*. Springer, 2014, pp. 740–755. 2, 3, 4

[11] Huichao Chen, Zheng Wang, and Linyuan Zhang, "Collaborative spectrum sensing for illegal drone detection: A deep learning-based image classification perspective," *China Communications*, vol. 17, no. 2, pp. 81–92, 2020. 2

[12] Artem Rozantsev et al., "Flight dynamics-based recovery of a uav trajectory using ground cameras," in *CVPR*, 2017, pp. 6030–6039. 2

[13] Jingtong Li et al., "Reconstruction of 3d flight trajectories from ad-hoc camera networks," in *IROS*. IEEE, 2020, pp. 1621–1628. 2

[14] Peixi Peng, Yonghong Tian, Yaowei Wang, Jia Li, and Tiejun Huang, "Robust multiple cameras pedestrian detection with multi-view bayesian network," *Pattern Recognition*, vol. 48, no. 5, pp. 1760–1772, 2015. 2

[15] Jian Yao and Jean-Marc Odobez, "Multi-camera multi-person 3d space tracking with mcmc in surveillance scenarios," in *Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications*, 2008. 2

[16] Jinchang Ren, James Orwell, Graeme A Jones, and Ming Xu, "Tracking the soccer ball using multiple fixed cameras," *Computer Vision and Image Understanding*, vol. 113, no. 5, pp. 633–642, 2009. 2

[17] Wanneng Wu, Min Xu, Qiaokang Liang, Li Mei, and Yu Peng, "Multi-camera 3d ball tracking framework for sports video," *IET Image Processing*, vol. 14, no. 15, pp. 3751–3761, 2020. 2

[18] Caroline Rougier and Jean Meunier, "3d head trajectory using a single camera," in *Image and Signal Processing: 4th International Conference, ICISP 2010*. Springer, 2010, pp. 505–512. 2

[19] Rahul Garg et al., "Learning single camera depth estimation using dual-pixels," in *ICCV*, 2019, pp. 7628–7637. 2

[20] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow, "Digging into self-supervised monocular depth estimation," in *ICCV*, 2019, pp. 3828–3838. 2

[21] Svante Wold, Kim Esbensen, and Paul Geladi, "Principal component analysis," *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1-3, pp. 37–52, 1987. 2, 3

[22] Nicolai Wojke, Alex Bewley, and Dietrich Paulus, "Simple online and realtime tracking with a deep association metric," in *ICIP*. IEEE, 2017, pp. 3645–3649. 3

[23] Yifu Zhang et al., "Bytetrack: Multi-object tracking by associating every detection box," *ECCV*, 2022. 3, 5

[24] Yifu Zhang et al., "Fairmot: On the fairness of detection and re-identification in multiple object tracking," *International Journal of Computer Vision*, vol. 129, no. 11, pp. 3069–3087, 2021. 3

[25] Zheng Ge et al., "Yolox: Exceeding yolo series in 2021," *arXiv:2107.08430*, 2021. 3

[26] Joseph Redmon et al., "You only look once: Unified, real-time object detection," in *CVPR*, 2016, pp. 779–788. 3

[27] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *NIPS*, vol. 28, 2015. 3

[28] Jia Deng et al., "Imagenet: A large-scale hierarchical image database," in *CVPR*. Ieee, 2009, pp. 248–255. 3, 4

[29] "https://www.kaggle.com/datasets/dasmehdixtr/drone-dataset-uav," 2019. 3, 4

[30] Angelo Coluccia et al., "Drone-vs-bird detection challenge at ieee avss2021," in *AVSS*. IEEE, 2021, pp. 1–8. 3, 4

[31] Xuebin Qin et al., "U2-net: Going deeper with nested u-structure for salient object detection," 2020, vol. 106. 3

[32] Lukas Block et al., "Image-bot: Generating synthetic object detection datasets for small and medium-sized manufacturing companies," *Procedia CIRP*, vol. 107, pp. 434–439, 2022. 5

[33] Glenn Jocher, "Yolov5 by ultralytics," 2020. 5

[34] Keni Bernardin and Rainer Stiefelhagen, "Evaluating multiple object tracking performance: the clear mot metrics," *EURASIP Journal on Image and Video Processing*, vol. 2008, pp. 1–10, 2008. 5

[35] Tianfeng Chai et al., "Root mean square error (rmse) or mean absolute error (mae)," *Geoscientific model development discussions*, vol. 7, no. 1, pp. 1525–1534, 2014. 5