Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institut für Geodäsie und
Photogrammetrie

Measuring Drone Trajectory using Total Station with Visual Tracking

Interdisciplinary Project

Yue Pan

yuepan@ethz.ch

Chair of Photogrammetry and Remote Sensing Group
Institute of Geodesy and Photogrammetry
ETH Zürich

Supervisors:
Dr. Cenek Albl, Dr. Jemil Butt, Andreas Baumann-Ouyang
Prof. Dr. Konrad Schindler

February 6, 2021

Acknowledgements

I would like to express my sincere gratitude to those who supported me during this interdisciplinary project:

- **Prof. Dr. Konrad Schindler** for for giving me the opportunity to work on this project as my interdisciplinary project.
- **Dr. Cenek Albl** for the supervision and support of this interdisciplinary project, especially on the project organization, schedule coordination and pipeline design. I learned a lot from his precious advice on data processing, experiment set-up and presentation.
- **Dr. Jemil Butt** and **Andreas Baumann-Ouyang** for the co-supervision of the project, especially on surveying adjustment and total station synchronization.
- **Alexander Wolf** for the assembly and repair of the drone as well as providing the radio synchronization hardware.
- **Thomas Posur** for the surveying instruments and related tutorials.
- **Tom Manu**, **Usvyatsov Mikhail** and **Mudathir Awadaljeed** for helping with data collection.

Abstract

With rapid development, drones are being widely used in industry and our daily life. To keep the sky safe and orderly, the reliable and efficient monitoring and regulating of drones has become a heated issue. To this end, numerous vision-based methods have been proposed in recent years to realize the tracking and trajectory reconstruction of those non-cooperative drones. Unfortunately, up to now, there's hardly a dataset targeting this task for algorithm development and evaluation.

In this project, we fill the gap by presenting a visual drone tracking and localization dataset collected by a multi-sensor system, including a total station, on-board sensor kits, and an ad-hoc network of cameras. The dataset includes six synchronized videos of about 10 minutes, each recorded by a camera in the network. During the period, three drones fly on site within a region of $\approx 100 \times 100$ m, at heights up to $\approx$ 50m. The cameras' position and one of the drones' poses at its body center are also provided as the ground truth. Our dataset is unique and challenging due to the snow-covered and cloudy background as well as the multiple drones tracking scenario.

Another contribution of the project is our pipeline for acquiring and processing the data, mainly focusing on the spatial alignment and joint synchronization of the sophisticated multi-sensor system. The drone in flight is recorded by the camera network simultaneously to capture the visual observations and a prism is attached to the drone for tracking its position accurately with the total station. On-board sensors such as IMU, GNSS antenna, and barometers are fused by an extended Kalman filter to estimate the drone's orientation, which is interpolated to further integrated with the total station's measurement. By calibrating the displacement of the prism to the drone's body center and conducting the resection of the total station afterward, the drone's pose in the local frame can be estimated. According to the sensor specification and covariance propagation, the estimated positioning accuracy is better than one centimeter. As for the joint synchronization, since the central PC can communicate with all the sensors involved in the dataset, the PC's time can be used as the referenced time frame after aligning with UTC. Besides, a radio-synchronized network of audio triggers is adopted to align all the videos' timeline with each other and then convert the frame-wise timestamp into PC's time. The overall synchronization error is at most 10 milliseconds.

The dataset will be publicly available via the link[1] together with four additional less comprehensive data sequences acquired previously in summer. The codes of the proposed pipeline for dataset collection and processing are also available via the link[2].

[1] https://github.com/CenekAlbl/drone-tracking-datasets
[2] https://github.com/YuePanEdward/drone-tracking-toolkits

Contents

Chapter 1

Introduction

1.1 Motivation

The era of the drone is coming. Drones can help us to take photos from where we cannot, transport packages without human labor, monitor the crop production and the soil condition from the sky, keep the power infrastructures safe from the surrounding conditions, as well as to conduct aerial surveying and mapping efficiently. However, drones can also be dangerous if there's no surveillance and regulation on them. Amateur drones out of control can be nightmares for civil airplanes during taking off and landing. In the past decades, several civil aviation accidents are caused by so-called rogue drones. Besides, spy drones are great threats to the military and national security.

In order to cope with these problems and prevent them from happening, we need to better monitor and regulate drones in sensitive areas. The essential operation for drone monitoring is tracking and localization, which has been a heated topic in research for a long time.

Traditionally, there are two solutions to realize the tracking and localization of drones. The first option is the active positional and dead reckoning methods. The localization is realized by onboard localization modules such as GNSS, inertial measurement unit (IMU), and visual odometry (optical flow). The regulator and monitor try to construct the real-time communication with the drone to receive the updated position of the drone. However, it is impossible to communicate with non-cooperative drones, which are quite common nowadays. Even if the communication is available, the absolute localization error might be unbearable in dense urban scenarios where the GNSS's performance deteriorates. The second option is to use stationary ground-based distance and angle measurement devices, which do not rely on communication with the aircraft. Such kind of passive ground-based navigation such as non-directional beacon, very high-frequency omni-range, and instrument landing system has been applied in the aviation community for several decades [1]. Total station positioning (TPS) constituted with a tracking total station and a 360° prism mounted on the drone also belongs to this category. Since the distance measurement accuracy is better than 1 mm + 1 ppm for cutting edge total stations, the highest relative positioning accuracy can be achieved by TPS. However, it's not realistic for non-cooperative drones to be equipped with prisms. Moreover, the initial target searching and locating need to be done manually. Once the total station loses track due to disturbance, occlusion or drastic motion, it can hardly automatically relocate the target.

Recently, the rapid development of computer vision provided us an efficient and low-cost alternative to track the drone and reconstruct its trajectory from several videos recorded from multiple viewpoints surrounding the flight site. Compared with the second option, this vision-based solution is easier to deploy and calibrate. These algorithms take the videos containing the drone as input and output the drone's 3D trajectory relative to the camera network. These methods do not require communication with the drone and expensive positioning instruments.

All they need is an ad-hoc network of cameras. The cameras can be of various types such as smartphones, compact cameras, and GoPro action cameras with arbitrary resolution and frame rate. If high detection precision of the drone and high accuracy reconstruction of the trajectory can be reached, these vision-based approaches would be the ideal solution to the tracking and localization of drones.

To better test and evaluate the vision-based algorithms and have a fair comparison with each other, a unique drone tracking dataset is in need. The dataset should contain the necessary input for the algorithm and the ground truth for the drone detection, tracking, and trajectory reconstruction tasks. The acquisition and procession of such a dataset would be the main focus of our project.

1.2 Problem formulation

The first component of the dataset is the videos of the drone. Several cameras should be set up around the site and kept stationary. Then the drone needs to be maneuvered to fly back and forth above the site, in such a way that the drone is almost always visible in at least two cameras' field of view. Besides, the cameras should also be synchronized so that the frame-wise timestamp of each video can be aligned to a common time frame. This information provides the ground truth for evaluating the algorithm's performance on synchronization and rolling shutter compensation.

The second component of the dataset is the ground truth trajectory (or even pose) of the drone at its body center. Acting as the reference for algorithm comparison, the trajectory's accuracy should be as high as possible, which can be reached by a multi-sensor system mainly based on total station. To be tracked by a total station, a 360° prism needed to be mounted on the drone. The relative coordinate of the prism in the total station's frame can be measured directly. The goal is to estimate the drone's body center's coordinate in the local geo-referenced frame. Therefore, the drone's orientation, the displacement from the prism to the drone's center, and the total station's pose in the local frame need to be measured. With the multi-sensor system, they can be achieved by the onboard sensors such as IMU, lab calibration, and total station resection, respectively. The resulting trajectory is preferred to provided with timestamp synchronized with the videos for finer evaluation.

Besides, for an ideal dataset targeting the aforementioned task, the ground truth position and the intrinsic parameters of the cameras should also be provided. They can be accomplished by total station measurement and on-site calibration with a chessboard, respectively.

In shorts, the problem is formulated as measuring drone trajectory with high accuracy using a multi-sensor system involving total station and onboard IMUs, with a network of cameras recording synchronized videos containing the drone.

1.3 Challenges

The main challenges for constructing the expected dataset are:

- The estimation of drone's pose at its body center using data measured by a multi-sensor system.
- The joint synchronization of the multi-sensor system, including the camera network

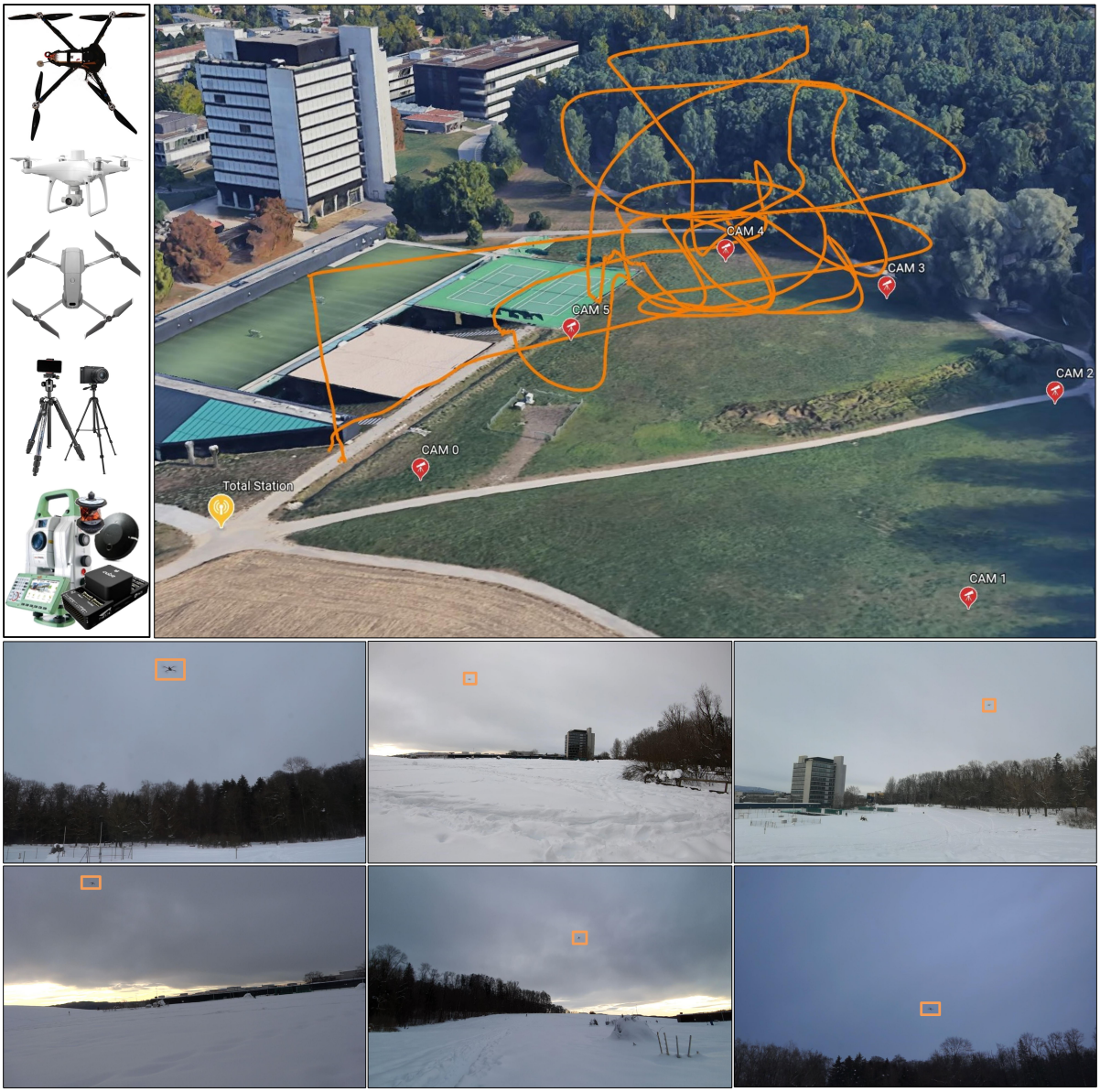


Figure 1.1: An overview of the presented dataset: The dataset contains six synchronized videos containing the three drones recorded by a camera network as well as the ground truth trajectory of the drone and the position of the cameras. The ground truth trajectory is acquired by a multi-sensor system including a total station and on-board sensor kits. This dataset can be used for algorithm development and evaluation on tasks such as visual tracking and localization of drones.

- The simultaneous control and operation of multiple drones, the total station, and the synchronization triggers
- Reliable accuracy evaluation of the estimated drone trajectory

These challenges will all be handled in the following chapters.

1.4 Contributions

The major contributions of the project are:

- We present a dataset for developing and evaluating the vision-based drone tracking and trajectory recovery algorithms. As shown in Fig.1.1, the dataset contains the synchronized videos of multiple drones recorded by an ad-hoc camera network, together with the ground truth pose of the drone as well as the cameras' position and intrinsic parameters.
- The dataset will be publicly available via [1] together with the previous dataset (Cenek 2020 [2]). The complete dataset would be the most comprehensive one with the highest ground truth accuracy.
- We propose a method to measure drone trajectory and orientation using a total station and onboard IMUs with subcentimeter ($<$ 1cm) positioning accuracy.
- We propose a pipeline to synchronize the total station, IMUs, and cameras jointly to achieve 10-millisecond synchronization accuracy via a radio-synchronized audio triggering system.

1.5 Report structure

This chapter introduces the background, motivation, goals, and challenges of the project. The remainder of this report is structured as follows.

Chapter 2 summarizes previous work on the topics of tracking and positioning of drones using computer vision algorithms and total station. The existing datasets on this topic are also reviewed. Chapter 3 describes the key instruments and softwares used for data acquisition. Chapter 4 presents the methodology and principles for estimating the drone's pose using a multi-sensor system and conducting the joint synchronization of all the data sources. Chapter 5 introduced the pipeline and implementation details of data acquisition and processing. Chapter 6 demonstrates the generated dataset and describes the dataset structure and format. Finally, the conclusions and outlooks are given in Chapter 7.

[1] https://github.com/CenekAlbl/drone-tracking-datasets

Chapter 2

Related Work

In this chapter, we will firstly review the vision-based drone trajectory reconstruction systems proposed in recent years. Then the available datasets targeting this task and their limitation would be introduced. After that, some earlier trials on drone trajectory measurement via total station are also reviewed.

2.1 Reconstruct drone trajectory via visual tracking

As shown in Fig.2.1, motion capture systems such as Vicon[1] has been widely utilized in computer vision and robotics research [3]. The high accuracy localization estimated by Vicon is often used as the ground truth for evaluating the visual odometry and sensor fusion algorithms for drones and other robots. The typical application scenario is an indoor lab setup, where several low-latency cameras are mounted on the wall surrounding the lab and connected to a

[1] https://www.vicon.com/

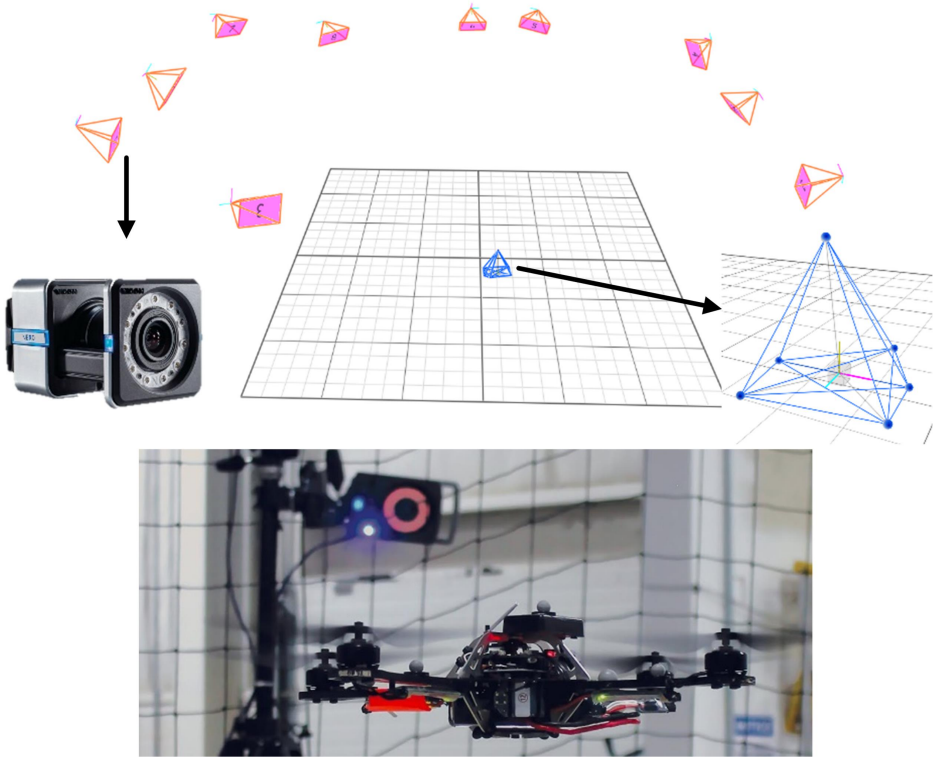


Figure 2.1: Vicon indoor motion capture system

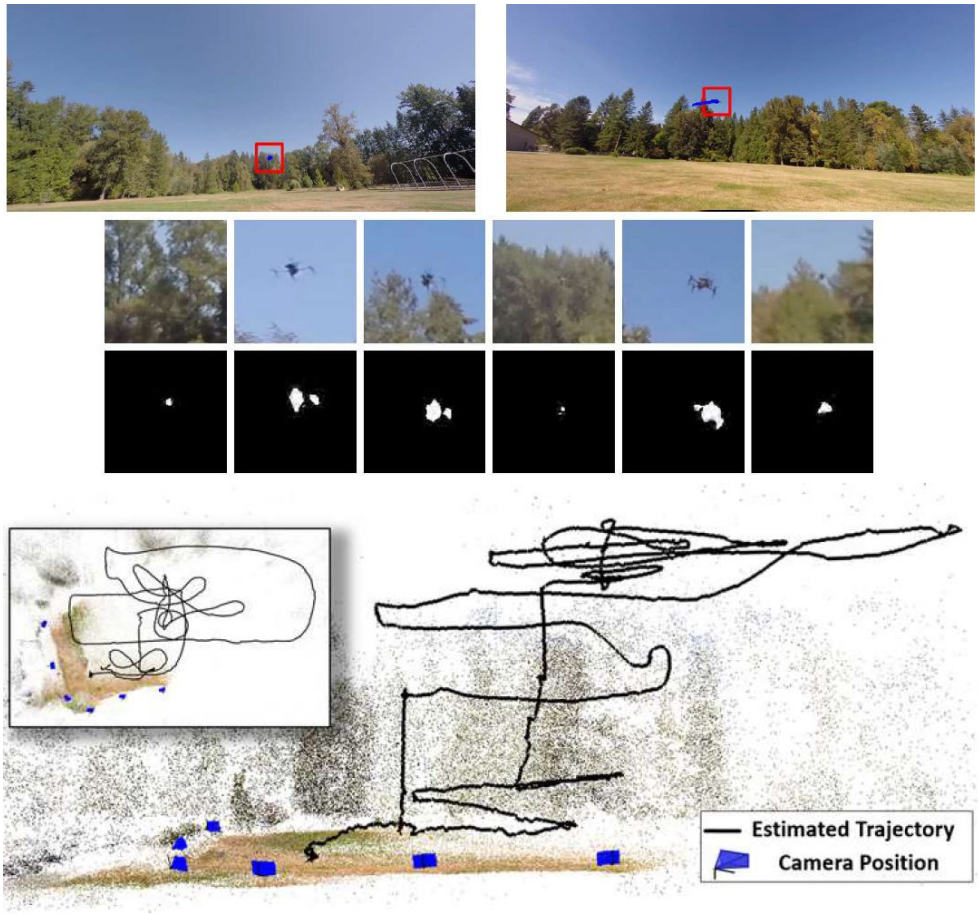


Figure 2.2: Overview of the vision-based drone trajectory recovery algorithm proposed in [4].

host computer. A few markers are attached to the rigid-body object such as a drone. The camera poses and scale can be calibrated beforehand. The 3D trajectory of the moving object with markers can be determined by the camera network with millimeter accuracy and 100 Hz frequency. However, such kind of motion capture systems cannot be set up in a larger-sized scene outdoor. Besides, the need for special markers attaching to the target object makes it impossible to conduct the tracking and trajectory recovery of non-cooperative drones outdoor.

In 2017, [4] proposed a novel method to estimate the drone trajectory within a outdoor site using multiple stationary ground cameras, as shown in Fig.2.2. The pipeline is based on a structure from motion framework to reconstruct a moving point with prior defined motion dynamics. The method includes the single-view detection and tracking of the drone as well as a bundle adjustment procedure regularized by the motion dynamics. By taking the SBAS GNSS positioning results (accuracy > 0.5 m) as the reference, the proposed system's trajectory recovery accuracy is about one meter.

In 2020, [2] further generalize the improve [4] by considering the synchronization among the cameras and the rolling shutter effect. The input of the proposed system is the unsynchronized videos containing the drone captured by different cameras surrounding the site. The output is the 3D trajectory of the drone with the by-product of camera pose and synchronization parameters. Procedures such as structure from motion and spatial-temporal bundle adjustment under certain motion dynamics are adopted in the pipeline. By taking the onboard differential GNSS's positioning result (accuracy ≈ 2 cm) as the reference, the positioning error of the proposed method is several decimeters.

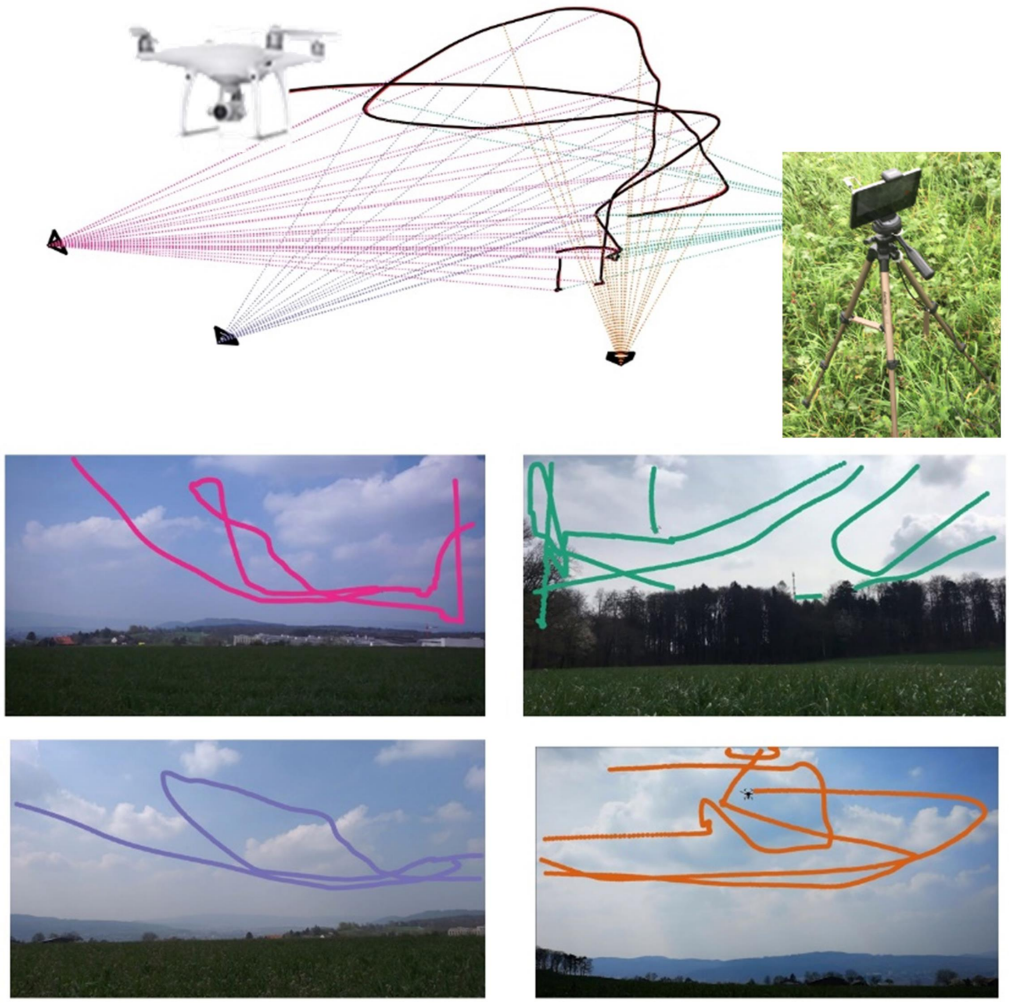


Figure 2.3: Overview of the vision-based drone trajectory recovery algorithm proposed in [2].

2.2 Datasets for visual tracking and localization of drones

To implement and evaluate the aforementioned methods, a dataset is in need. The expected dataset should include the videos as the algorithm's input. It should also consist of an accurate measurement of the drone's trajectory, which can be regarded as the ground truth. Since the quantity of the state-of-the-art algorithm's accuracy is tens of centimeters, the accuracy of ground truth measurement is expected to be better than one centimeter. What's more, the timestamps of all the data should ideally register under a unified time system. Because a time delay of one second is corresponding to a positioning error of several meters for a drone moving at moderate speed.

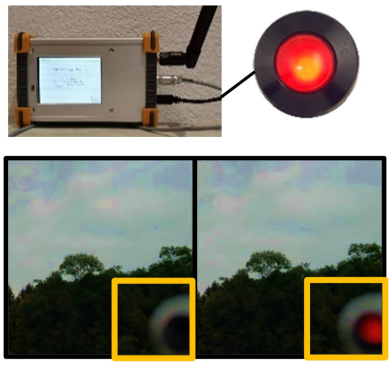
Several vision-based drone detection and tracking datasets such as [5], [6] and Anti-UAV[3] have been published in recent years. They can be utilized to test and evaluate the single-view detection and tracking algorithms targeted on drones, which is often the first and fundamental step of the drone trajectory recovery pipelines. However, it is found that there's almost no public dataset targeting the specific 3D reconstruction part of the task. The only two datasets are provided together with the proposed approach [4] and [6] introduced above .

[3] https://anti-uav.github.io/dataset/

Dataset	(Rozantsev 2017) [4]	(Cenek 2020) [2]	AntiUAV[2]	Ours
# cameras	6	7	1	6
durations [s]	500	1500	-	800
2D tracking	✓	✓	✓	✓
multiple drones	✗	✗	✓	✓
3D trajectory	✓	✓	✗	✓
accuracy [cm]	50	2	-	**1**
3D orientation	✗	✗	✗	✓
camera position	✗	partial ✓	✗	✓
accuracy [cm]	-	<5	-	**3**
synchronization	✗	partial ✓	✗	✓

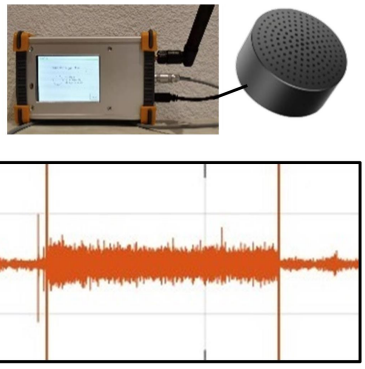
Table 2.1: Comparison of the properties of our proposed new dataset with existing datasets on the tasks of visual tracking and passive positioning of drones.

(a) Fixposition RTK as the ground truth positioning module



(b) Synchronization of camera network based on LED flash

(c) Total station and on-board sensors constitute the ground truth positioning module



(d) Synchronization of camera network based on audio triggering

Figure 2.4: Previous dataset (Cenek 2020) based on (a) and (b) [2] v.s. the proposed new dataset based on (c) and (d)

In the dataset collected and used by [4], the ground truth drone trajectory is obtained by SBAS GNSS instead of differential GNSS. Its accuracy is at best 0.5 m and the specific value is unknown.

In [2], a dataset meeting the minimum requirement is collected. In this dataset, there are altogether four data sequences with different properties. Sequences 1 and 2 stand for the simple cases, where a single drone flies with moderate speed and changes direction smoothly. Sequences 3 and 4 are much challenging with the higher speed and aggressive motion of the drone. In sequence 4, the fast-moving clouds in the background cast challenge on stable tracking of the drone. In the dataset, a single drone flies in a region of about 100×100m at height up to 50m. The number of cameras in the network is up to seven. The ground truth 3D trajectory is acquired by a FixPosition Nav-RTK box[4] mounting close to the drone's body center. The manufacturer claims that the RTK box can reach 2cm localization accuracy. However, the RTK box only outputs the 3D positioning results without the drone's orientation as well as the raw measurements and the detailed accuracy evaluation, thus making it a black box for us. The synchronization among the videos and the position measurement of the cameras are only done in sequence 3. The ground truth camera synchronization is realized by a radio-synchronized network of LEDs. In the field of view of each camera, a LED is placed. The periodical synchronous LED flashes are visible in all the videos. By aligning the timestamp of each flash incident with each other, the synchronization parameters, namely the drift and scale, can be calculated. However, the synchronization resolution is limited to the frame duration and the co-synchronization of the RTK box and the camera network is not accomplished. The camera position is measured directly by a survey-grade GNSS receiver, whose positioning accuracy is about 1 cm. Since it's hard to make the GNSS antenna's phase center coincide with the projection center of each camera, we estimate the accuracy of the cameras' ground truth position to be better than 5cm.

To make up for the deficiency of the previous dataset and achieve a reliable, accurate, and synchronized dataset, we plan to use a multi-sensor system containing a total station and onboard sensor kits. With the total station, it's possible to get sub-mm ranging accuracy. With the onboard IMU, the orientation of the drone can also be measured. What's more, in the new dataset, we replace the visual signal (LED light on or off) with the audio signal as the media for synchronization. By doing so, sub-frame synchronization would be available and there's no longer the LED's red flash disturbance to the video. The joint synchronization of the total station, onboard sensors, and the synchronized camera network are realized by registering all the measurements to a common time frame. The comparison of the proposed new dataset with existing ones is shown in Table 2.1. The main improvements of the new dataset over the previous one (Cenek 2020) [2] are shown in Fig.2.4. It is shown that the new dataset would be the most comprehensive one with the highest ground truth accuracy. It can also serve for tasks like multiple object tracking and visual orientation estimation of the drones.

2.3 Measure drone trajectory by total station

Though the tracking function of total station is widely applied in engineering geodesy, there's only a few studies in literature on tracking aircrafts by total station.

To evaluate the accuracy of the direct geo-referencing of a drone based on differential GNSS, in 2011, [7] firstly try to use the tracking measurements of total station as the reference. A 360 ° prism is attached to a octocopter flying outdoor and is successfully tracked by a Leica TPS-1200 total station, as shown in Fig.2.5(a). This work is followed by [8], in which the prism

[4] https://www.fixposition.com/

is mounted on a larger helicopter. Later in 2017, [9] measure the trajectory of a non-cooperative aircraft using total station and CCD cameras without mounting prism on the aircraft, as shown in Fig.2.5(b).

To summarize, using a total station to do the tracking and measure drone trajectory is accurate but quite expensive. To guarantee the highest accuracy, a 360 ° prism needed to be mounted on the target object. The deployment is relatively complicated since the resection of the total station is often compulsory. The final positioning accuracy depends on the tracking status, the specification of the total station as well as the displacement of the prism from the target origin such as the body center.

(a) Measure the trajectory of a octocopter using total station [7]

(b) High accuracy remote determination of trajectories of non-cooperative aircraft by total station [9]

Figure 2.5: Related works on drone trajectory measuring by total station

Chapter 3

Instruments

In this chapter, we introduce the key instruments used for acquisition of the presented dataset. Fig.3.1 and Fig.3.2 demonstrate an overview of the used instruments.

3.1 Survey instrument

As shown in Fig.3.3, the surveying instruments used in the project are a total station, a survey-grade GNSS antenna and the prisms.

3.1.1 Total station

The total station used by us is Leica Nova TS-60[1]. It's one of the world's most accurate total stations with sub-second angular accuracy of 0.5″ and sub-millimeter ranging accuracy of 0.6mm + 1ppm. It is also quite robust to harsh conditions such as rain, fog, dust, and reflections. It's also a type of tracking total station [10] with automatic target recognition (ATR) function and a maximum motorized rotation speed of 45° per second. The tracking sampling rate can reach 8 to 10 Hz with IR tracking mode targeting a Leica 360 ° (mini) prism.

In order to monitor and control the total station via a computer and make the operation of the total station programmable, GeoCOM [11] communication protocol is applied. GeoCOM provides normal function call interfaces for Matlab to remote functions such as taking angle measurements, changing face and acquiring current timestamp under the total station time system. These interfaces enable a programmer to implement an application as if it would be carried out manually on the total station. As shown in Fig.3.4, GeoCOM is implemented as a client-to-server communication system. One communication takes place when the client (computer) sends a request to the server (total station) and the server sends a reply back to the client. For example, the request is taking a ranging measurement and the reply is the measured distance. The medium of communication is a serial communication line between the total station and the computer.

In our application, we can set the tracking mode, the prism type, the distance measurement mode, and the angle measurement tolerance via GeoCOM as the initialization step. Then we manually target the telescope to the prism or wait for the ATR function to find the prism with a traverse search. Once the prism is locked by the total station, the function *BAP_MeasDistanceAngle* is called via GeoCOM in a loop. For each call of the function, the angles and distance measurements are executed and the prism's polar coordinates relative to the total station, namely the distance, horizontal angle, and vertical angle are returned. Besides, the tracking status indicator would also be returned.

[1] https://leica-geosystems.com/products/total-stations/robotic-total-stations/leica-nova-ts60

Figure 3.1: Overview of all the instruments used for collecting the dataset

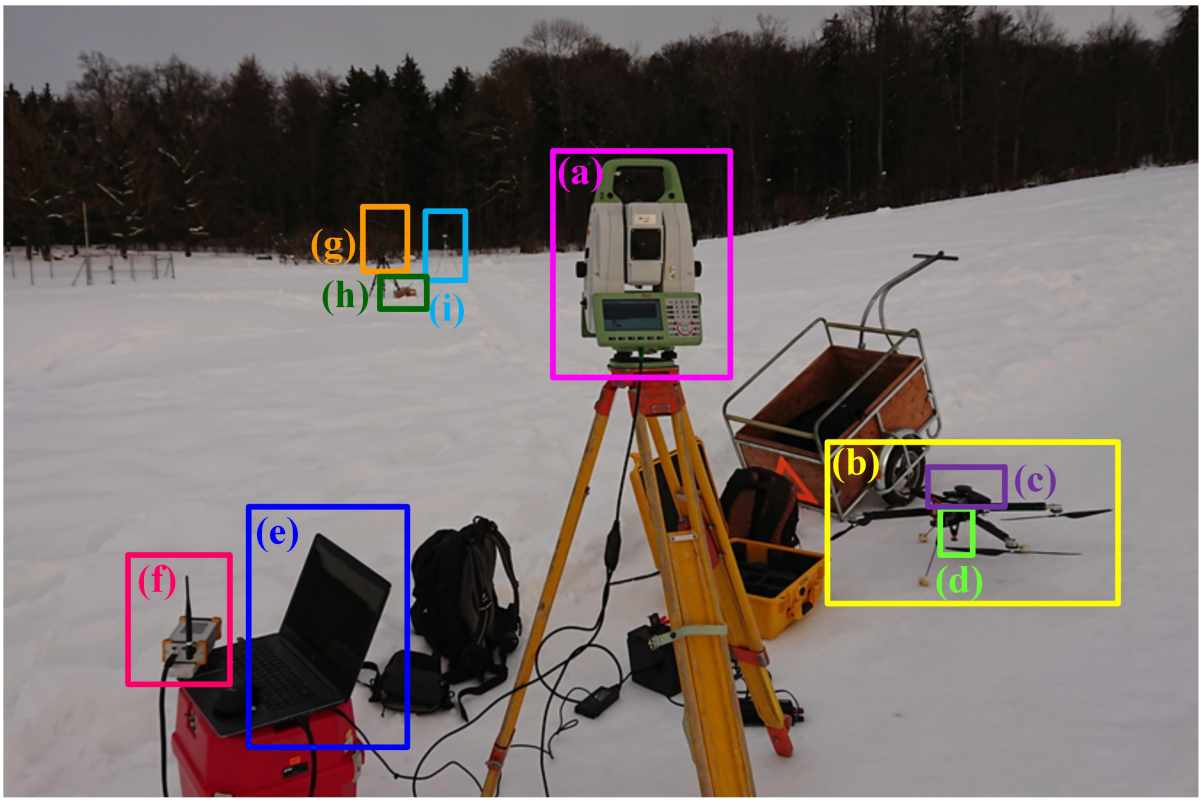


Figure 3.2: Key instruments for collecting the dataset on site: (a) Leica Nova TS 60 Total station connected to PC via GeoCOM, (b) Quadcopter drone, (c) on-borad Pixhawk sensor module, including IMU and GNSS antenna, (d) Leica mini 360° prism mounted beneath the drone, (e) PC, (f) radio-synchronized audio trigger connected to PC, (g) a camera on tripod as one part of the visual tracking camera network, (h) radio-synchronized audio trigger mounted close to the camera, (i) Trimble GPS with SwiPos service, mounted above a Leica 360° prism.

(a) Leica Nova TS60 total station and 360 ° mini prism for drone tracking purpose

(b) Trimble R8 GNSS system and 360 ° prism for resection purpose

Figure 3.3: Total station and survey-grade GNSS instruments

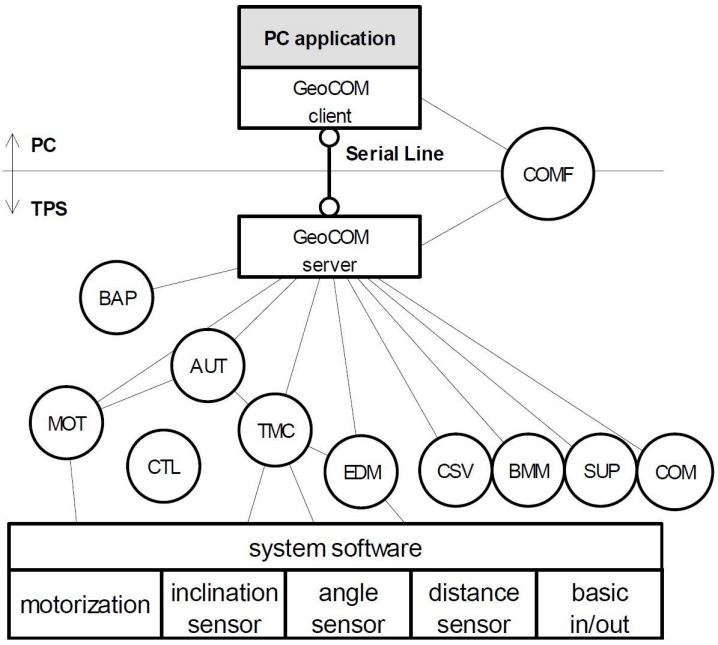


Figure 3.4: Overview of GeoCOM's client/server application [11]

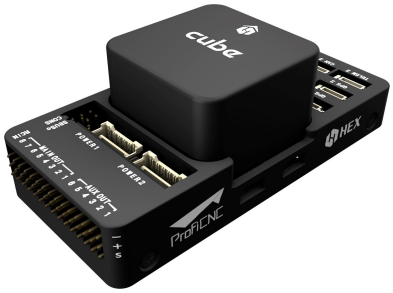
3.1.2 GNSS receiver

For total station resection, the geodetic coordinates of several control points needed to be measured. In this project, We use Trimble R8 integrated GNSS system[2] with SwiPos CORS service[3] to acquire single-point measurements with a horizontal accuracy of 8 mm and a vertical accuracy of 15 mm. In practice, the acquired coordinates are in LV 95 frame, which needed to be transformed to the WGS84 frame for our application. The transformation can be realized by NAVREF[4].

[2] https://geospatial.trimble.com/products-and-solutions/trimble-r8s
[3] https://shop.swipos.ch/
[4] https://www.swisstopo.admin.ch/en/maps-data-online/calculation-services/navref.html

3.2 Onboard sensor kits

(a) Quadcopter drone



(b) Pixhawk cube

(c) Here GNSS antenna

(d) Leica mini 360° prism

Figure 3.5: Drone related hardwares: (b) - (d) are mounted on the drone (a).

In our project, the PX4 open hardware platform[5] is used as the onboard sensor kits as well as the processing and control unit of the quadcopter drone [12], as shown in Fig.3.5.

PX4, originally known as Pixhawk, uses multiple sensors to estimate the drone's states such as pose and velocity, which are needed for self-stabilization and autonomous control. The system minimally requires a gyroscope, accelerometer, magnetometer (compass), and barometer. A GPS receiver and antenna are also needed to enable all automatic modes such as the so-called return to launch (RTL) landing mode. In our case, the Hex Here GNSS antenna is adopted.

The Pixhawk standard autopilot used by us is the Hex cube black flight controller[6], as shown in Fig.3.5(b). The multiple sensors embedded inside the cube are listed as follows:

- LSM303D integrated accelerometer + magnetometer.
- L3GD20 gyroscope
- MPU9250 IMU (gyroscope + accelerometer)
- MS5611 barometer

The sensor calibration and parameter setting for the sensor fusion algorithms can be realized via the ArduPilot MissionPlanner software[7] with the connection to the Pixhawk cube fixed on the drone.

[5] https://px4.io/
[6] https://docs.px4.io/master/en/flight_controller/pixhawk-2.html
[7] https://ardupilot.org/copter/index.html

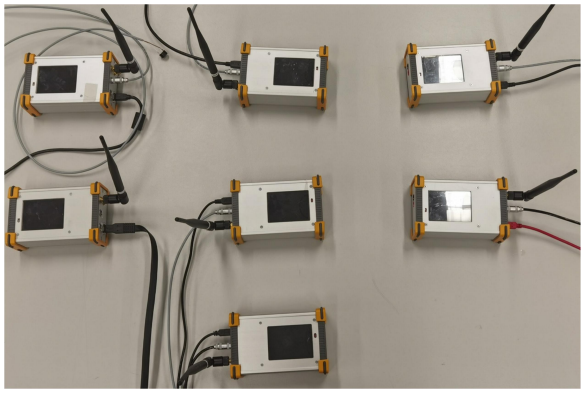
3.3 Other instruments

3.3.1 Cameras

Three types of cameras, namely smartphones, compact cameras and GoPro action cameras are adopted in the project. For each camera, a tripod is prepared.

3.3.2 Radio-synchronized audio triggering system

To enable the joint synchronization among the cameras and the computer, a radio-synchronized audio triggering system is employed. A shown in Fig.3.6, the radio-synchronized system is composed with several radio transceiver boxes. These boxes can transmit and receive signals with each other via radio communication. Besides, an audio speaker is connected to each box and its state (on or off) can be controlled by the received signal of the box. In other words, the signal acts as the trigger of the audio speaker.



(a) radio transceiver boxes

(b) Audio speaker connected to the box

Figure 3.6: Raido-synchronized aduio triggering system

Chapter 4

Methodology

In this chapter, we firstly introduce the principle of drone pose determination and then describe the joint synchronization of the system.

4.1 Pose estimating

4.1.1 Spatial Frames

There are four key frames (coordinate system) for the task, namely the body frame, the total station frame, the local frame and the world frame[13], as shown in Fig.4.1. The details are described as follows:

- Body frame $b-frame$: This frame is fixed with the drone and the onboard IMU. The aligned center of the onboard IMU is regarded as the drone's body center and the origin of the body frame. x, y and z axis are aligned with the corresponding axis of the IMU. With the rotation around x , y and z axis by roll, pitch and yaw angle, the body frame can be aligned with the body-carried north-east-down (NED) frame.
- Total station frame $t-frame$: This frame is fixed with the total station. The origin and axis are aligned with the Cartesian coordinate system for total station measurements.
- Local frame $l-frame$: This frame is also known as navigation or ground coordinate system. It is a coordinate frame fixed to the earth's surface. Its origin is an arbitrary referenced point on the earth's surface. Two types of local frames, namely the NED and the east-north-up (ENU) frame can be defined, according to different directions of the axis aligned with the WGS84 ellipsoid's north, east and normal direction. All the 3D coordinates provided in the presented dataset are in local ENU frame.
- World frame $w-frame$: The WGS84 geodetic coordinate system.

4.1.2 Drone center position estimation

As shown in Fig.4.1, after leveling the total station, the resection is conducted to get the transformation from total station frame $t-frame$ to the local ENU frame $l^{enu}-frame$ since the GNSS measurments in WGS84 frame $w-frame$ can be converted to $l^{enu}-frame$ by providing the WGS84 coordinates of the rerferenced point (local frame origin).

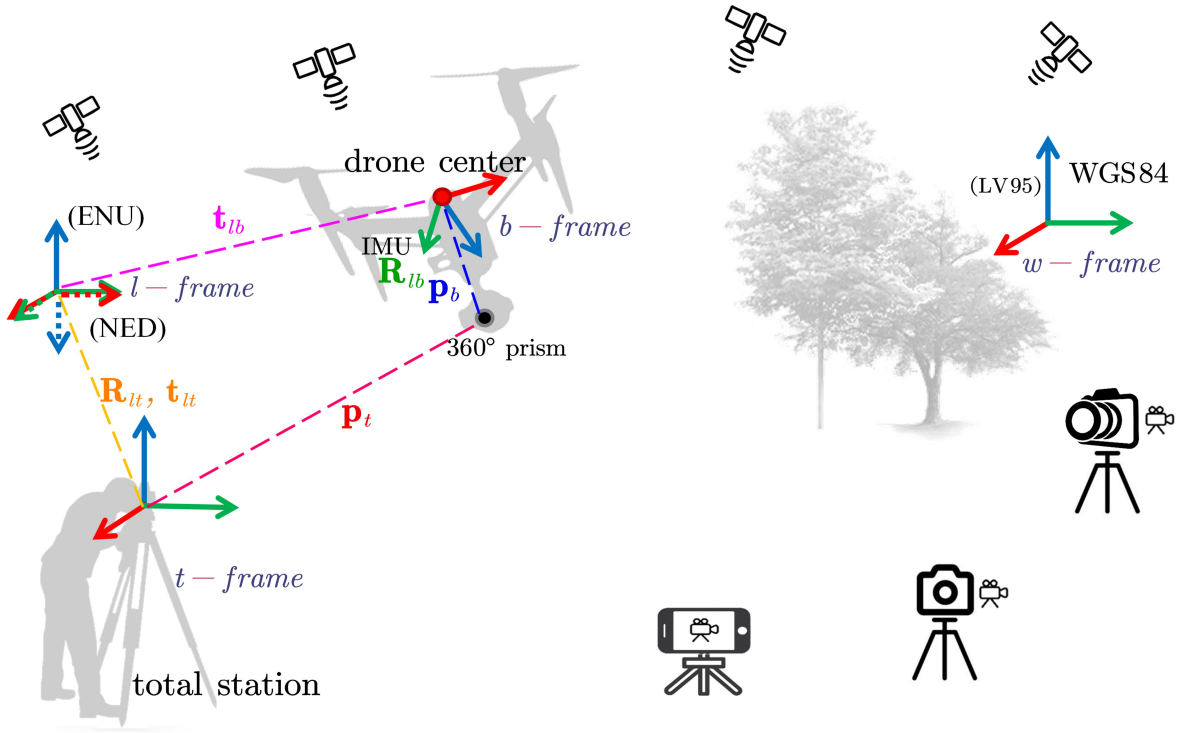


Figure 4.1: The spatial frames and the geometry relationships for drone pose estimation

The total station's measurements of the prism can be converted to Cartesian coordinates from polar coordinates representations ρ, θ and α as:

$$\mathbf{p}_t = \begin{bmatrix} \rho cos\theta cos\alpha \\ \rho cos\theta cos\alpha \\ \rho sin\alpha \end{bmatrix}, \tag{4.1}$$

which is also the prism's coordinate in total station frame $t-frame$. $\mathbf{p}_b$ is the prism's coordinate in the drone's body frame $b-frame$, it can be measured by lab calibration. $\mathbf{t}_{lb}$ is the drone centre's position in local ENU frame $l-frame$, which is what we aim to estimate here. $\mathbf{R}_{lb}$ is the drone's orientation, which can be estimated using the measurements of onboard IMU and other sensors as described in 4.1.3. Our goal is to estimate $\mathbf{t}_{lb}$. With resection and total station tracking, we get the following equation:

$$\mathbf{p}_l^{(enu)} = \mathbf{R}_{lt}\mathbf{p}_t + \mathbf{t}_{lt}. \tag{4.2}$$

Similarly, we get:

$$\mathbf{p}_l^{(ned)} = \mathbf{R}_{lb}\mathbf{p}_b + \mathbf{t}_{lb}. \tag{4.3}$$

Then by applying a NED to ENU transformation of $\mathbf{p}_l$, we get:

$$\mathbf{R}_{lt}\mathbf{p}_t + \mathbf{t}_{lt} = \mathbf{T}_{ned}^{enu}\left(\mathbf{R}_{lb}\mathbf{p}_b + \mathbf{t}_{lb}^{(ned)}\right), \tag{4.4}$$

which can be arranged as:

$$\mathbf{t}_{lb}^{(enu)} = \mathbf{T}_{ned}^{enu}\mathbf{t}_{lb}^{(ned)} = \mathbf{R}_{lt}\mathbf{p}_t + \mathbf{t}_{lt} - \mathbf{T}_{ned}^{enu}\mathbf{R}_{lb}\mathbf{p}_b, \tag{4.5}$$

in which

$$\mathbf{T}_{\text{ned}}^{\text{enu}} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix}. \tag{4.6}$$

4.1.3 Orientation estimation via EKF and interpolation

We define a 24-dimensional state vector, constituted by quaternions, velocity, position, gyro delta angle bias, accelerometer delta velocity bias, earth magnetic field vector, magnetometer bias and the air velocity according to [12] for EKF. Within the onboard sensor kit, the measurements by IMU, GNSS antenna, barometer, magnetometer and airspeed sensor are fed into EKF framework[1] as observations. Therefore, the orientation (represented by roll, pitch and yaw angle or rotation matrix $\mathbf{R}_{lb}$) of the drone and its covariance matirx can be estimated for each Kalman time step.

Since the total station takes measurements with varying time intervals, the timestamps of total station measurements and orientation estimations are not likely to coincide, even with perfect synchronization. Therefore, to keep the most accurate component in the system, namely the total station's measurement, unchanging in data fusion, the orientation estimations are interpolated at the total station's measuring timestamp [14]. Spherical linear interpolation (SLERP) is applied on two temporal adjacent orientation estimations, represented by quaternions. The output orientations of this function are corresponding one on one to the total station tracking measurements.

4.1.4 Positioning accuracy evaluation

The evaluation of the estimated drone center position's accuracy is necessary for verifying the feasibility of the multi-sensor system. The positioning accuracy can be expressed with a covariance matrix. From Eq.4.5, we can represent the target covariance $cov\left(\mathbf{t}_{lb}\right)$ with two parts as:

$$cov\left(\mathbf{t}_{lb}\right) = cov\left(\mathbf{p}_{l}^{'}\right) + cov\left(\mathbf{p}_{b}^{'}\right). \tag{4.7}$$

Each part can be regarded as the covariance of a spatial vector transformed by a rigid-body transformation, in which the both the origin vector and the the transformation's covariance matrices are known. According to appendix A [15], the two parts can be represented as:

$$cov\left(\mathbf{p}_{l}^{'}\right) = \mathbf{J}_{lt} cov\left(\xi_{lt}\right) \mathbf{J}_{lt}^{\top} + \mathbf{R}_{lt} cov\left(\mathbf{p}_{t}^{cart}\right) \mathbf{R}_{lt}^{\top} \tag{4.8}$$

and

$$cov\left(\mathbf{p}_{b}^{'}\right) = \mathbf{J}_{lb} cov\left(\xi_{lb}\right) \mathbf{J}_{lb}^{\top} + \mathbf{R}_{lb} cov\left(\mathbf{p}_{b}\right) \mathbf{R}_{lb}^{\top}, \tag{4.9}$$

where the Jacobian matrices are in the form of Eq.A.4 and the covariance matrix of total station measurements in Cartesian coordinates can be calculated from the covariance in polar coordinates representation as:

$$cov\left(\mathbf{p}_{t}^{cart}\right) = \mathbf{J}_{\rho\theta\alpha}^{xyz} cov\left(\mathbf{p}_{t}^{polar}\right) \mathbf{J}_{\rho\theta\alpha}^{xyz\intercal}, \tag{4.10}$$

in which $cov\left(\mathbf{p}_{t}^{polar}\right)$ can be determined from the total station's specification (the ranging measurement accuracy σ_{d}^{m} m + σ_{d}^{ppm} ppm and the angle measurement σ_{a} rad) as:

$$cov\left(\mathbf{p}_{t}^{polar}\right) = \begin{bmatrix} \sigma_{d}^{m} + \rho\sigma_{d}^{ppm} & 0 & 0 \\ 0 & \sigma_{a} & 0 \\ 0 & 0 & \sigma_{a} \end{bmatrix}, \tag{4.11}$$

and the Jacobian is:

$$\mathbf{J}_{\rho\theta\alpha}^{xyz} = \begin{bmatrix} sin\theta cos\alpha & \rho cos\theta cos\alpha & -\rho sin\theta sin\alpha \\ cos\theta cos\alpha & -\rho sin\theta cos\alpha & -\rho cos\theta sin\alpha \\ sin\alpha & 0 & \rho cos\alpha \end{bmatrix}. \tag{4.12}$$

[1] https://github.com/PX4/PX4-ECL

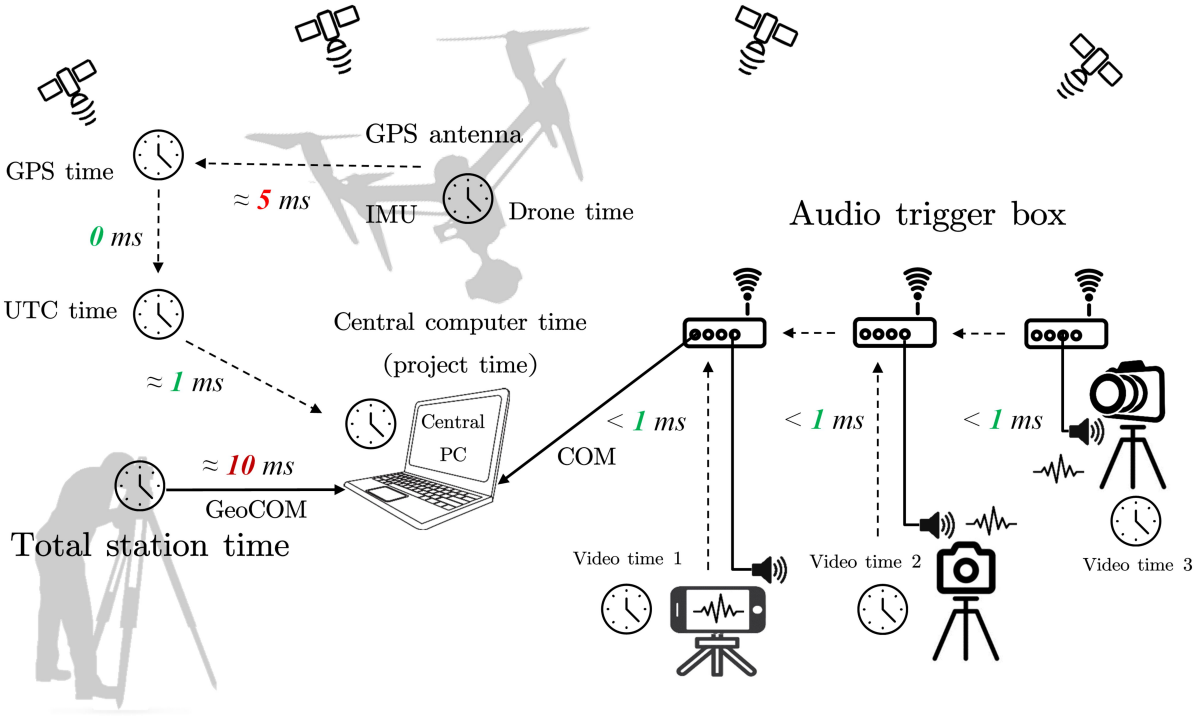


Figure 4.2: Overview of the time systems and the pipeline for the joint synchronization of the multi-sensor system. The number in color represents for the approximate synchronization error.

4.2 Joint synchronization

4.2.1 Time systems

In this multi-sensor system, there are six kinds of time systems used by different instruments, as shown in Fig.4.2. The details are described as follows:

- Central computer time: the time system used by the central computer, which is connected to the total station and the radio-synchronized system. It can be synchronized with Coordinated Universal Time (UTC) by windows online time service[2].
- Project referenced time: By accumulating seconds from the first total station measurements, we then get the project referenced time. All the timestamps provided in the presented dataset would lay under this time system.
- Video time: the time system based on the video recorded by each camera, taking the beginning of the video as this video time system's origin.
- Drone time: the time system used for all the onboard sensor measurements. All the data recorded by the drone is provided with a timestamp under the drone time system.
- GPS time: time system used by GPS, only the onboard GNSS measurements records timestamp under GPS time system. GPS time can be converted to UTC by adding the leap seconds[3].
- Total station time: time system used by the total station's internal computer.

[2] https://docs.microsoft.com/en-us/windows-server/networking/windows-time-service
[3] https://www.iers.org/IERS/EN/Science/EarthRotation/EarthRotation.html

4.2.2 Synchronization among cameras and central computer

Fig.4.3 shows how the radio-synchronized audio trigger system works. One radio transceiver box is connected to the central computer and the others are placed close to each camera. The radio transceiver box can receive a serial signal (simply 1 or 0 for on or off) from the computer through the COM USB connection and then transmit the signal to all the other boxes via radio communication. The computer sends a triggering command to the radio transceiver box connected to it. The box shares the command with all the other boxes via radio communication. In an open outdoor scenario, the radio transmitting distance can reach several hundred meters and there's almost no latency. Besides, we connect one audio speaker to each box. The speakers are attached to each camera's microphone hole. Once the trigger is switched on or off by the computer, all the radio transceiver boxes would get the signal to turn on or turn off the audio speaker, so that the beeping sound made by the speaker is recorded by the camera during shooting the video. To distinguish the beeping from noise and the human voice, a specific triggering pattern with a fixed frequency is used. After filtering the audio signal extracted from the video recorded by each camera with a bandwidth filter, the occurrence of the triggering pulse can be pinpointed on the timeline. By aligning the timestamp of the triggering pulse for each camera with the timestamp of the triggering command in central computer time, the synchronization among the cameras and the central computer can be accomplished.

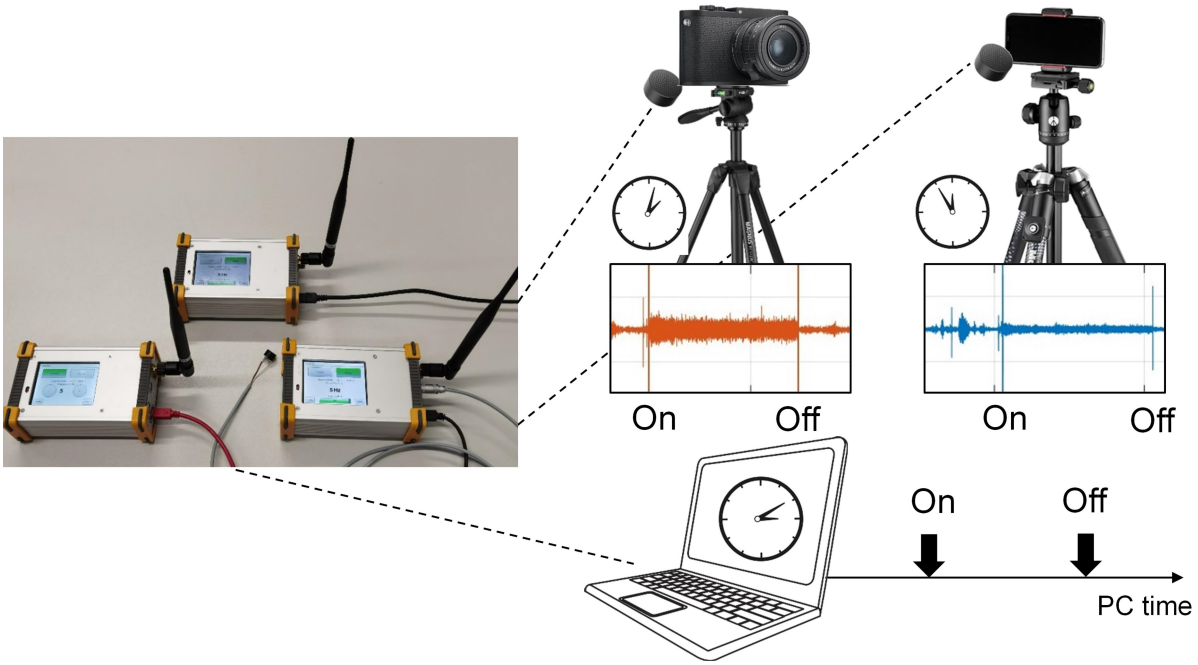


Figure 4.3: Sketch map of the radio-synchronized audio triggering system

4.2.3 Synchronization between onboard sensors and central computer

As shown in Fig.4.2, the synchronization is accomplished by taking the GNSS measurements as the bridge. Onboard GPS record both the GPS time and the drone time, so that the time shift between GPS time and drone time can be calculated. Since GPS time can be converted to UTC and the central computer time is also synchronized with UTC, onboard sensor measurements can all be converted to central computer time. The synchronization error is about five milliseconds.

4.2.4 Synchronization between total station and central computer

The total station is controlled by the central computer via GeoCOM, thus making it possible to synchronize between total station and central computer as well as pinpoint of the exact measuring timestamp.

According to [16], each measurement period of the total station can be divided into four parts, as shown in Fig.4.4. The serial signal transfer time can be estimated as:

$$t_{\text{trans.}} = \frac{m_{\text{char}} n_{\text{bits}}^{\text{char.}}}{\text{baudrate}}. \tag{4.13}$$

The bit number for the forward and back serial signal can be counted so that the forward and back transfer duration can also be estimated. A method is proposed in [16] on calibrating the dead time of measurement. Since the central computer timestamps of the beginning and the end of the measurement period can be recorded by the computer, it's possible to roughly estimate the exact measurement time as the middle of $t_{meas.}$ in Fig.4.4 by subtracting the transfer time and dead time from the single measurement duration. In practice, the estimated synchronization error is about 10 milliseconds, which is the largest among the joint synchronization system.

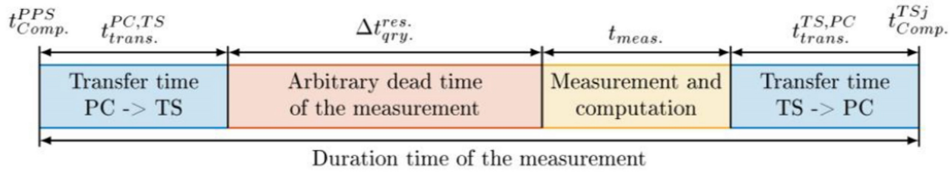


Figure 4.4: Division of the duration time of a single measurement by the total station [16]

Chapter 5

Experiments

In this chapter, we describe the experiments for collecting the presented dataset in detail. Data processing results would be shown in this chapter as well.

As shown in Fig.5.1, the experiment site locates in the peripheral area of the ETH Honggerberg campus. The experiment is conducted in winter after heavy snow so the site is covered by snow, as shown in Fig.5.2. During the experiment, it's cloudy with a temperature of about 0 ℃.

Totally 6 cameras and 3 drones are used for acquiring the dataset. The drones are flown manually with the loiter mode, within a region of about 100×100m, at heights up to about 50m. The mean and maximum velocity is about 12 km/h and 40 km/h, respectively. The flight duration is about 12 minutes. The experiment instruments are described in detail in chapter 3 and the pipeline of the experiment is introduced in chapter 4.

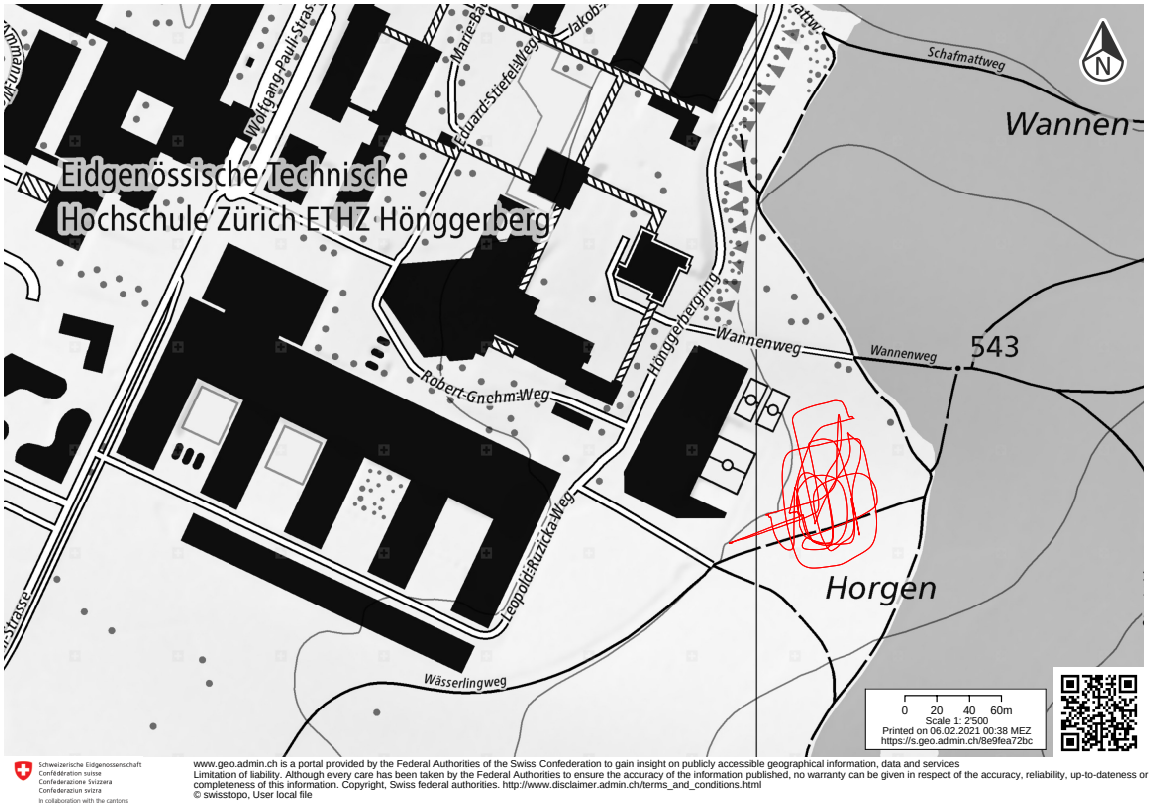


Figure 5.1: Map of the experiment site: the red line denotes the Pixhawk drone's trajectory.

Figure 5.2: Photo of the snow-covered experiment site

5.1 Total station tracking

The total station is set up on site, leveled and targeted to the 360 ° prism on the Pixhawk drone. Then the tracking mode is switched on via GeoCOM. The tracking results during the flight are shown in Fig.5.3.

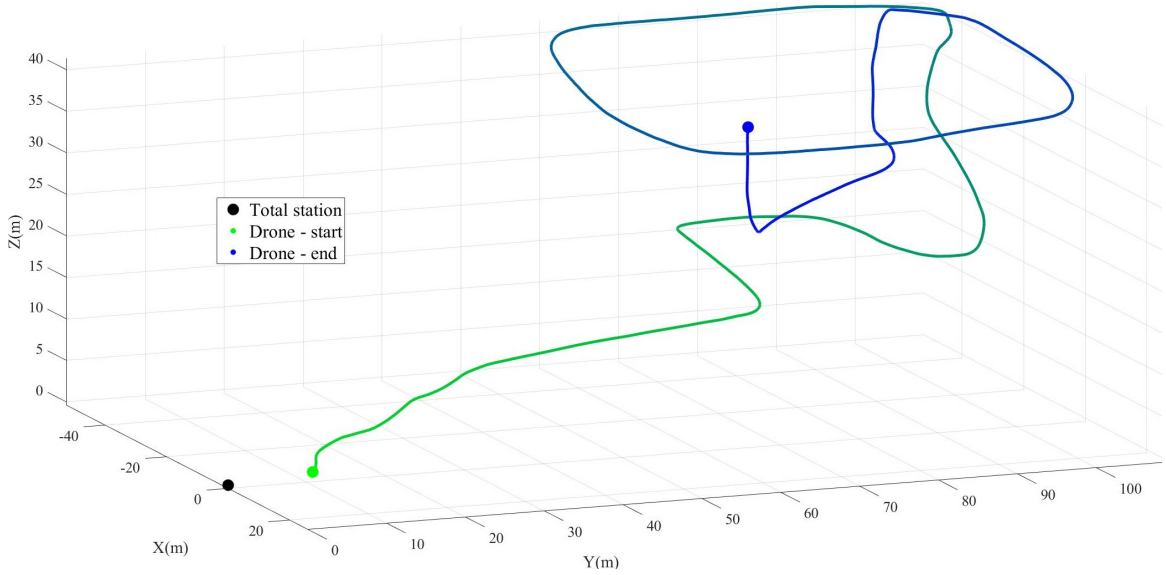
The average sampling interval is about 0.14 seconds but there's a longer interval of about 0.4 seconds for every 100 measurements according to Fig.5.3(b). At about 190 seconds from the beginning, there's a sudden acceleration of the drone, and the tracking from the total station is lost. The total station is not able to track moving objects with high speed and drastic acceleration. Once the tracking is lost, it's also difficult for the total station to find and lock the non-stationary target again. Since a ground truth trajectory of 3 minutes is enough for the evaluation of the visual 3D trajectory recovery task, we continue the experiment without total station tracking afterward.

It should be noted that the tracking status of 1 (warning) means the highest measurement accuracy may not be reached due to the fast motion of the target. Therefore, the tracking status would also be included in the ground truth pose file of the dataset.

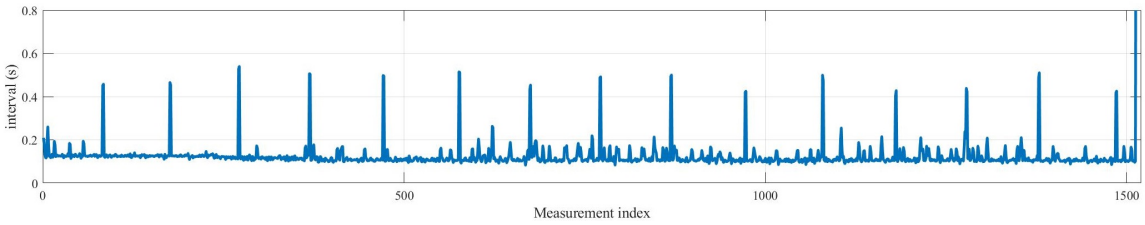
5.2 Onboard sensor measurements

The onboard sensor kits keep taking measurements during the flight and write the raw data into a log file. By parsing the log file after the experiment, we get the raw measurements of IMU as Fig.5.4. The GNSS's measurement after the transformation from WGS84 to local ENU frame together with its accuracy evaluation is shown in Fig.5.5.

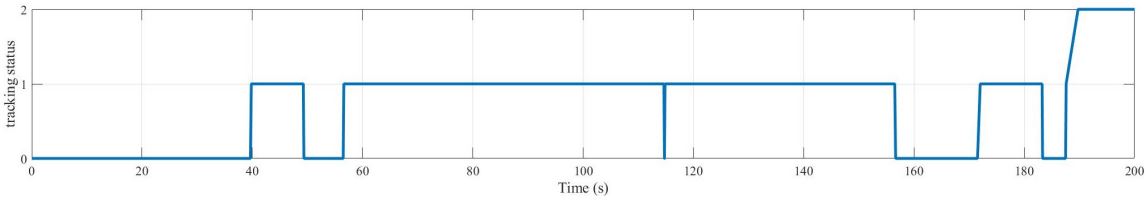
The sampling frequency of IMU and GNSS is about 100 Hz and 10 Hz, respectively. It is shown that the non-differential GNSS's positioning accuracy is about one meter, which is far beyond the error tolerance of the ground truth trajectory.



(a) Tracked trajectory (from green to blue) of the prism mounted on the Pixhawk drone in total station frame



(b) Total station tracking sampling interval (s)



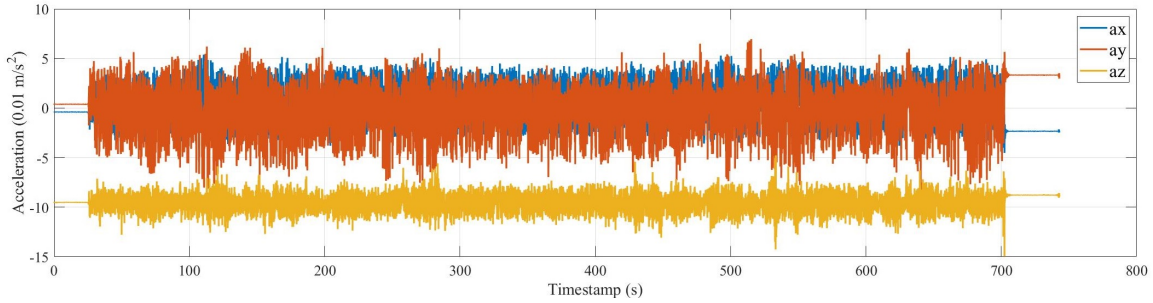
(c) Total station tracking status (0: fine, 1: warning, 2: tracking lost)

Figure 5.3: Total station tracking measurements for generating the dataset

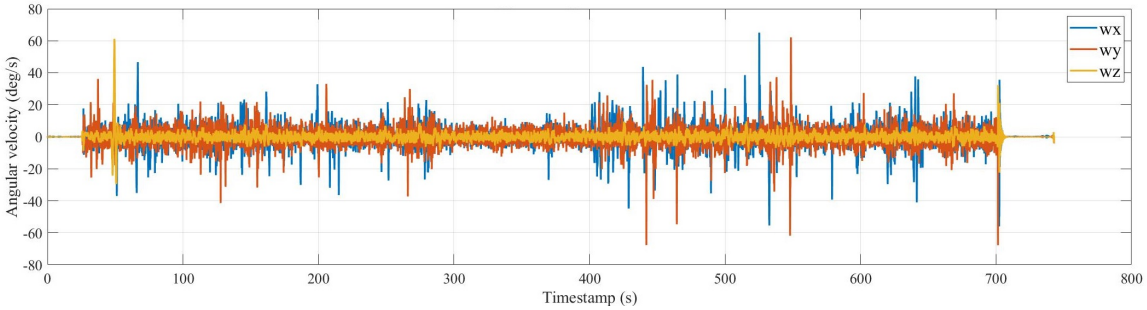
5.3 Estimated drone pose

By applying the calculation introduced in section 4.1, the pose at drone's body center in local ENU frame is estimated, as shown in Fig.5.6. The positioning accuracy is estimated using the covariance propagation equations introduced in section 4.1.4 and the instruments' accuracy specification described in chapter 3.

Fig.5.7(a) demonstrates the estimated error in different frames. It is shown the positioning accuracy of the prism in the total station frame is always better than one millimeter thanks to the highly accurate total station. The prism's positioning accuracy in the rotated body NED frame is estimated from both the prism's displacement from body center and the drone's orientation. Since the orientation is mainly estimated from the ever-drifting IMU, the resulting positioning error also drifts over time, from about one millimeter to about six millimeters in the end. Then with the geo-referenced transformation from the total station frame to the local frame, the resection error is involved in the total error budget. The drone center's positioning



(a) IMU accelerometer measurements



(b) IMU gyroscope measurements

Figure 5.4: On-board IMU measurements during the flight for generating the dataset

accuracy in the local frame, also known as the absolute positioning accuracy, ranges from 6 mm to 9 mm during the tracking, which is always better than one centimeter.

Fig.5.7(b) shows how the absolute positioning accuracy distributes on each direction. The error on Z axis is much larger than the horizontal error due to the larger geo-referencing error on vertical direction.

As shown in Fig.5.8, the estimated accuracy of drone center position can be rendered by error ellipsoids as well. The increasing trend over time and the main contribution on Z axis can be detected from the figure.

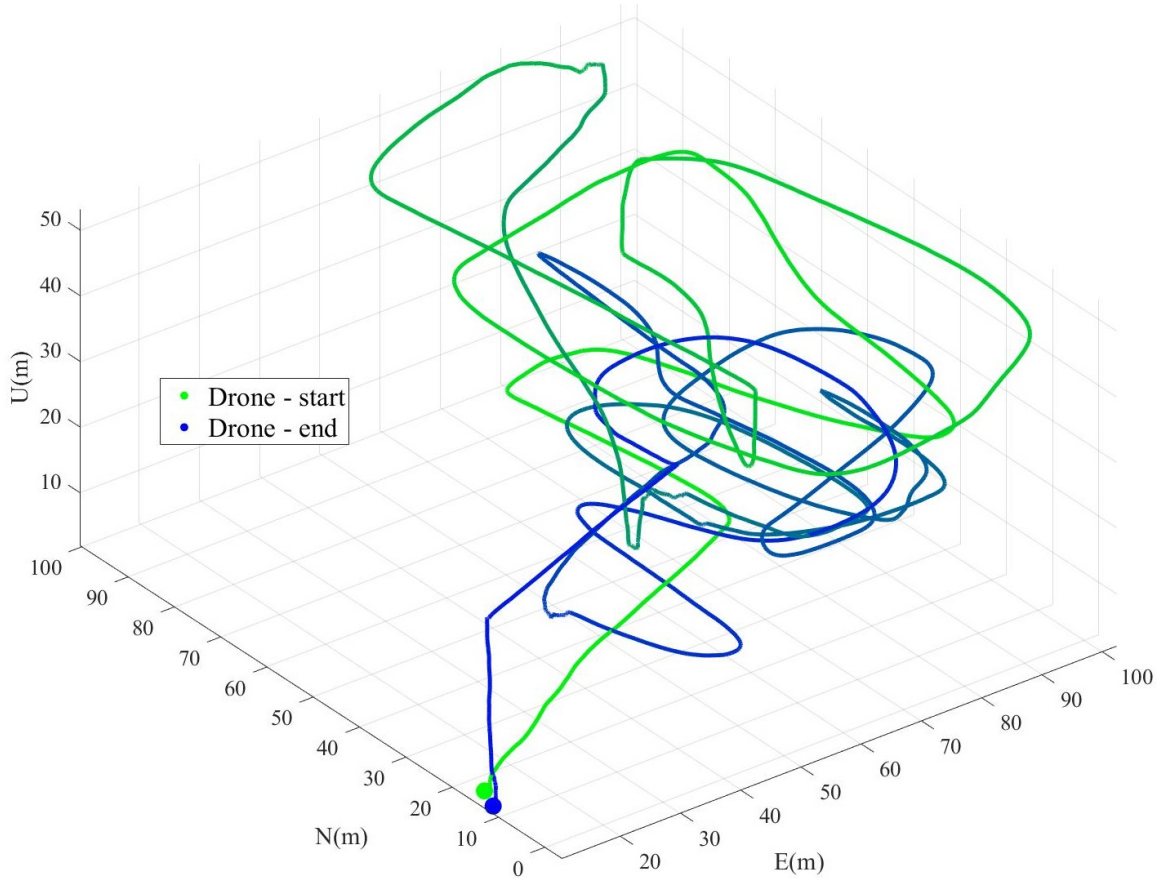
5.4 Videos of the drones

5.4.1 Overview of the cameras

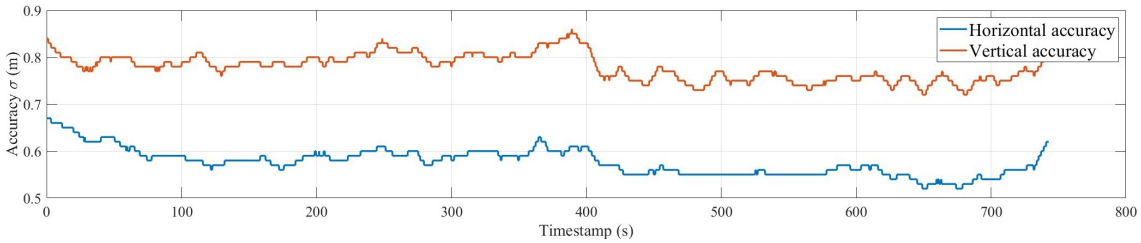
Before the flight, six cameras are mounted on tripods and set up around the site. The ID of these cameras are assigned anti-clockwise. Table 5.1 demonstrates the specification of each camera. Example frames from the recorded videos are shown in Fig.6.1.

Camera ID	Camera	Type	Approximate frame rate (fps)
0	Sony A5100	compact camera	30
1	Huawei P40 Pro	smartphone	30
2	GoPro 7	action camera	60
3	Sony G	compact camera	50
4	Samsung S10	smartphone	30
5	Sony NEX5N	compact camera	25

Table 5.1: Specification of the camera used for the dataset



(a) Drone trajectory (from green to blue) measured by on-board GNSS in local ENU frame



(b) On-board GNSS measurement accuracy estimation

Figure 5.5: On-board GNSS measurements during the flight for generating the dataset

5.4.2 Video synchronization

After the experiment, the videos from the six cameras are synchronized using the method described in section 4.2.2. The audio signals are extracted from the video and filtered by a bandwidth filter. As shown in Fig.5.10, the audio pattern (10 pulses with the interval of half a second) is revealed from the noise made by the drone motors. Therefore, we can pinpoint the timestamp of the first pulse from the triggering pattern.

During the experiment, the aforementioned audio pattern is triggered multiple times. The synchronization parameters, namely the time scale k and delay δt for each video are estimated using least square, as shown in Table 5.2.

Assume t_v is the timestamp in the video's time system, by applying $t_p = kt_v + \delta t$, the corresponding timestamp in the project referenced time system t_p can be calculated. The audio

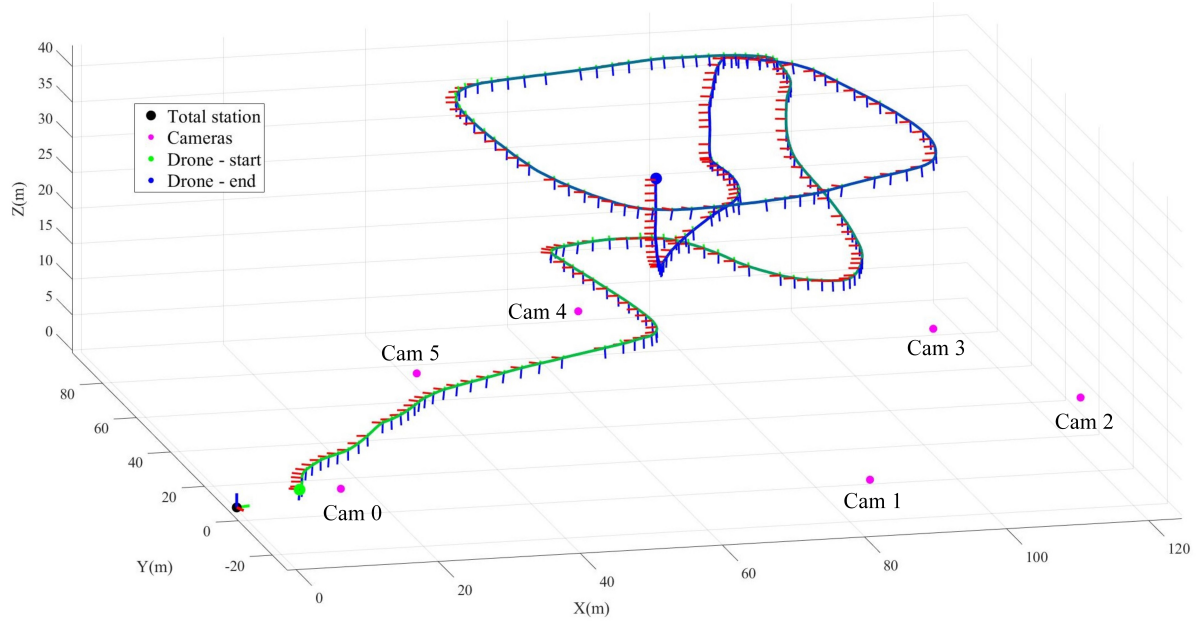


Figure 5.6: Estimated pose at the center of the drone in local ENU frame: the trajectory is rendered from green to blue and the orientation of each ten samples is rendered as the coordinate axis (x: red, y: green, z: blue).

signals are jointly aligned after applying the transformation, as shown in Fig.5.11. The frame-wise timestamps for each video are transformed to the project referenced time to realize the joint synchronization.

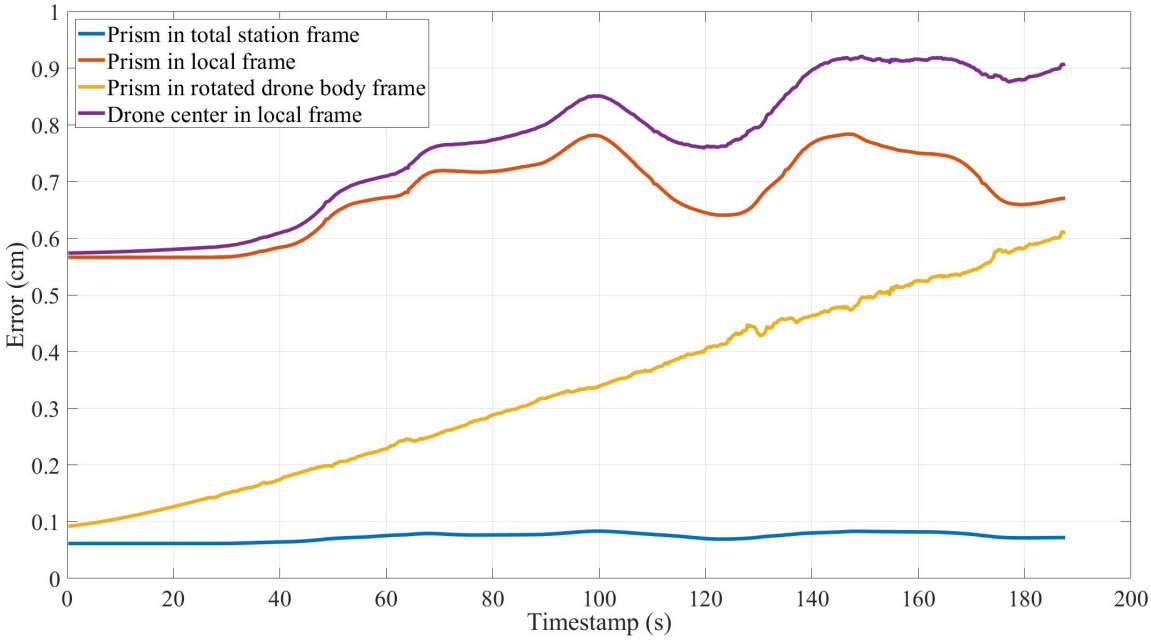
Camera ID	k	δt
0	1.000	10.48
1	1.000	7.73
2	0.998	9.80
3	1.000	-8.04
4	1.000	-7.25
5	1.000	14.41

Table 5.2: Parameters for camera synchronization.

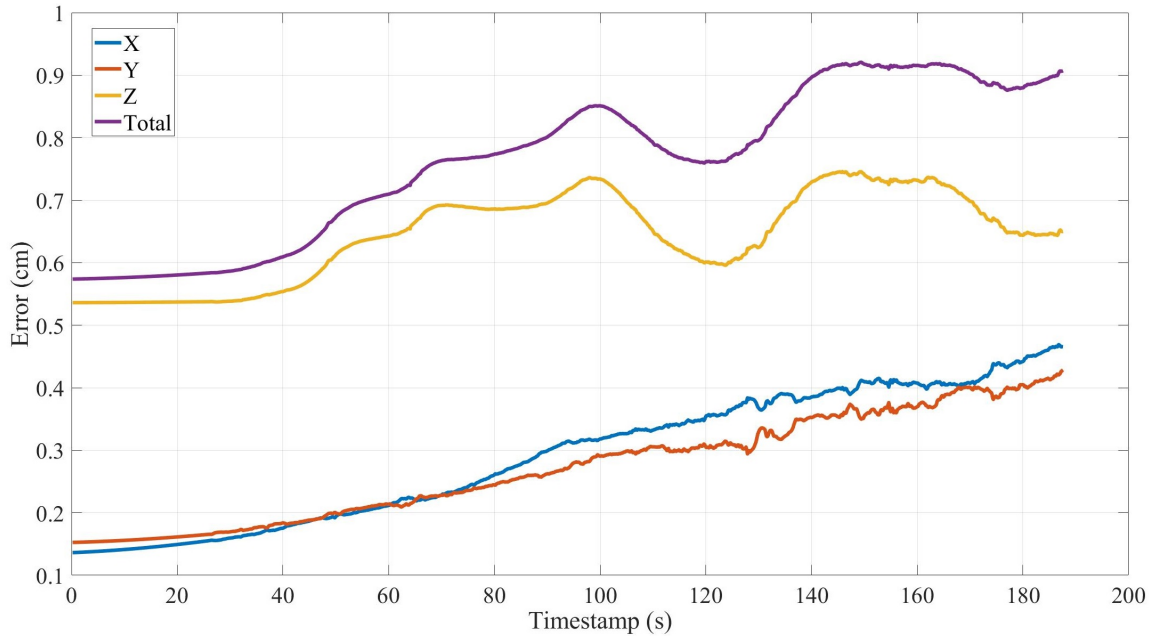
5.5 Post-flight measurements

5.5.1 Total station resection

Resection of the total station is implemented after the flight. The Trimble GNSS antenna is mounted together with a prism on a tripod pole. The vertical displacement from the GNSS antenna's bottom to the prism's center is 5.5 cm. After leveling the tripod pole, there's no horizontal displacement. The tripod pole is set up on five positions distributed evenly in the site. Such control points are both measured by GNSS and the total station. GNSS measurements in LV95 frame is coverted to WGS84 and local ENU frame sequentially. With the five sets of coordinates from both the local ENU and the total station frame, the 4 degree-of-freedom transformation (total station with leveling) between them can be estimated.



(a) Error quantity in different frames



(b) Positioning accuracy on each direction of the local ENU frame

Figure 5.7: Posterior positioning accuracy evaluation of the drone

5.5.2 Camera position measurement

Camera position measurement is conducted after the flight. It is accomplished by placing a prism exactly at the camera's position and taking the measurement of the prism by the total station. With the geo-referenced transformation of the total station achieved by resection, the cameras' coordinates in the local ENU frame are calculated as Table 5.3. The cameras' position relative to the drone's trajectory is shown in Fig.5.6.

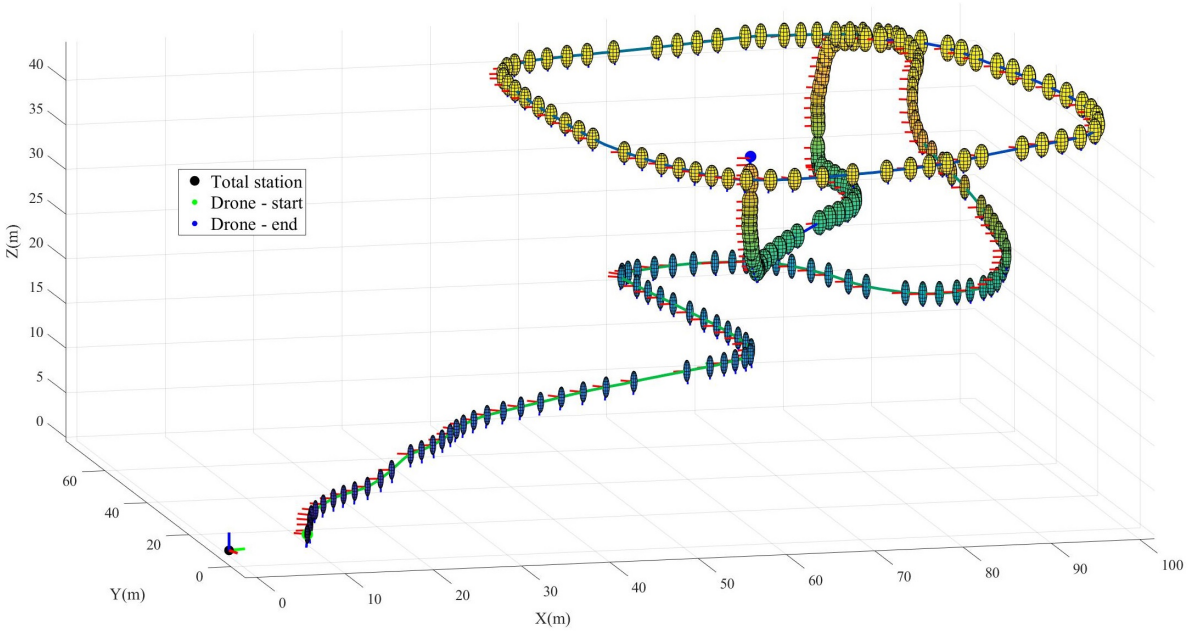


Figure 5.8: Positioning error ellipsoids in local ENU frame (the ellipsoid scale is magnified by 200 times, which means an error of 1mm corresponding to 0.2m on the figure)

(a) Camera 0

(b) Camera 1

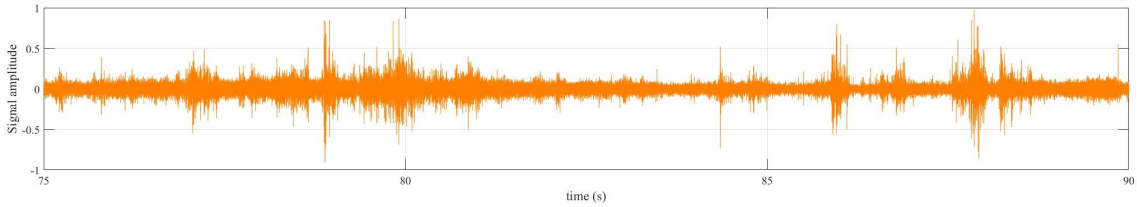
(c) Camera 2

(d) Camera 3

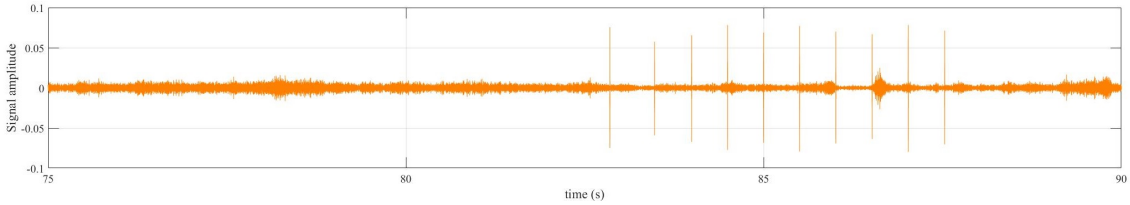
(e) Camera 4

(f) Camera 5

Figure 5.9: Example frames captured by six cameras for the dataset



(a) An example of the audio signal before filtering



(b) Triggering pattern clearly recognized after applying a band-pass filter

Figure 5.10: Audio signal before and after filtering

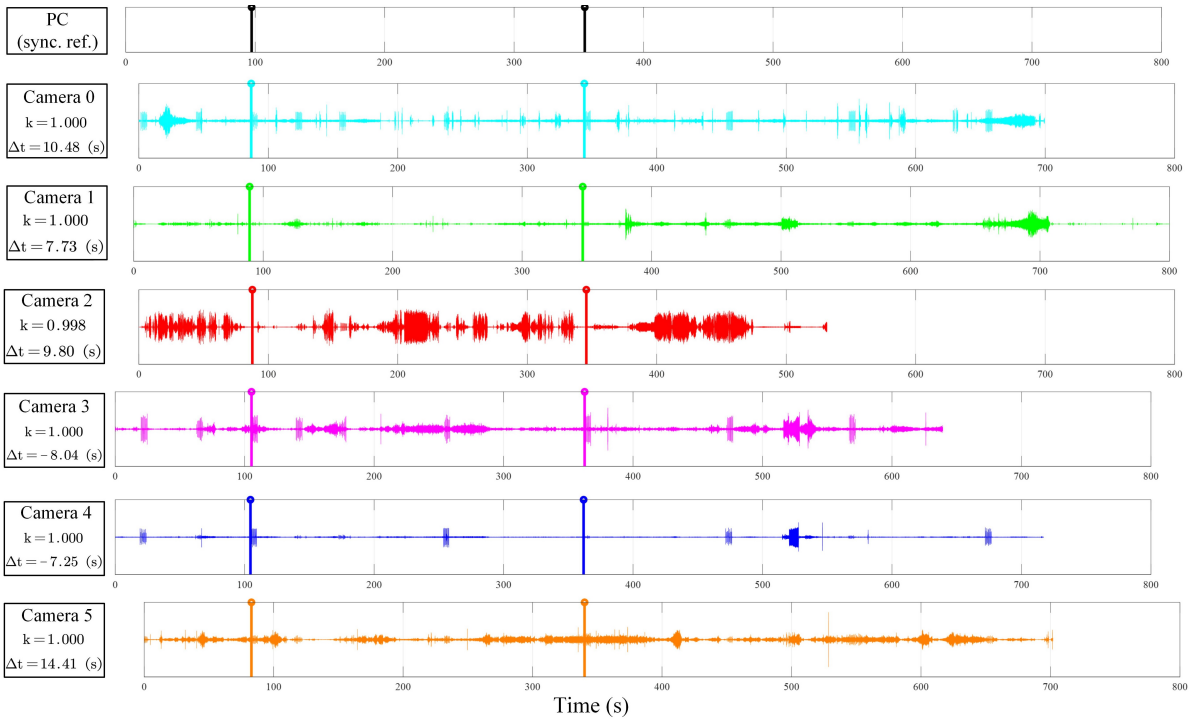


Figure 5.11: Joint synchronization among the cameras and the central computer by aligning the audio signals according to the triggering pattern. The aligned stems of different colors represent the beginning timestamps of the triggering pattern.

Camera ID	X	Y	Z
0	14.840	6.939	1.494
1	80.795	-28.735	7.588
2	124.730	30.981	2.161
3	114.509	74.924	1.801
4	69.976	97.659	1.343
5	40.785	70.713	0.804

Table 5.3: Position of the cameras in the local ENU coordinate system (unit: m)

5.5.3 Camera calibration

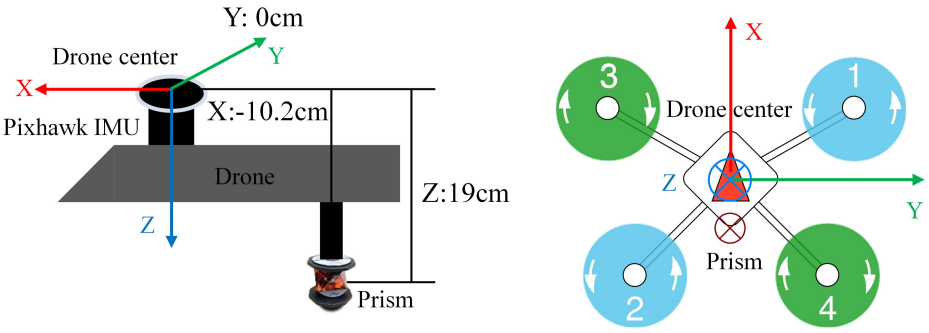
Camera calibration is also conducted for each camera on site after the flight with a calibration cheeseboard, as shown in Fig.5.12. Thus, intrinsic parameters for each camera are estimated.

(a) (b) (c)

Figure 5.12: Camera calibration post-flight using a chessboard: three example images.

5.5.4 Prism displacement calibration

Last but not least, the prism's displacement to the drone's (Pixhawk Cube's) center is measured by a steel measuring tape. The result is shown in Fig.5.13.



(a) Right side view (b) Top view

Figure 5.13: Calibration result of the prism's displacement from the drone's body center, as $\mathbf{p}_{\mathrm{b}} = \begin{bmatrix} -0.102 & 0.000 & 0.190 \end{bmatrix}$ (m).

Chapter 6

Dataset

In this chapter, we describe the structure and datum of the final dataset. We also provide an overview on how to use the dataset.

6.1 Dataset structure

The directory structure containing the dataset is shown as follows:

- Videos
  - Cam0
    - cam0.mp4
    - cam0_frame_ts.txt (frame-wise timestamp in project time system)
      Format: Frame_ID Timestamp(s)
  - Cam1
  - Cam...
  - Cam5
- Calibration (calibration video and results for each camera)
- Camera-locations
  - campos.txt (each camera's position in project local ENU frame)
    Format: Cam_ID X(m) Y(m) Z(m)
- Pose
  - fused_pose.txt/.csv (This file contains the Pixhawk drone's pose at its body center in project local coordinate system with its corresponding timestamp in project time system. The positioning accuracy evaluation is also provided. The pose can be regarded as the ground truth pose of the drone.)
    Format: Timestamp(s) X(m) Y(m) Z(m) Roll(deg) Pitch(deg) Yaw(deg) Std.X(m) Std.Y(m) Std.Z(m) TrackingStatus
- Detections
  - Drone0
    - cam0.txt (frame-wise ground truth 2D label of the drone's center in the image coordinate system. The coordinate would be [0,0] when the drone does not exists in the frame)
      Format: Frame_ID X(pix) Y(pix)

        * cam...
        * cam5.txt
    - Drone1
    - Drone2

- Raw-data
    - drone_log (drone's raw log file in Pixhawk's format, containing the raw measurements of onboard GNSS and IMU)
    - tps_tracking (total station's raw measurements)
    - leverarm_prism_in_body.txt (prism's relative position to drone center)
    - tran_mat_tps2local.txt (transformation matrix from total station frame to project local ENU frame)
    - local_origin_in_wgs84.txt (the coordinate of the project local coordinate system's origin in WGS84 geodetic frame)
    - project_time_origin_in_utc.txt (project time system's origin in UTC+0)
    - sync_coefficients_cam2pc.txt (synchronization coefficients for each camera with regards to project time system)
      Format: Cam_ID Time_scale Time_shift(s)
    - measured_coordinates.csv (raw measurement of cameras and resection control points' coordinates)

- cameras.txt (description of each camera according to its ID)
- drones.txt (description of each drone according to its ID)
- README.md (overview description of the dataset)

6.2 Dataset datum

Project referenced time system:

Origin: 19-01-2021 14:52:51 (UTC+0)

Project referenced local ENU frame:

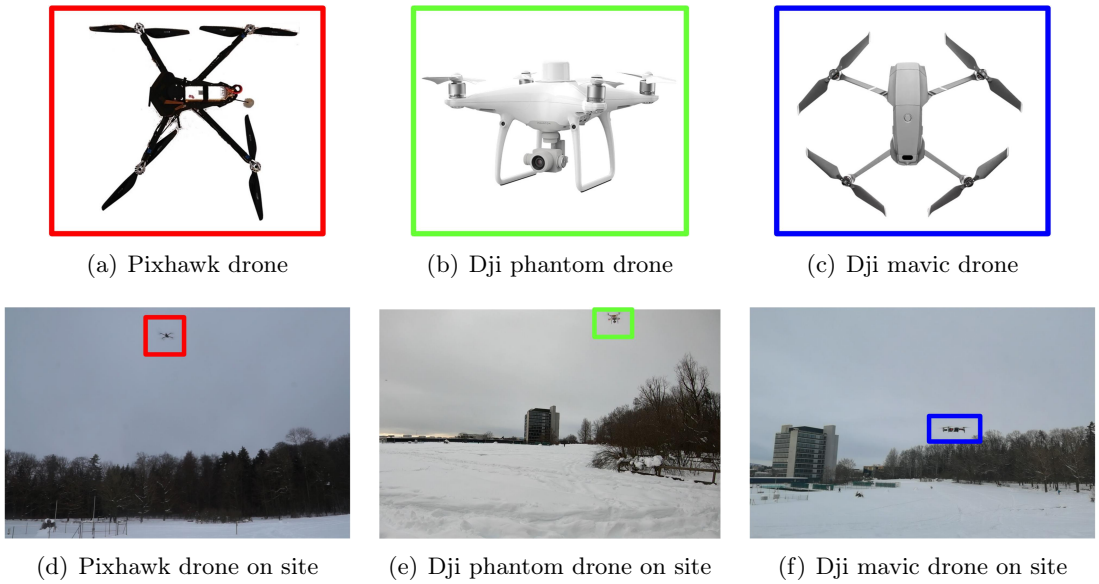
Latitude(deg)	Longitude(deg)	Elevation(m)
47.406149945	8.511420736	539.985

Table 6.1: Origin point of the project's referenced local coordinate system

6.3 Potential task

- Drone detection (object recognition) from single-view images, as shown in Fig.6.1 and Fig.6.2.
- Drone tracking from a single-view video. Since there are multiple drones in the dataset, multiple objects tracking task is also available.

- 3D Drone trajectory reconstruction from multi-view videos, along with tasks such as Camera pose estimation and video synchronization.
- 6D Drone pose (position + orientation) estimation from multi-view videos.



(a) Pixhawk drone (b) Dji phantom drone (c) Dji mavic drone

(d) Pixhawk drone on site (e) Dji phantom drone on site (f) Dji mavic drone on site

Figure 6.1: Drones used in the dataset

Figure 6.2: Multiple drones captured by one of the cameras: red: Pixhawk drone, green: Dji phantom drone, blue: Dji mavic drone.

Chapter 7

Conclusion and Outlook

In this project, we managed to construct a visual drone tracking and positioning dataset collected by a multi-sensor system, including a total station, on-board sensor kits, and an ad-hoc network of cameras.

By leveraging high accuracy total station measurements and sensor fusion techniques such as the extended Kalman filter, the absolute positioning accuracy for the drone's body center in the local frame can be better than one centimeter. By employing a radio-synchronized network of audio triggers, we can recognize the triggering pattern from the audio signal of each video, thus accomplishing the synchronization among the videos with sub-frame accuracy. The overall synchronization time delay of the system is about 10 milliseconds after aligning all the measurements' timestamp to the referenced PC time.

To the best of our knowledge, our dataset is the first one to contain subcentimeter accuracy ground truth drone trajectory, drone orientation, camera position, and the synchronized videos of multiple drones at the same time under a common space and time frame. Together with the previous dataset (Cenek 2020 [2]) acquired in summer, we present the society a comprehensive dataset[1] for research topics on tracking and localization of drones using computer vision and deep learning approaches.

Our future work will focus on using the dataset for algorithm development on certain tasks like high accuracy drone trajectory reconstruction, pose estimation, and multiple drones tracking in challenging conditions. We will also consider collecting similar data sequences with other sensors such as infrared cameras and event cameras under various weather conditions to enrich the dataset for more applications.

[1] https://github.com/CenekAlbl/drone-tracking-datasets

Bibliography

[1] G. Moeller and A. Geiger, *Navigation lecture notes*, 2020.

[2] J. Li, J. Murray, D. Ismaili, K. Schindler, and C. Albl, "Reconstruction of 3d flight trajectories from ad-hoc camera networks," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2020)(virtual)*, 2020, pp. 1621–1628.

[3] S. Lupashin, M. Hehn, M. W. Mueller, A. P. Schoellig, M. Sherback, and R. D'Andrea, "A platform for aerial robotics research and demonstration: The flying machine arena," *Mechatronics*, vol. 24, no. 1, pp. 41–54, 2014.

[4] A. Rozantsev, S. N. Sinha, D. Dey, and P. Fua, "Flight dynamics-based recovery of a uav trajectory using ground cameras," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6030–6039.

[5] A. Rozantsev, V. Lepetit, and P. Fua, "Detecting flying objects using a single moving camera," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 5, pp. 879–892, 2016.

[6] M. Ł. Pawełczyk and M. Wojtyra, "Real world object detection dataset for quadcopter unmanned aerial vehicle detection," *IEEE Access*, vol. 8, pp. 174 394–174 409, 2020.

[7] M. Bláha, H. Eisenbeiss, D. Grimm, and P. Limpach, "Direct georeferencing of uavs," in *Proceedings of the International Conference on Unmanned Aerial Vehicle in Geomatics (UAV-g)*, vol. 38. Copernicus, 2011, pp. 131–136.

[8] T. K. Kohoutek and H. Eisenbeiss, "Processing of uav based range imaging data to generate detailed elevation models of complex natural structures," *International archives of the photogrammetry, remote sensing and spatial information sciences*, vol. 39, no. B1, pp. 405–410, 2012.

[9] S. Guillaume and A. Geiger, "High precision remote determination of trajectories of non-cooperative aircraft," in *International Symposium on Certification of GNSS Systems & Services (CERGAL2017)*, 2017.

[10] H. Kirschner and W. Stempfhuber, "The kinematic potential of modern tracking total stations-a state of the art report on the leica tps1200+," in *Proceedings of the 1st International Conference on Machine Control & Guidance, June*, vol. 24, no. 26. Citeseer, 2008, pp. 51–60.

[11] A. Leica Geosystems, "Leica nova ms50 geocom reference manual," 2014.

[12] L. Meier, P. Tanskanen, L. Heng, G. H. Lee, F. Fraundorfer, and M. Pollefeys, "Pixhawk: A micro aerial vehicle design for autonomous flight using onboard computer vision," *Autonomous Robots*, vol. 33, no. 1, pp. 21–39, 2012.

[13] G. Cai, B. M. Chen, and T. H. Lee, *Unmanned rotorcraft systems*. Springer Science & Business Media, 2011.

[14] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "Imu preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation." Georgia Institute of Technology, 2015.

[15] J.-L. Blanco, "A tutorial on se (3) transformation parameterizations and on-manifold optimization," *University of Malaga, Tech. Rep*, vol. 3, p. 6, 2010.

[16] Z. Gojcic, S. Kalenjuk, and W. Lienhart, "Synchronization routine for real-time synchronization of robotic total stations," in *INGENEO 2017: Proceedings of the 7th International Conference on Engineering Surveying*, 2017, pp. 83–91.

Appendix A

Covariance propagation for SE(3) transformation

In this appendix, the covariance propagation rule for the SE(3) transformation of a 3D vector is demonstrated. Assume a spatial vector $\mathbf{p}$ is transformed to $\mathbf{p}'$ by a rotation matrix $\mathbf{R}$ and a translation vector $\mathbf{t}$, as:

$$\mathbf{p}' = \mathbf{R}\mathbf{p} + \mathbf{t} \tag{A.1}$$

where $\mathbf{t} = \begin{bmatrix} t_x & t_y & t_z \end{bmatrix}^\top$ and $\mathbf{R}$ can also be represented by Euler angle as the sequential rotation around x,y and z axis, as:

$$\mathbf{R}(\phi, \chi, \psi) = \mathbf{R}_z(\phi)\,\mathbf{R}_y(\chi)\,\mathbf{R}_x(\psi) \tag{A.2}$$

so that the SE3 transformation can be uniquely represented by a 6-dimensional pose vector $\xi = \begin{bmatrix} t_x & t_y & t_z & \phi & \chi & \psi \end{bmatrix}^\top$. Suppose ξ is known and the covariance matrix $cov(\mathbf{p}) \in \mathbb{R}^{3\times3}$ and $cov(\xi) \in \mathbb{R}^{6\times6}$ are also known. The unknown covariance matrix of the transformed vector $\mathbf{p}'$ can be calculated through:

$$cov(\mathbf{p}') = \mathbf{J}_{rp} cov(\xi)\,\mathbf{J}_{rp}^\top + \mathbf{R} cov(\mathbf{p})\,\mathbf{R}^\top \tag{A.3}$$

where the Jacobian matrix $\mathbf{J}_{rp} \in \mathbb{R}^{3\times6}$ has the form:

$$\mathbf{J}_{rp} = \left(\begin{array}{c|ccc} & j_{14} & j_{15} & j_{16} \\ \mathbf{I}_{3\times3} & j_{24} & j_{25} & j_{26} \\ & j_{34} & j_{35} & j_{36} \end{array}\right) \tag{A.4}$$

where each element can be calculated as:

$$\begin{aligned}
j_{14} &= -p_x \sin\phi\cos\chi + p_y(-\sin\phi\sin\chi\sin\psi - \cos\phi\cos\psi) + p_z(-\sin\phi\sin\chi\cos\psi + \cos\phi\sin\psi) \\
j_{15} &= -p_x \cos\phi\sin\chi + p_y(\cos\phi\cos\chi\sin\psi) + p_z(\cos\phi\cos\chi\cos\psi) \\
j_{16} &= p_y(\cos\phi\sin\chi\cos\psi + \sin\phi\sin\psi) + p_z(-\cos\phi\sin\chi\sin\psi + \sin\phi\cos\psi) \\
j_{24} &= p_x \cos\phi\cos\chi + p_y(\cos\phi\sin\chi\sin\psi - \sin\phi\cos\psi) + p_z(\cos\phi\sin\chi\cos\psi + \sin\phi\sin\psi) \\
j_{25} &= -p_x \sin\phi\sin\chi + p_y(\sin\phi\cos\chi\sin\psi) + p_z(\sin\phi\cos\chi\cos\psi) \\
j_{26} &= p_y(\sin\phi\sin\chi\cos\psi - \cos\phi\sin\psi) + p_z(-\sin\phi\sin\chi\sin\psi - \cos\phi\cos\psi) \\
j_{34} &= 0 \\
j_{35} &= -p_x \cos\chi - p_y \sin\chi\sin\psi - p_z \sin\chi\cos\psi \\
j_{36} &= p_y \cos\chi\cos\psi - p_z \cos\chi\sin\psi
\end{aligned} \tag{A.5}$$