CEAS-GNC-2022-060

CEAS EuroGNC 2022
"Conference on Guidance, Navigation and Control"
3-5 May 2022 @ Technische Universität Berlin, Germany

Implementation of a 3D laser-based mapping system using an sUAS swarm

Fabian Franz M.Sc., Chair of Flight Guidance & Air Transport, TU Berlin, 10587, Berlin, Germany. fabian.franz@schwerelos-berlin.de

Maarten Uijt de Haag Professor, Chair of Flight Guidance & Air Transport, TU Berlin, 10587, Berlin, Germany. maarten.uijtdehaag@tu-berlin.de

ABSTRACT

This paper presents the implementation of a laser-based mapping mission using a swarm of smaller unmanned aerial vehicles (UAVs) rather than a single larger and heavier UAV that is typically used in most mapping applications. The proposed system includes various functions to perform the mapping task collaborative including path planning and point cloud generation. These functions have been implemented in software and hardware using the Crazyflie 2.1 development platform and validated in a relevant test environment in the UAS Swarm Lab at the TU Berlin. Three different tests in various urban environments were executed successfully to analyze the developed system's performance. It is shown that a swarm of UAVs in combination with a user-friendly remote operator and planning tool, is a very promising approach to perform urban mapping and surveying tasks more efficiently.

Keywords: UAV; swarms, collaboration, navigation, planning, mapping

1 Introduction and Motivation

The use of Unmanned Aerial Vehicles (UAVs) for a variety of applications has become ever more promising over the last decade. The enormous cost and weight reduction of the components during the last years make this technology attractive. Applications for which expensive equipment (e.g., helicopters or aircraft) would have been needed some years ago can be made very cost-effective using UAVs. UAVs are used in almost every industry for missions like movie-making, surveying, cargo transport, healthcare, search and rescue, and many more [1].

Today's industrial UAV applications are mostly based on a single platform containing all the equipment necessary for the desired mission. Often this results in large UAVs, especially for challenging tasks where heavy hardware and long flight times are required. This increase in weight and size implies many disadvantages, like an increased risk of injury or fatality in case of an accident, stricter regulations and higher costs.

To alleviate these disadvantages, a swarm of multiple smaller UAV (multi-UAVs) platforms can be used rather than a single platform, with mission-relevant hardware (e.g., sensors) distributed across the group members [2]. Application of a swarm will require careful planning or coordination of the swarm members' behavior to achieve the mission and a reliable relative and absolute positioning capability for both autonomous or manual operation. Currently, UAV swarms are not common in industrial applications, but there is growing evidence that swarms will be an integral part of future UAV missions [3]. The following benefits of swarms and multi-UAVs compared to single platforms make specific commercial tasks very suitable [1]:

- **lower costs**: *due to smaller and cheaper parts of the UAVs*
- **better reliability**: *due to reduced complexity of the member UAVs*
- **better maintainability**: *due to reduced complexity and redundancy among the swarm members*
- **greater safety**: *due to the lighter UAVs*
- **higher speed**: *due to parallelization of the mission tasks*
- **longer flight times**: *due to weight reduction*
- **fewer regulations**: *due to the lighter and smaller UAVs*

One of the most promising fields for UAV swarms are applications where unknown or large areas need to be explored. The advantage of using a swarm instead of a single UAV is that the mission can be completed much faster and more efficiently. Using UAV swarms instead of a single UAV is not always the best solution, and its necessity has to be evaluated each mission type. But from the shown benefits above, four typical application fields (see Fig. 1) can be identified, where the usage of swarms is very promising [4].

Fig. 1 Overview of potential application field for UAV swarm [5], [6], [7], [8]

Previous work has addressed some aspects of the use of swarms for mapping. For example, Matthaei et al. [9] discussed a system to autonomously explore and map the Mars surface by using a robot swarm consisting of multiple UAVs, equipped with laser range scanners, and ground vehicles. The data from the UAVs was used to create a detailed map of the environment that was then used to perform path planning for the ground vehicles. As no global navigation satellite systems exist on Mars, simultaneous localization and mapping (SLAM) techniques were used for navigation purposes. Another example is an indoor swarm exploration mission, described by McGuire et al. [10]. The described approach uses a decentralized algorithm that allows a swarm of small UAVs to explore and map an indoor environment autonomously. A decentralized architecture is used here because a reliable data link between the main computer and the UAVs can not be guarenteed. For the implementation of the developed algorithms, the development platform Crazyflie 2.1 from Bitcraze was taken, which is also used in this paper.

This paper aims to develop a swarm mission for a task that is typically performed by one UAV nowadays. The desired mission's objective is to generate a 3D point-cloud of buildings within a defined area of a city by using more than one UAV as illustrated in Fig. 2. To perform this mission, the high-level concept is designed first. This includes the development of a path planning algorithm and the hardware and software concept. Afterward, these concepts are implemented in a test environment at the TU Berlin - UAV swarm lab. For the swarm members, the UAV development platform Crazyflie 2.1 from Bitcraze is used. Furthermore, a Graphical User Interface (GUI) is designed to guarantee the software's simple usability. Based on the gained insights and test results, an evaluation of the benefits of using UAV swarms for mapping missions is given.

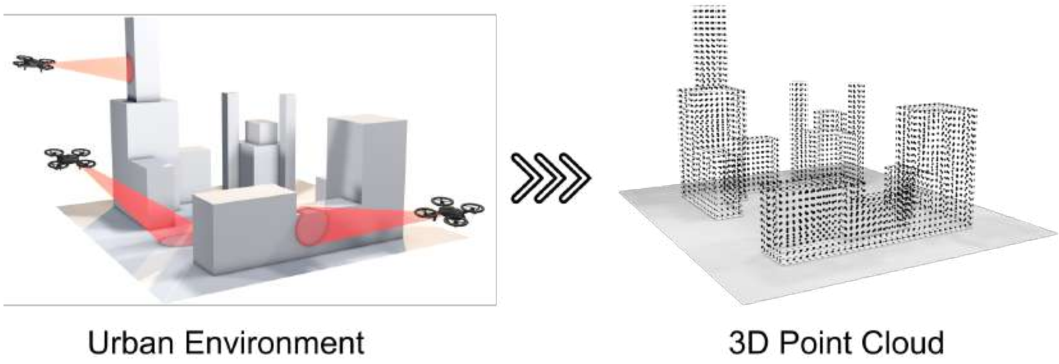

Fig. 2 Mapping mission of an urban environment

2 Swarm Mission - Mapping

As mentioned in the previous section, this paper focuses on UAV swarm mapping missions. The purpose of 3D mapping and surveying systems is to use sensors to observe objects or environments and use the sensor measurements to derive 3D models of their shapes and possible appearances [11]. For architectural mapping missions, mostly photogrammetry or laser range scanning are used [12]. The principles of both technologies are illustrated in Fig.3.

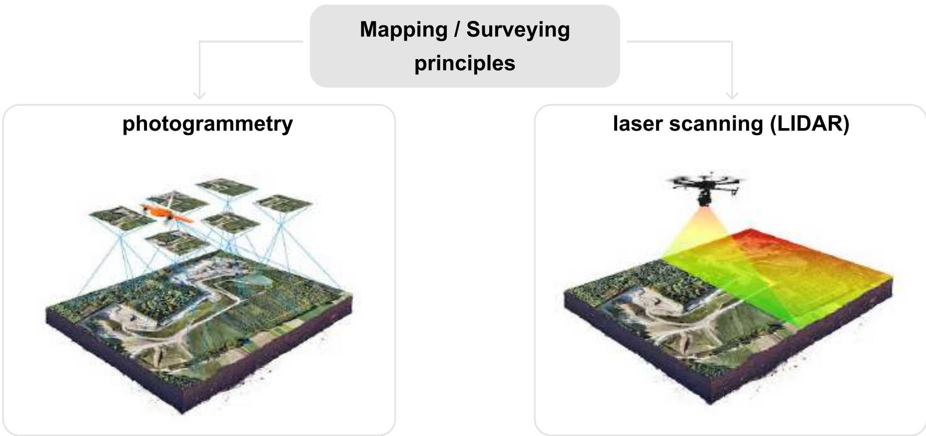

Fig. 3 3D mapping principles [13]

Digital photogrammetry uses digital images captured by photo or video cameras and uses triangulation of common points between multiple consecutive frames to derive a 3D point cloud of the observed environment. A disadvantage is that the photogrammetric survey is not possible in the absence of light since it is a passive method [12]. In contrast, laser range finding is an active method, as it is based on emitting laser light. The laser range finder determines the distance to an object or surface by timing the round-trip time of the pulse of light. Since the speed of light c is known, the round-trip time determines the travel distance of the light, which is twice the distance between the scanner and the surface. The scanner's laser beam can only point to one direction per pulse. This means only one point can be determined per period of time. To observe a 3D object, the laser beam must be moved after each pulse so that a 3D point cloud can be generated. This can be achieved by either moving the entire platform on which the laser ranger is mounted (e.g., the UAV) or changing the laser beam's exit angle (e.g., laser scanning). The second approach describes the so-called LIDAR (light detection and ranging) principle, which is common for UAV laser-based mapping tasks [14]. The advantages of laser methods are a high accuracy and resolution, light and weather independence, and fast point cloud generation. For both the laser and photogrammetry principles, it is necessary to map the desired object from multiple positions and heights to generate a full 3D point cloud. Therefore, the platform scanning sensor must be moved to different positions during the mission. This change of location can be done very easily by using UAVs for those missions, especially when the aim is to map environments that are hardly accessible.

To **georeference** the 3D point cloud, it is necessary that the UAV can determine its position, $\mathbf{r}^n$, and its attitude (pitch, roll, yaw angle) in a navigation coordinate frame. The UAV position can be determined by a variety of methods such as GNSS or based on local sensors such as inertial measurement units (IMU), laser range scanners, sonic rangers, or cameras. In the paper by Strümpfel et al., [15], it is shown that the GNSS availability in the urban environments is limited by shadowing and its performance deteriorated by none-line-of-sight reception and multipath. It is stated that it is essential to use multiple different localization methods to achieve the required navigation performance (i.e., accuracy, integrity, availability and continuity), especially for challenging missions where, e.g., the GNSS availability cannot always be achieved. If the mapping mission takes place in an unknown environment, where no absolute localization is available, SLAM (simultaneous localization and mapping) algorithms are used. These algorithms combine the scanning/mapping and localization system data so that a map can be constructed while the location is tracked simultaneously.

Finding the **optimal flight path** of an UAV for the desired mission is another challenging problem to be explored. Planning the path for mapping missions is especially complicated since these missions are often in an unknown environment with many obstacles that must be avoided. because of this operational environment, the path planning must be discrete as described by Massagué et al. [16]. This means that the flight path is planned based on observations of the environment during the flight (e.g., obstacles, etc.). The principle of path planning algorithms is first to determine the obstacles in the detection area and describe them by mathematical models such as geometric surfaces. Based on these models, possible UAV trajectories are calculated and smoothed afterward. The path planning algorithms can be divided into two groups: heuristic and non-heuristic methods. The heuristic algorithms are more time-efficient, but they don't always find the optimal solution. Non-heuristic methods use mathematical methods to find the optimal path, but they typically require more computational resources [17].

The last aspect of the multi-UAV mapping mission is the **communication network** that is used to relay coordination commands and exchange information between the swarm members: centralized or decentralized systems [18]. When using a centralized architecture, all the sensor data and flight parameters are sent to a ground station via a data link. The path planning and point cloud computations are performed on the ground, and control commands are sent back to the UAV. Advantages are that the UAV doesn't need to have a high computation power, and therefore cheaper and lighter components can be used. A reliable and high-quality datalink is also necessary during the whole mission. In contrast, using the decentralized system means that all the computations are made on the UAV. This requires a higher computation power for the on-board components and should be used if a continuous data-link cannot be guaranteed.

3 Methodology

Based on the mission's aim to generate a georefenced 3D point cloud of buildings within a defined urban area using an UAV swarm managed by a remote operator graphical user interface (GUI), the following high-level system requirements were defined:

1) The required UAV platforms shall be sufficiently agile to operate safely in an urban environment
2) Each swarm member shall be equipped with one or more laser range scanning sensors capable of supporting the pointcloud generation.
3) Each swarm member shall be able to operate autonomously given a specific flight plan.
4) The system shall have a collision-avoidance function to mitigate the collision risk while operating in close vicinity to other swarm members.
5) The user shall be able to define the mission using a Graphical User Interface.

Fig. 4 summarizes the mission aims and requirements and shows their relationship to the various resulting required system functions that will be addressed in the following sections.

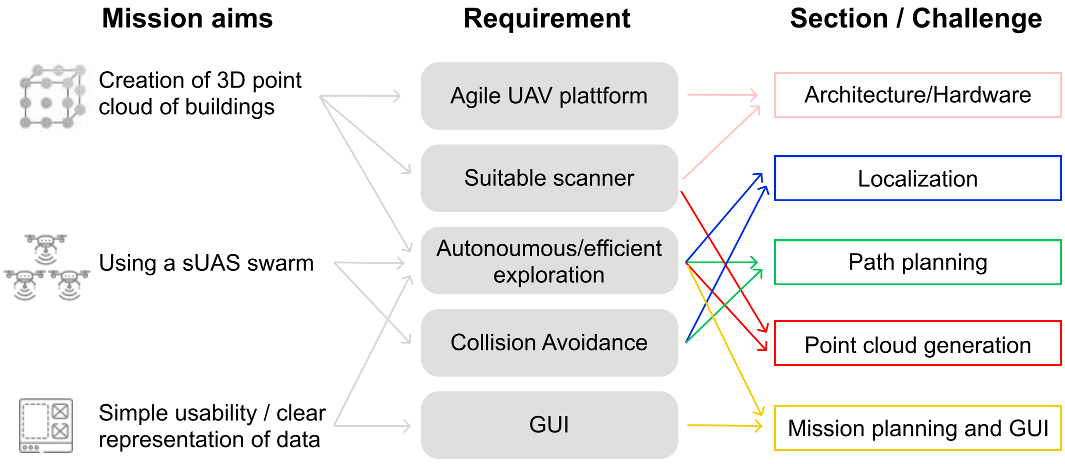

Fig. 4 Aims, requirements and challenges of the UAV swarm mapping mission

3.1 Hardware selection and swarm architecture

As high agility is required to navigate safely in the limited space of the urban environment, it is obvious that fixed-wing UAVs are not practical for the desired mission. Instead, a multi-rotor UAV is an adequate platform that fulfills the requirements. To evaluate the swarm mapping concept on a smaller scale, the Crazyflie 2.1 was selected (see Fig. 5) for the scaled-down version. Later the proposed architecture must be scaled to actual urban and UAV dimensions.

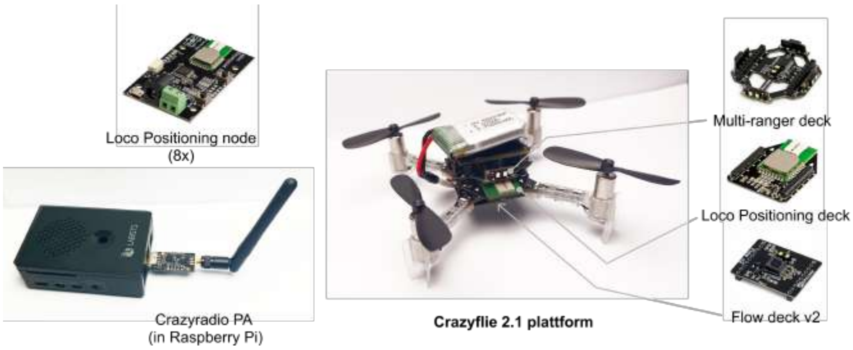

Fig. 5 Bitraze Crazyflie system - Components

To realize the desired swarm UAV mission, the Crazyflie 2.1 platform is equipped a PMW3901 optical flow sensor to estimate the velocity in x- and y-direction. In addition, the UAV is also equipped with a loco positioning tag, that can be used in collaboration with the Loco Positioning systems to obtain a 3D position in an in-door space in a manner similar to GNSS. The Loco Positioning tag and beacons (anchors) are based on the Decawave DWM1000 UlraWideband (UWB) transceiver and ranging sensor [1]. The communication between the Crazyflie 2.1 and central processor (a Raspberry Pi 4) is realized by a wireless radio link, the so-called Crazyradio (nRF24LU1 USB/Radio chip). Finally, the Crazyflie can make range measurements (distances) to the environment using the six VL53L1x time-of-flight laser ranging sensors with a maximum range of 4m (left right, forward, backward, up and down). The range measurements, d_i, can be converted to coordinates in the body-frame $\mathbf{p}_i^b$ where i refers to the specific laser pointing direction; left, l, right, r, up, u, down, d, forward f, or backward, b, given by:

$$\mathbf{p}_f^b = \begin{bmatrix} d_f & 0 & 0 \end{bmatrix}^T, \mathbf{p}_b^b = \begin{bmatrix} -d_b & 0 & 0 \end{bmatrix}^T, \mathbf{p}_l^b = \begin{bmatrix} 0 & -d_l & 0 \end{bmatrix}^T, \mathbf{p}_r^b = \begin{bmatrix} 0 & d_r & 0 \end{bmatrix}^T \tag{1}$$

[1]https://www.decawave.com/product/dwm1000-module/

$$\mathbf{p}_u^b = \begin{bmatrix} 0 & 0 & d_u \end{bmatrix}^T, \mathbf{p}_b^b = \begin{bmatrix} 0 & 0 & -d_d \end{bmatrix}^T \tag{2}$$

3.2 Localization and attitude method

The quality of the 3D point cloud is not only determined by the ranging performance of the ranging sensors described in the previous section, but also by the accuracy of the position estimator. To obtain the most accurate estimate, $\hat{\mathbf{r}}^n$, of the UAV position, $\mathbf{r}^n$, given the on-board sensors, a Kalman filter was used to integrate the data from the IMU, the flow sensor, and the downward pointing laser rangefinder (note: ˆ symbol indicates an estimate rather than the true parameter). The paper from Ashraf et al. [19] describes such a multi-sensor integration approach. A similar approach is also implemented in the Crazyflie 2.1 development system (Fig. 6).

Fig. 6 Position and attitude estimation using a Kalman filter

3.3 Point cloud generation

The 3D point cloud is derived from the range measurement vectors given in Eqns.1 and 2 using the approach shown in Fig. 7. First, a filter algorithm is used to smooth the range measurements and remove implausible values (e.g., steps and outliers). Next, the range measurement vectors are transformed to the global coordinate system using the output of the position estimator, $\hat{\mathbf{r}}^n$, and the attitude calculation. The Euler angles obtained from the attitude computation are used to obtain the coordinate transformation matrix from the body- to the navigation-frame, $\mathbf{C}_b^n(\phi, \theta, \psi)$ whose expression can be found in any text book.

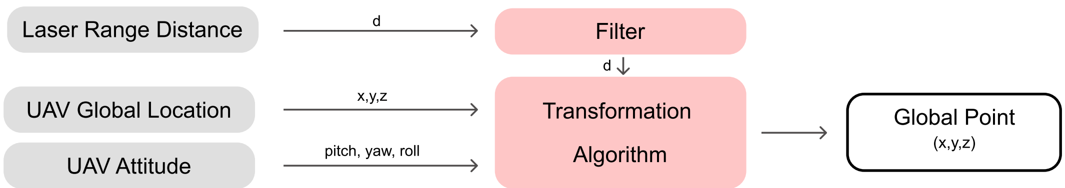

Fig. 7 Transformation Algorithm - I/O

Given $\hat{\mathbf{p}}_i^b$, $\hat{\mathbf{r}}^n$, and $\hat{\mathbf{C}}_b^n$, the point cloud points can be obtained as follows:

$$\hat{\mathbf{p}}_i^n = \hat{\mathbf{r}}^n + \hat{\mathbf{C}}_b^n \hat{\mathbf{p}}_i^b \tag{3}$$

These computed points are added to the point cloud matrix *PC* and displayed in the graphical user interface afterward.

3.4 Path planning

The path planning algorithm for the desired mission must meet the following two requirements: (A) to avoid obstacles and at the same time guaranteeing a scan from all perspectives; (B) to optimally distribute multiple UAVs in a swarm and keeping a safe distance to each other. The developed algorithm is based on the methods proposed by Smith et al. [20] and Almeida et al. [21]. The algorithm, described in [20] consists of four phases: In the first phase, a geofence is defined that includes that part of the urban environment that will be mapped. The area outside of this geofence is referred to as the obstacle-free zone. In the second phase, the UAV moves according to a fixed pattern in the obstacle-free area and scans the urban environment from a safe distance. These scanning results are used in the third phase to calculate an optimal and obstacle-free path inside the urban environment. In the fourth phase, the UAV moves along this calculated path and scans the environment from a closer distance, which makes it possible to create a high-resolution point cloud. The path algorithm designed for our proposed swarm mapping mission follows the principles of these phases, but as the requirements and conditions differ from the ones used in [20], a simplified algorithm has been developed. This algorithm first calculates an optimal observation path independent from the number of UAVs in the swarm. Afterward the UAVs are distributed to this calculated paths by using an the "UAV distribution algorithm". The modules and interfaces of the developed path planing algorithm are illustrated in Fig. 8 and explained in the sections below.

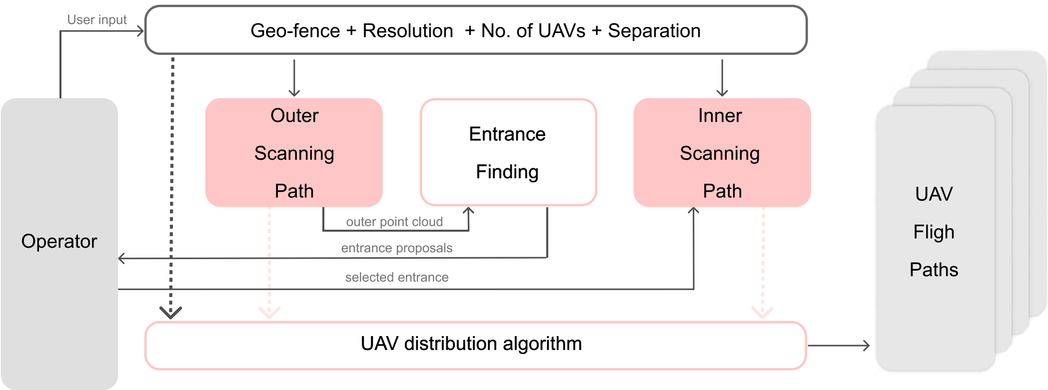

Fig. 8 Path Planning Algorithm - Overview

3.4.1 Geo-fence definition

In the first step, the operator defines a geofence surrounding all the buildings that must be scanned by using a Graphical User Interface (GUI). The definition of this geofence could either be done using ground-based surveys or by using existing (satellite) maps.

The 2D shape (x-y-plane) of this geofence could be any polygon or circle and should be selected based on the environment that will be observed. For this project a simple rectangular shape will be used, so that the results are four 2D coordinates P (x, y) and a maximum height z_{max}.

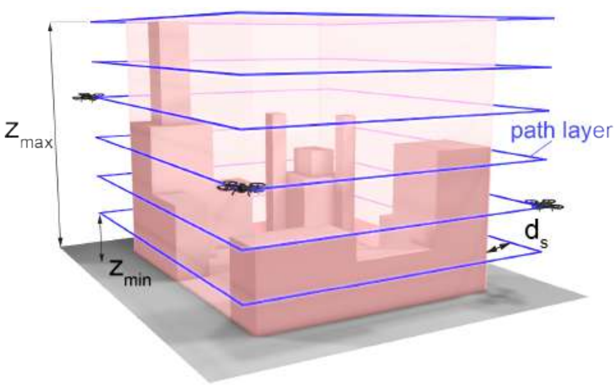

Fig. 9 Outer Scanning Path - Principle

3.4.2 Outer scanning path

Based on the defined geofence and the point cloud resolution, the "Outer Scanning Path" is generated. The proposed algorithm defines multiple height layers between the minimum flight height z_{min} and the maximum geofence height z_{max}. The number of height layers and the distance between them depends on the desired point cloud resolution. A path is generated on each height layer, which follows all four side surfaces of the geofence by a safety distance d_s. To achieve a fully scanned environment, the mapping UAV starts at the first height layer with an orientation so that the measuring direction is orthogonal to the flying direction. It follows this defined path and changes afterward to the next height level, where the next scanning loop is executed. This process is repeated until the last height layer (z_{max}) is completed. An example of a planned path is illustrated in Fig. 9. The output of the "outer scanning procedure" is the outer scan point cloud PC_{os}.

$$PC_{os} = \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \\ \vdots & \vdots & \vdots \end{bmatrix} \tag{4}$$

3.4.3 *Entrance finding*

To generate a full 3D point cloud, it is necessary to perform a scan from the inner side of the city as well. Therefore, a collision-free path must be found, allowing the UAVs to fly to the city center. To solve this problem, an algorithm has been developed to identify gaps between the buildings based on the scan results from the *Outer Scan*. The identified gaps are sent to the GUI, where the operator can decide which entrance will be used. Based on this decision, the path to the center of the city is generated. There are various approaches that which can be used to identify holes and other geometries in point clouds such as DBSCAN [22] or Alpha Shapes [23]. However, since the methods are universally valid for all types of point clouds, the are very complex. A tailored algorithm has been developed based on the problem definition and the output of the "Outer Scanning path" algorithm. The entrance finding algorithm consists of three steps and is introduced below.

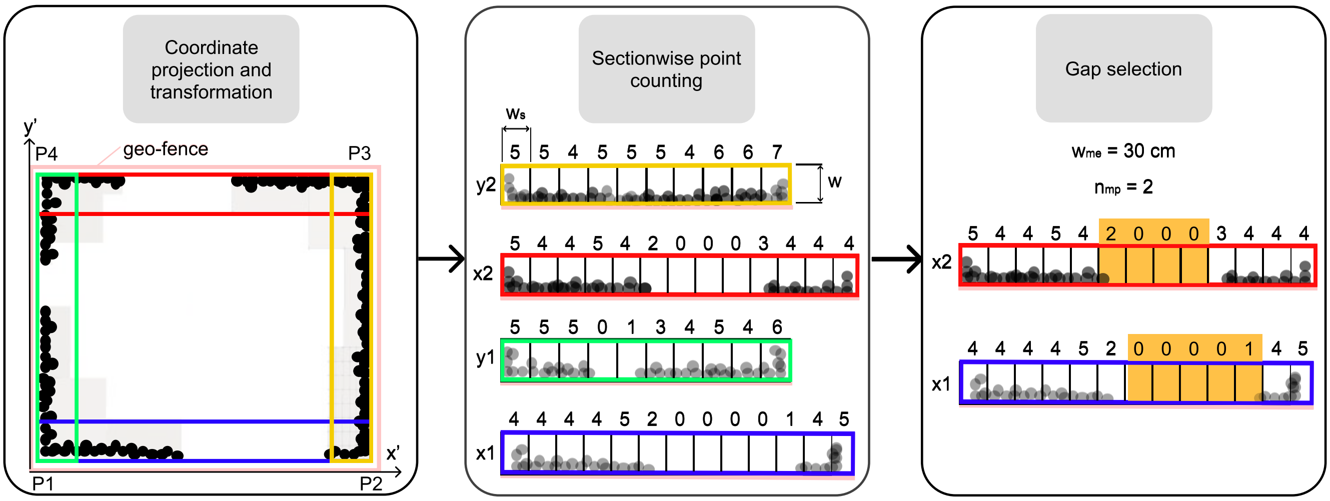

Fig. 10 Entrance finding algorithm

In the first step, all points of PC_{os} are projected on the x-y-plane, resulting in a 2D-matrix PC_{osXY}. Afterward, a coordinate transformation is performed to align the point cloud axes as much as possible with the geofence sides. This transformation includes a shift of origin followed by a rotation. Mathematically, this transformation can be expressed by multiplying the point cloud matrix PC_{osXY} by the transformation matrix T:

$$PC'_{osXY} = \begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \\ \vdots & \vdots & \vdots \end{bmatrix} \cdot \begin{bmatrix} cos(\Theta) & -sin(\Theta) & 0 \\ sin(\Theta) & cos(\Theta) & 0 \\ -x_{P1} & -y_{P1} & 1 \end{bmatrix} \tag{5}$$

where angle Θ is the angle between the x-axis and the side of the geo-fence between $P1$ and $P2$. It can be calculated as follows:

$$\Theta = acos\left(\frac{x_{P2} - x_{P1}}{\sqrt{(x_{P2} - x_{P1})^2 + (y_{P2} - y_{P1})^2}}\right) \tag{6}$$

In the next step, strips along all sides of the geofence and parallel to x' or rather y' are defined. These strips are as long as the respective side and have a defined width w in the direction of the geofence center, as illustrated in Fig. 10. They are divided into multiple sections of the width w_s. Afterward, the amount of points in each section and strip is determined. In the third step, an optimization algorithm finds all connected sections that do not exceed the maximum point amount per entrance n_{mp}. If the connected sections are larger than a minimum entrance width w_{me} they can be proposed as a possible entrance. The algorithm runs through all sections and finds all connected sections that match the requirements. All identified possible entrances are sent to the user and displayed in the GUI. Afterward, the user must select one of the proposed entrances to continue the mission.

3.4.4 *Inner scanning path*

After the UAVs are entering the city, a path must be generated to scan the buildings from the inner side. The first approach was to implement a wall following algorithm, however, as there can be gaps between the buildings, a simple wall following approach does not work reliably. Also, there is the problem that the 1D laser ranger can easily miss protruding edges, and thus a collision-free path could not be guaranteed. Instead, a three-stage planning algorithm has been developed and introduced below.

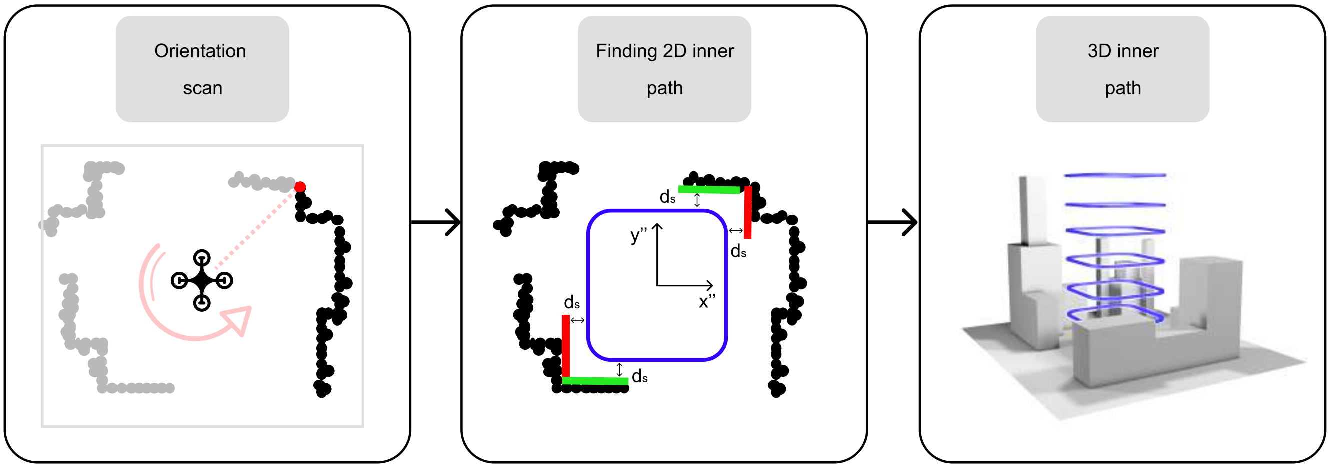

Fig. 11 Inner scanning path algorithm

In the first step of this algorithm, the UAV moves to the center of the city (center of the geofence). At a relatively low altitude, the UAV turns around its yaw axis and scans the surrounding environment by using its front range finder. Unfortunately, an earlier test showed that the accuracy of this scan is not very high, as the used hardware (Crazyflie 2.1) is not capable of holding its position steady while rotating around the yaw axis. However, the results of this scan are good enough to obtain an initial estimate of the environment. Thus, these results are used to determine a collision-free path, which allows performing a more accurate scan. To find this path, a coordinate transformation similar to Eq. 5 is performed. Thus, the coordinate system aligns with the orientation of the major buildings, and the origin is in the city center. The aim of the following algorithm is to find the nearest orthogonal building fronts for each direction ($+/-x''$ and $+/-y''$). Therefore, the algorithm loops through the transformed point cloud matrix. The angle between the current and the previous point is calculated for each iteration as follows:

$$\Theta = acos\left(\frac{x_i - x_{i-1}}{\sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}}\right) \tag{7}$$

The vector between the points can be interpreted as parallel if one of the following conditions is true; parallel to x-axis: $-10^\circ <= \Theta <= 10^\circ$ and parallel to y-axis: $80^\circ <= \Theta <= 90^\circ$ or $-90^\circ <= \Theta <= -80^\circ$.

Furthermore, a group of points can be interpreted as a building front if at least for five consecutive iterations the parallelism condition is true. After all non-parallel point groups are identified, the two closest orthogonal building fronts for each axis in positive and negative directions are determined. Based on these results, the *Inner Path* is generated by keeping the safety distance d_s to all four determined building fronts, as illustrated in Fig. 11. As the measuring direction of the laser rangefinder is orthogonal to the flying direction, the corners must be rounded to achieve a full scan. The 3D path is determined afterward by projecting this 2D path on the same height layers as introduced for the Outer Scan algorithm.

The developed and described algorithm is tailored to urban scenarios with specific geometries where the geofence is rectangular and all buildings are parallel or at right angles to each other. To extend this algorithm to more general scenarios, the "Finding 2D inner path" method must be modified. A possible approach could be to combine a wall following algorithm with the results from the orientation scan. The development of a more advanced algorithm should be investigated in a future project.

3.4.5 UAV distribution algorithm

The path planning algorithms designed thus far do not consider that multiple UAVs are used for the mission. Therefore, a UAV distribution algorithm is developed to allocate the UAVs to suitable path sections. This algorithm should ensure collision avoidance between the UAVs and realize a time-efficient scanning procedure. As the results of the outer scanning phase are necessary to continue the inner scan, it is not possible to distribute the UAVs between these phases. Instead, the algorithm only considers a distribution where all UAVs are in the same mission phase.

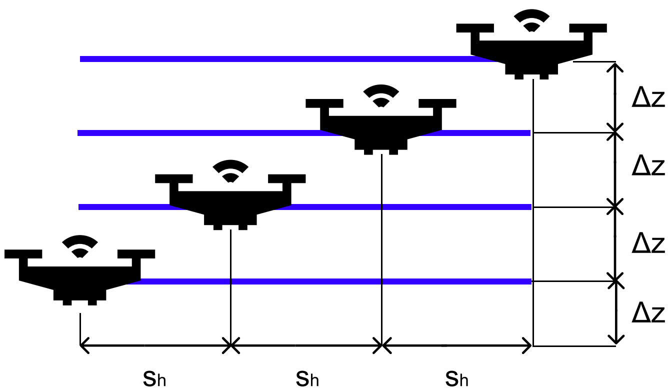

Fig. 12 Height layer distribution

Two possible solutions to this problem have been identified in the context of this paper: local distribution and height layer distribution. Whereas, the local distribution concept distributes the UAVs to local sections (e.g., side surfaces of geo-fence), where it scans the whole area, the height layer distribution algorithm distributes the UAVs to defined height layers. For this mission, a height layer distribution algorithm has been implemented. Each UAV is assigned to another height layer and follows the determined path. After the UAV completes its path on the actual layer, it changes to a next free path until all heights are scanned. Additionally, to the height layer separation Δz, a horizontal separation logic is implemented to ensure that the UAVs are not flying in each other's down-wash as that could lead to a crash.

For the outer scanning phase, it is possible that all UAVs are moving at the same time. Thus, the scanning process can be performed very fast. But as the space in the center of the city of test environment is limited and the navigation performance is not precise enough, a safe horizontal separation of multiple UAVs cannot be guaranteed. This could result in UAVs flying on top of each other, which would cause a crash due to the generated wind. Thus, for the given test scenario only one UAV can be at the inner of the city at the same time. The inner scanning phase needs to be executed consecutively by the UAVs. So, the first UAV enters the city and does the orientation scan. Afterward, it's doing the assigned scanning heights while the other UAVs are on standby outside. After it has completed its paths, it leaves the city, and the next UAV enters the urban environment. This process is repeated until all UAVs have scanned their assigned layers. For actual urban scenarios it will also be possible, that multiple UAVs enter the city at the same time, due to the larger scale of of building with respect to the UAV dimensions.

4 Test setup and results

Multiple tests have been performed in the TU Berlin UAS Swarm Lab to evaluate the developed and implemented system. To facilitate this, a city model was built consisting of different-sized cardboard boxes representing various buildings. To meet the project's requirements, there must be at least one entrance (gap) between these buildings to make the inner scan possible. Also, all buildings were placed at right angles to each other. In addition, eight anchors of the Crazyflie Loco Positioning system were installed to provide part of the positioning capability. These anchors are positioned in each corner of the room where the city model was built up. The urban configurations used for tests are shown in Fig. 13 below.

Fig. 13 Urban configurations used for tests

4.1 Adaptability test

The first test aims to validate the correct operation of the developed path planning algorithms using three city configurations (A, B and C). For each test round, two UAVs are used so that also the distribution algorithm can be verified. For all three urban configurations, the scanning mission was completed successfully. The outer paths were generated based on the geofence defined by the user for each test individually. The entrance finding algorithm has identified reasonable possibilities to enter the city. The inner path algorithm also calculated a suitable path to scan the inner environment of the scenarios, and the distribution algorithm distributed the two UAVs as expected. Thus, it is proven that the developed and implemented algorithms are working and allow the defined missions to be executed.

4.2 Precision test

In the second test the accuracy of the developed scanning system is evaluated. Part 1 of this test aims to determine the accuracy of the system, which is dependent on the performance of the localization system and the laser ranger. A simplified urban environment, as shown in Fig. 13 (D), is setup and surveyed using the proposed system. For simplification reasons, the Outer Scan is only performed on two height layers. The points obtained by the Crazyflies and processed using the proposed algorithms are projected to the x-y plane and compared to the ground truth, i.e. the real map of the outer environment (blue lines). The results are depicted in Fig. 14. In the plot, all points with a deviation greater than 100 *mm* to a perfect measurement are colored in red. Most of the 186 determined points have a deviation less than 100 *mm* (115 points). The average deviation of all points is $\overline{\Delta d_i} = 84$ *mm*. As illustrated in the x-y-plot in Fig.14, the greatest deviations of the points are reached at the left and upper parts. Reasons for this measuring inaccuracy could be incorrect UAV localization estimates or noise of the laser measurements.

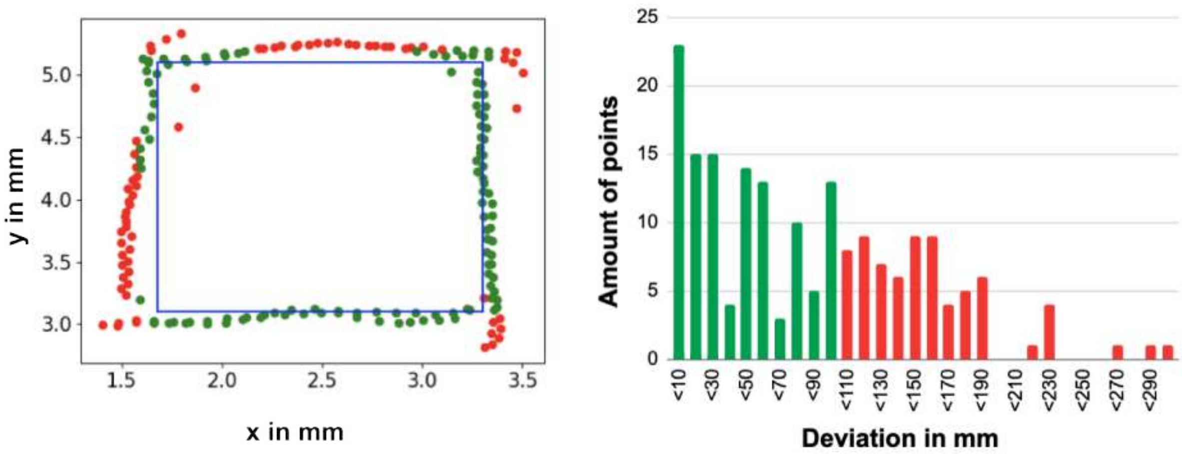

Fig. 14 Precision test results - Point deviation from ideal scan

As the first sub-test represents an idealized situation, the second sub-test aims to perform a more realistic evaluation of the system's accuracy. Therefore, a full scan of the urban environment shown in Fig. 13 (A) is performed. For the mission, two Crazyflies are used, and four height layers are scanned (inner and outer). The resulting 3D point cloud is compared with an ideal 3D model based on the laser survey. Areas, where the ideal and the UAV-based measurements have a high deviation from each other, are identified to evaluate the mapping systems precision for real applications.

As earlier tests showed similar results, the systems measuring accuracy is specified with $\Delta d_a = \pm 100$ *mm* as a guide value. From this evaluation, and as expected, it is clear that the system cannot generate high-resolution scans. However, the accuracy is sufficient to create a point cloud that gives an estimation of the urban environment. This evaluation is confirmed by the scanning results from the second sub-test, which included the mapping of the urban environment illustrated in Fig. 14 (A). In the generated point cloud, typical characteristics like the height and constellation of buildings can be identified. But the test also showed some weaknesses of the system. The first problem identified is that the localization performance is becoming worse for higher UAV flying altitudes. This results in more inaccurate measurements and can be explained by a lesser influence of the flow sensor. The second problem is that the laser range sensor noise increases as a function of range with complete random readings for ranges further than $4m$. This also results in mistaken measurements of the 3D point cloud and could be eliminated by implementing a more advanced filter algorithm.

4.3 Swarm test

The third test is designed to validate the developed UAV distribution algorithm. Thus, it checks that the system is capable to perform the mission task by using different numbers of UAVs. Three tests are performed: with one, two, and four UAVs. As an urban environment, the standard scenario shown in Fig. 13 (A) is used. The paths will be divided for all tests into four height layers. Besides the validation of the distribution algorithm, also the duration of each test is determined. As an result, it will be analyzed if the stated hypothesis, that swarms perform missions more time efficient than single UAVs, is true.

The last test performed, was the swarm test. All three tests which had each a different swarm configurations were completed successfully and the distribution of the UAVs was performed as expected. Thus, the correct functionality of the distribution algorithm has been shown. The resulting mission times of each test are provided in Table 1.

Table 1 Swam test results - duration (see Appendix E)

	1 (1 CF)	2 (2 CFs)	3 (4 CFs)
Outer Scan	$t_{1o} = 132\ s$	$t_{2o} = 86\ s$	$t_{3o} = 47\ s$
Inner Scan	$t_{1i} = 174\ s$	$t_{2i} = 184\ s$	$t_{3i} = 207\ s$
Total	$t_{1t} = 306\ s$	$t_{2t} = 270\ s$	$t_{3t} = 254\ s$

As expected, it can be seen that adding UAVs to the swarm results in shorter times for the Outer Scan and, therefore, a more efficient scanning process. In contrast, the times for the Inner Scan are increased. As explained in Section 3.4, the reason for this phenomena is that due to the limited space in the center of the city, only one UAV can observe the inner part at the same time. The change of the UAVs causes a short pause of the scanning process and results in longer mission times. For larger test scenarios or in a real environment, where more space is available, multiple UAVs could enter the city at the same time, which would result in a more efficient scans. But as the change of UAVs is performed fast, the total mission time is still decreased the more UAVs are used.

5 Conclusion and Future Work

The paper presents the development process of a mission that uses a swarm instead of a single Unmanned Aerial Vehicle (UAV) and analyzes the resulting benefits and challenges. To illustrate the benefits, a 3D mapping mission of an urban environment was chosen, as those tasks are typical applications for single UAVs nowadays. Multiple algorithms were developed and implemented in a testing environment based on a simplified building arrangement. Three tests were performed to validate that the developed algorithms are working as expected. The adaptability test confirmed that the system is adaptable to different environments and that the path planning algorithms are working as expected. The swarm tests confirmed that the distribution of the UAVs for different constellations is working properly. The tests also confirmed that the mapping mission is performed faster as more UAVs are used. Nevertheless, the test showed that there is a maximum number of UAVs that should be used due to limited space. The precision tests proved that the developed system could produce a point cloud capable of estimating the surrounding environment. However, it was also determined that due to inaccurate localization and sensor noise, the mapping precision is not good enough to generate high-resolution point clouds. The next steps are to scale the solution to an actual city with larger UAVS.

References

[1] Anam Tahir, Jari Böling, Mohammad-Hashem Haghbayan, Hannu T. Toivonen, and Juha Plosila. Swarms of unmanned aerial vehicles — a survey. *Journal of Industrial Information Integration*, 16:100106, 2019. DOI: https://doi.org/10.1016/j.jii.2019.100106.

[2] Zoltan Koppanyi and Charles Toth. Swarm mapping: Investigation on strategies to explore building interiors. 05 2017.

[3] Maurizio Prada Luca Gemma Maurizio Rossi Pietro Tosato, Daniele Facinelli and Davide Brunelli. An autonomous swarm of drones for industrial gas sensing applications. *2019 IEEE 20th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, 3, 06 2019.

[4] Swarmsense for commercial and industrial use cases. `https://www.episci.com/product/swarmsense/`, 2020 (accessed February 10, 2021).

[5] The 5 best drones for mapping and surveying. `https://pilotinstitute.com/drone-mapping/`, 2020 (accessed February 10, 2021).

[6] A history of drones. `https://plantbreedingsoftware.guru/2020/08/25/a-history-of-drones-from-humble-beginnings-to-the-impact-on-plant-breeding/`, 2019 (accessed February 10, 2021).

[7] Could these drones become the future of firefighting? `https://www.dronesglobe.com/guide/firefighting/`, 2021 (accessed February 10, 2021).

[8] The perfect swarm: Drones descend on manufacturing. `https://www.newequipment.com/plant-operations/article/22059204/the-perfect-swarm-drones-descend-on-manufacturing`, 2020 (accessed February 10, 2021).

[9] Johanna Matthaei, Thomas A. Krueger, Stefan Nowak, and Ulf Bestmann. Swarm exploration of unknown areas on mars using slam. 2013.

[10] K.N McGuire. *Indoor swarm exploration with Pocket Drones*. dissertation, TU Delft, 2019.

[11] Dr. Mostafa Abdel-Bary Ebrahim. 3d laser scanners: History, applications, and future. 10 2011.

[12] Francesco Nex and Fulvio Rinaudo. Lidar or photogrammetry? integration is the answer. *Italian Journal of Remote Sensing*, 43:107–121, 06 2011. DOI: 10.5721/ItJRS20114328.

[13] A history drone photogrammetry vs. lidar: what sensor to choose for a given application drones. `https://wingtra.com/drone-photogrammetry-vs-lidar/`, (accessed February 11, 2021).

[14] Bharat Lohani and Suddhasheel Ghosh. Airborne lidar technology: A review of data collection and processing systems. *Proceedings of the National Academy of Sciences, India Section A: Physical Sciences*, 87, 11 2017. DOI: 10.1007/s40010-017-0435-9.

[15] Christoph Strümpfel, Eric Schuster, Svenja Huschbeck, Christian Berth, and Maarten Uijt de Haag. Assured multi-mode navigation for urban operations of small uas. American Institute of Aeronautics and Astronautics (AIAA), 2020. Available Open Access acceptedVersion at https://depositonce.tu-berlin.de/handle/11303/10619. DOI: 10.2514/6.2020-1945.

[16] Victor Massagué Respall, Dmitry Devitt, and Roman Fedorenko. Unmanned aerial vehicle path planning for exploration mapping, 02 2020. DOI: 10.13140/RG.2.2.11126.75841.

[17] Mohammadreza Radmanesh, Manish Kumar, Paul Guentert, and Mohammad Sarim. Overview of path planning and obstacle avoidance algorithms for uavs: A comparative study. *Unmanned Systems*, 6:1–24, 04 2018. DOI: 10.1142/S2301385018400022.

[18] Kimon P. Valavanis and George J. Vachtsevanos. *Handbook of Unmanned Aerial Vehicles*. Springer Publishing Company, Incorporated, 2014.

[19] Shahrukh Ashraf, Priyanka Aggarwal, Praveen Damacharla, Hong Wang, Ahmad Javaid, and Vijay Devabhaktuni. A low-cost solution for unmanned aerial vehicle navigation in a global positioning system–denied environment. *International Journal of Distributed Sensor Networks*, 14:155014771878175, 06 2018. DOI: 10.1177/1550147718781750.

[20] Neil Smith, Nils Moehrle, Michael Goesele, and Wolfgang Heidrich. Aerial path planning for urban scene reconstruction: A continuous optimization method and benchmark. volume 37, pages 1–15, 12 2018. DOI: 10.1145/3272127.3275010.

[21] M. Almeida, Hanno Hildmann, and G. Solmaz. Distributed uav-swarm-based real-time geomatic data collection under dynamically changing resolution requirements. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-2/W6:5–12, 08 2017. DOI: 10.5194/isprs-archives-XLII-2-W6-5-2017.

[22] Xiaoping Huang, Rujun Cao, and Yanyan Cao. A density-based clustering method for the segmentation of individual buildings from filtered airborne lidar point clouds. *Journal of the Indian Society of Remote Sensing*, 47(6):907–921, Jun 2019.

[23] Renato Dos Santos, Mauricio Galo, and André Carrilho. Building boundary extraction from lidar data using a local estimated parameter for alpha shape algorithm. volume XLII-1, 09 2018. DOI: 10.5194/isprs-archives-XLII-1-127-2018.