A MOBILE MULTI-SENSOR PLATFORM FOR BUILDING RECONSTRUCTION INTEGRATING TERRESTRIAL AND AUTONOMOUS UAV-BASED CLOSE RANGE DATA ACQUISITION

A. Cefalu [a], N. Haala [a], S. Schmohl [a], I. Neumann [b], T. Genz [c]

[a] Institute for Photogrammetry, University of Stuttgart, Germany - (alessandro.cefalu, norbert.haala)@ifp.uni-stuttgart.de
[b] Geodetic Institute, Leibniz Universität Hannover, Germany - neumann@gih.uni-hannover.de
[c] Geo-Office, Gesellschaft für graphische Datenverarbeitung und Vermessung mbH, 14612 Falkensee - torsten.genz@geo-office.de

ICWG I/II

KEY WORDS: Mobile Multi-Sensor Platform, Flight Planning, Autonomous Flight, Laser Scanning, Building Reconstruction

ABSTRACT:

Photogrammetric data capture of complex 3D objects using UAV imagery has become commonplace. Software tools based on algorithms like Structure-from-Motion and multi-view stereo image matching enable the fully automatic generation of densely meshed 3D point clouds. In contrast, the planning of a suitable image network usually requires considerable effort of a human expert, since this step directly influences the precision and completeness of the resulting point cloud. Planning of suitable camera stations can be rather complex, in particular for objects like buildings, bridges and monuments, which frequently feature strong depth variations to be acquired by high resolution images at a short distance. Within the paper, we present an automatic flight mission planning tool, which generates flight lines while aiming at camera configurations, which maintain a roughly constant object distance, provide sufficient image overlap and avoid unnecessary stations. Planning is based on a coarse Digital Surface Model and an approximate building outline. As a proof of concept, we use the tool within our research project MoVEQuaD, which aims at the reconstruction of building geometry at sub-centimetre accuracy.

1. INTRODUCTION

One of the most popular applications of UAS photogrammetry is the substitution of standard aerial image flights while aiming at cost efficient and flexible data collection for areas of limited extent. In such scenarios UAS platforms capture nadir imagery in the framework of a standard block, typically consisting of parallel flight lines, potentially enhanced by some cross strips. Such imagery typically serves as the basis for generating DSMs, DTMs and true-ortho photos and is thus captured at a Ground Sampling Distance (GSD), which corresponds to the aspired DSM and ortho raster-width. The (constant) flying height above ground of the close-to-nadir imagery captured at straight, parallel flight lines is then simply determined by the pixel size and focal length of the used camera. Usually a rather high image overlap of 80% or 90% in flight direction and 60% to 70% cross flight direction is used to support multi-view stereo image matching and avoid occlusions. In contrast to such rather simple flight scenarios, the planning of optimal flight patterns can become much more complex if photogrammetric data collection is for example applied in the context of 3D city models. Typically, the acquisition of complex 3D objects like buildings, bridges or monuments presumes image collection from rather short distances and varying viewing directions. While data processing for UAV flights of nadir image blocks is similar to the normal case of airborne photogrammetry, such “arbitrary” block configurations relate much more to techniques that are typical for close range photogrammetry. Data processing for such scenarios frequently integrates approaches originally developed in Computer Vision. Prominent examples are Structure-from-Motion and dense multi-view stereo image matching. While traditional products like 2.5D DSM raster representations are typically generated from nadir imagery, dense matching in close range scenarios and from oblique views aims at data collection in complex 3D environments. This calls for the reconstruction of true 3D geometry represented by point clouds and triangle meshes but also depth maps and volume scalar-fields.

During retrieval of complete surfaces with high precision by dense image matching methods, the selection of suitable camera station is one of the key challenges. Since the respective image network geometry directly impacts the accuracy, as well as the completeness of the point cloud, optimal configurations have to be found to retrieve the required resolution, precision and completeness in the resulting dataset. This can be rather complex, in particular for objects with strong depth variations which are acquired at short distance. The precision of the photogrammetric measurement mainly depends on the two components - the image scale and the intersection angle. Typically, a wide angle lens is used in order to cover a large area at each station and to enable an accurate bundle adjustment. The used camera defines the pixel size, and with that the angular resolution. According to the required depth precision, image scale and intersection angle should be chosen. Small intersection angles and image scales lead to high completeness due to the high image similarity and consequently good matching performance, but also poor depth precision due to the weak geometrical conditions. In contrast, large intersection angles and large image scales provide better depth precision, but suffer from the lower image similarity. Even though small intersection angles lead to noisy results, models with small baselines should be acquired and used within the surface reconstruction. Since large baseline models have lower image similarity – which is challenging for the matching method, small baseline models are required additionally. Furthermore, highly overlapping imagery leads to high redundancy, which is beneficial for the precision in object space.

We present our work on automatic generation of suitable flight plans for architectural inspection and reconstruction within the project MoVEQuaD, which is embedded in the research

https://doi.org/10.5194/isprs-archives-XLII-2-W6-63-2017 | © Authors 2017. CC BY 4.0 License.

network FROLE[1]. The latter aims at the development of a holistic and sustainable process chain in the context of noise protection measures - ranging from noise mapping over inspection and documentation of building structures to the financial and administrative closure of a project. MoVEQuaD focusses on the efficient and complete survey and documentation of outer geometry of realty suffering from noise pollution. Modern technology at a moderate financial impact is employed to acquire data at sub-centimetre level. This includes an off-road capable quad as the core component of the terrestrial data acquisition process (Figure 1). It has been modified for transportation and employment of various surveying equipment and can be prepared for measurement, quickly. Apart from reflector, (panoramic) camera and GPS antennas, the system is equipped with an automatic levelling device for a tachymeter or laser scanner and has been designed to allow an in-situ calibration of the current platform configuration. Furthermore, a field computer for on-site processing and examination of preview results and a virtual reality environment for off-site re-visitation of the site are part of the concept.

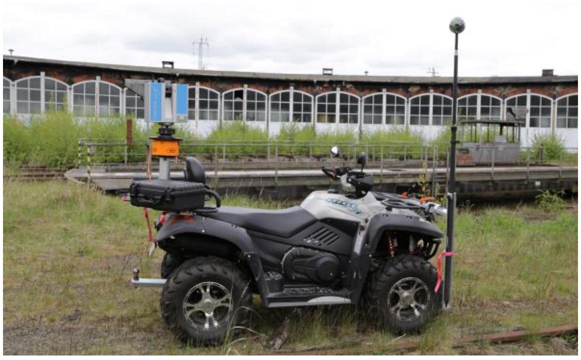


Figure 1. An off-road capable quad, equipped with a variety of sensors, is the core of terrestrial data acquisition and serves as a mobile workstation.

A low-cost quadrocopter (Phantom 4) for nadir and/or close range image acquisition complements the concept in order to guarantee completeness and quality at higher facades and roof areas. Simplification of its operation - and thereby a raise in efficiency - is achieved through a software process chain for mission planning and execution, which will be described in the remainder of this paper.

2. FLIGHT MISSION PLANNING

Tools for generation of (close to) nadir case flight missions which take these considerations into account are common and may include an adaptation to terrain shape (Gandor et al., 2015). Also geometrical shapes, e.g. circles or helixes, or free form paths can be created and flown autonomously, depending on the capabilities of - and correlating necessary financial investments for - the used hard- and software. However, objects of higher geometrical complexity, e.g. buildings, require more complex camera constellations. Inspection or reconstruction of single facades or entire building structures is of increasing interest in the domain of survey services and is required for various applications, e.g. cultural heritage preservation (Cefalu et al., 2013; Deris et al., 2017), disaster management (Achille et al., 2015), thermal (Zhang et al., 2015) and general visual inspection, to name a few.

Composing an adequate flight mission from standard flight patterns, as in (Grenzdörffer et al., 2015) is a possible approach in these scenarios. Designing the mission, however, mainly remains manual work and the resulting image configuration may not be ideal in all cases. Therefore, manual flight remains a frequently used alternative (Achille et al., 2015; Cefalu et al., 2013; Deris et al., 2017; Eschmann et al.,2013), but may put high demands on piloting skills. The pilot needs to steer to adequate positions, align the camera, take care of obstacles and make sure that all areas of the object are captured. Simultaneously, he needs to reposition himself to keep the UAV in line of sight and an overview of its surrounding. Creating a homogeneous camera distribution in such a situation may become even more difficult, when a constant time interval is used to trigger the camera. Often, two persons are required to safely and efficiently carry out the task. Figure 2 depicts a comparatively complex building structure which will serve as example in the remainder of this paper. It is located on a partially abandoned train station and is in some areas surrounded by strong vegetation and uneven ground, which additionally complicates the situation. The distribution of images acquired of the building, using a manually piloted Phantom 4 in time interval triggering mode, exhibits clusters as well as missing areas (see section 4 for more details and figures). Further, the camera has not always been pointing towards the object and the distance to the object could not be held constant.

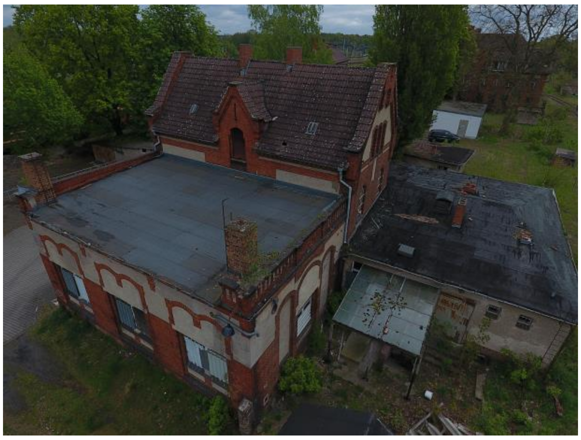


Figure 2. A rather complex building structure in difficult surrounding. Strong vegetation and uneven ground complicate the data acquisition.

The example demonstrates the need for an improved image acquisition process in the case of architectural inspection or reconstruction. Research on this specific task has e.g. been conducted in (Daftry et al., 2015), where near-real-time reconstruction is performed and an online indication of redundancy supports the pilot during manual flight. (Nieuwenhuisen & Behnke, 2016) describe a volumetric approach to autonomously navigate an UAV along camera stations for building mapping. The used UAV is equipped with a variety of sensors, enabling navigation between specific mission waypoints on two levels - a global routing based on prior knowledge represented as a static map and a local rerouting to avoid dynamic or unknown obstacles observed by the sensors. However, the definition of mission relevant waypoints is left to an operator.

Similarly, our work bases on a volumetric representation of the surrounding of a building, which in our case is given by a 2.5D DSM and 2D polygons representing building ground plans and no-trespass areas. In contrast to (Nieuwenhuisen & Behnke,

[1] Flugrobotereinsatz zur Objektdatenerfassung für Lärmschutz und energetische Sanierung

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLII-2-W6-63-2017 | © Authors 2017. CC BY 4.0 License.

2016), we automatically derive flight paths and camera stations from the input data. A flight assistant app for mobile devices supports the pilot during the execution of the flight mission, reducing the pilot's workload to supervising the flight and, if necessary, applying simple corrections to the overall trajectory. Following basic photogrammetric principles, the flight mission planning tool was developed to provide a camera station configuration which (a) maintains a roughly constant distance to the object, while (b) aligning the optical axis of the camera to perpendicularly point towards the object surface. Furthermore, for neighbouring camera stations the configuration should (c) provide sufficient image overlap while avoiding identical stations and (d) avoid strong changes in the viewing direction. Additionally, the final flight trajectory should result in a safe, intuitive and easy-to-supervise behaviour of the drone. We achieve this by extracting two-dimensional flight tracks at different height levels from a volumetric representation of the building's surrounding. The separate tracks are fused to a single flight mission by intermediate linking manoeuvres. The implementation of the mission planning has been carried out using Matlab.

2.1 Volumetric Map, Scalar & Vector Fields

A georeferenced 2.5D DSM H, along with a 2D polygon B describing the building contours serve as main data input (Figure 3). We generate a volumetric occupancy map M of the environment of the building which classifies voxels (volume elements) of user-defined size into the classes *free space*, *object of interest*, and *obstacle* (Figure 4). Optionally, an additional set of polygons N may be used during map generation to define no-trespass areas. This option allows compensating for unreliably reconstructed areas in the DSM, e.g. poles, lanterns, vegetation, etc.

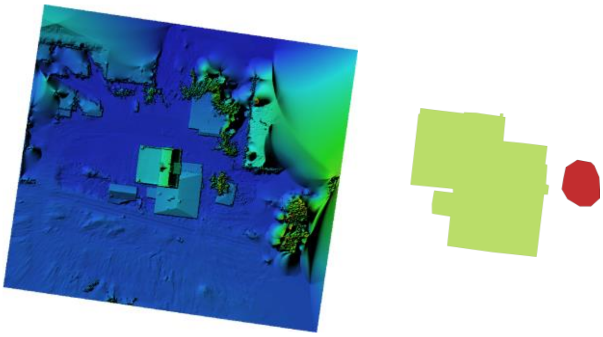


Figure 3. A DSM H (left) and a polygon B representing the building contour (green) are used as main data input for mission planning. Optionally, polygons N defining no-trespass areas can be used. Here, a single polygon (red) is used to mask an imprecisely reconstructed tree close to the building.

The horizontal extent of the map is derived from the building polygon's bounding box, enlarged by a predefined buffer size. The minimum and maximum values of the height map in this area define the vertical extent. Here, the buffer is applied to the top, only. If the horizontal position of a voxel falls into any polygon of N it is classified as *obstacle*. Otherwise, if its lower bound is above H at the corresponding location, we consider it to be *free space*. The remainder of voxels are considered to be occupied and classified as *object of interest*, if their horizontal locations fall into B, or *obstacle*, respectively. The volumetric map allows computing two three-dimensional scalar fields holding distance measures.

The field O holds the distances of voxels to the nearest voxel of the class *object of interest* (Figure 5). Isosurfaces within this scalar field represent a surface at constant distance from the building. Camera stations should be distributed on such a surface, according to the desired GSD. It may be thought of as a buffered (or dilated) and thereby smoothed copy of the building's surface, while the degree of smoothing depends on the chosen distance. Accordingly, the three dimensional gradient field V of O represents the smoothed normal vector directions of the building surface and is negated to determine adequate viewing directions for the camera at different locations in space. A second scalar field S holds the distances of voxels to the nearest occupied neighbour, i.e. the nearest voxel which is not of the class *free space* (Figure 6). Using a user-defined safety distance, a corresponding isosurface in this scalar field partitions the space into safe and unsafe flight areas, of which the latter should be subtracted from the putative flight surface.

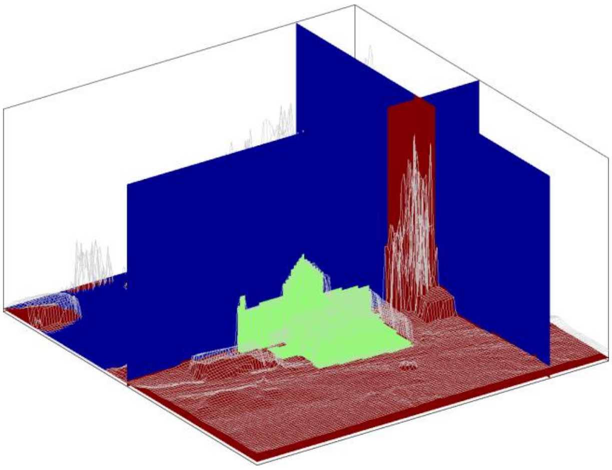


Figure 4. The volumetric map M segments space into the classes *object of interest* (green), *obstacle* (red) and free space (blue). No-trespass areas N create vertical obstacle areas. The DSM H is indicated as grey mesh.

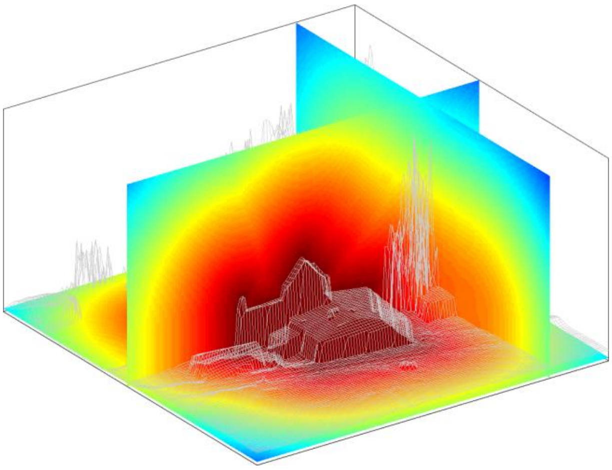


Figure 5. The three-dimensional scalar field O holds distances to the *object of interest* (close to far, depicted as red to blue). Its negated three-dimensional gradients are stored in the vector field V and used as camera viewing directions.

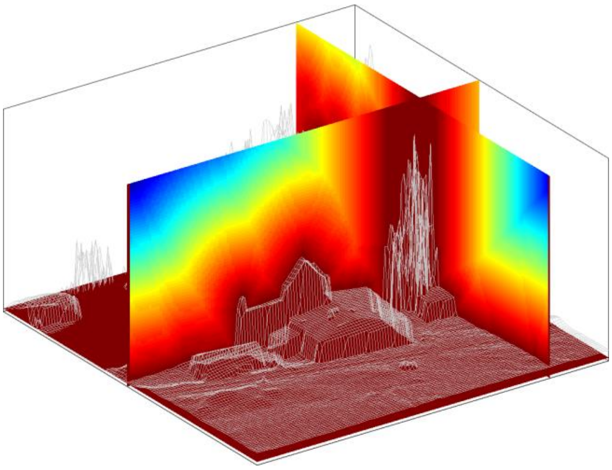


Figure 6. The three-dimensional scalar field S holds distances to occupied space (from close to far, depicted as red to blue). It is used for obstacle avoidance during mission planning.

2.2 Horizontal Paths & Camera Station Distribution

As our desired flight behaviour aims at mainly orbiting horizontally around buildings, we may reduce the problem of camera distribution to separate two-dimensional tasks at different height levels. According to the given camera parameters, desired image overlap and GSD, we derive a vertical step size, at which corresponding Z-layers of the scalar and vector fields are extracted. In every layer, we first determine the isolines in O at the appropriate flight distance. The result may be an arbitrary number of (usually) closed curves, which represent a set of putative flight paths p (Figure 7, left). Analogously, we extract the border between safe and unsafe space $\boldsymbol{b}$, as the isoline corresponding to the safety distance in S (Figure 7, right). In case of intersections between both sets of curves, we segment accordingly. Parts of p passing through unsafe areas are removed. If possible, the corresponding segments of b are used as replacement to reroute the UAV (Figure 8). A user-defined threshold restricts these manoeuvres to short travel distances. In cases where two bypass routes are possible, the one closer to the object is chosen.

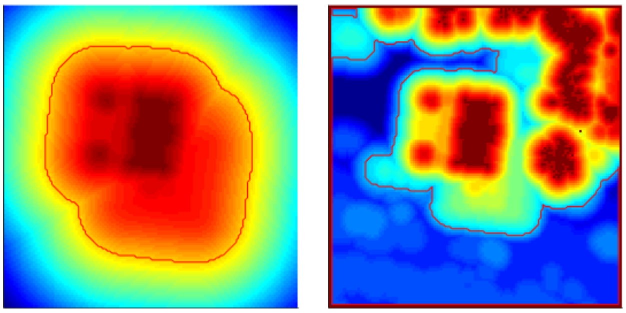


Figure 7. A putative path p at constant distance from the building, extracted as isoline in a Z-layer of O (red curve, left). The border of traversable space b is extracted as isoline at the safety distance from S in the same layer (red curve, right).

The result of the process is a set of arbitrarily shaped paths at different height levels, though several paths may exist on the same height level. Every path is separately converted into a viewpoint trajectory t, by distributing camera stations along the segments. We set a first camera station at the starting point of a path segment and scan the curve for the next node at which either a distance threshold (according to the desired overlap) or a maximum angular change in viewing direction is exceeded. A new camera station is set accordingly. The process is repeated until the end of the segment is reached, while testing the thresholds on the current last camera station. As the camera stations are directly used as waypoints, the test further includes a traversability test. In very rare cases, intermediate curve nodes are kept as pure waypoints (no image acquisition) to restrict to traversable space.

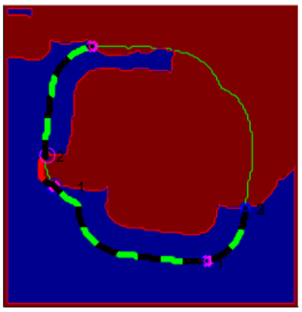


Figure 8. Parts of p (green curve) passing through unsafe areas (dark red) are removed. In feasible cases (middle left), the path is rerouted following the border of traversable space b (red curve). The final paths on a height level are composed from save segments. Here, a single path is created (bold dashed line).

2.3 Trajectory Fusion

The separate trajectory segments created so far could be used as single missions and flown by the UAV from start to end point or in the reverse order. However, we need to fuse them to a single final flight plan. Our strategy is to use the trajectories as a whole and connect start and end points via simple linking manoeuvres. As the trajectories can be approached from both ends, we refer to these points as entry and exit points. Whenever a new trajectory is approached using its end point as entry point, the trajectory is added to the flight plan in reverse order. Otherwise it is added unaltered.

The user indicates a probable take-off area in the DSM. This selection is not part of the final flight plan but is used to select the trajectory with the nearest entry point. The corresponding trajectory initialises the flight plan and is removed from the set of trajectories. We proceed iteratively until all trajectories have been added to the flight plan by first, computing manoeuvres and corresponding travel costs c (1) from the exit point (current last point in flight plan) to all remaining entry points. Second, the manoeuvre with the lowest cost is appended to the mission. Finally the corresponding trajectory is added to the plan and removed from the set of trajectories.

Linking Manoeuvres: As our representation of obstacles is of 2.5D nature, we may consider the vertical space (i.e. a column of voxels) above any camera station to be freely traversable. Depending on the horizontal distance between the exit and entry point, we define two types of manoeuvres, constructed from a horizontal and two vertical path segments - s_h, s_{v_1} and s_{v_2}, respectively. In cases of $\|s_h\| > 0$, we identify a height at which the two columns can be connected by a direct horizontal flight path through free space (Figure 9). Starting at the higher of the two stations we test for traversability and incrementally increase the height for the test until a valid solution is found. If the upper border of the volumetric map is exceeded during testing, we set the height of the manoeuvre to a user-defined safe height, at which safe traversing must be guaranteed at all times. If the two points have identical horizontal location, i.e. $\|s_h\| = 0$, a direct vertical connection can be applied, leading to $\|s_{v_2}\| = 0$.

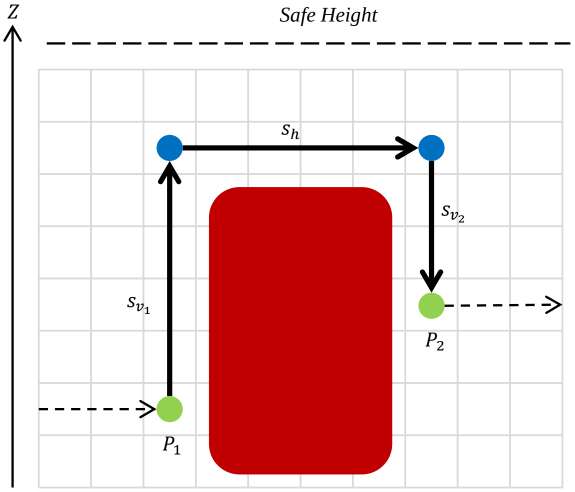


Figure 9. Linking exit point P_1 and entry point P_2 of two trajectories. Obstacles (red) are avoided through vertical segment s_v and a horizontal segment s_h. A single vertical segment can be used to connect points with identical horizontal location.

Travel costs: The cost function (1) used to evaluate the manoeuvres and select the most suitable next entry point is designed to favour horizontal over vertical movement. Vertical travel distances are penalised by a factor λ. λ is set to one of two levels, depending on whether exit and entry points are at same height (2). This allows differentiating numerically between vertically bypassing an obstacle in order to proceed at the same height or switching to another trajectory layer.

$$c = \|s_h\| + \lambda\left(\|s_{v_1}\| + \|s_{v_2}\|\right) \tag{1}$$

with

$$\lambda = \begin{cases} \lambda_1 \gg 1, & Z_{P_1} = Z_{P_2} \\ \lambda_2 \gg \lambda_1, & Z_{P_1} \neq Z_{P_2} \end{cases} \tag{2}$$

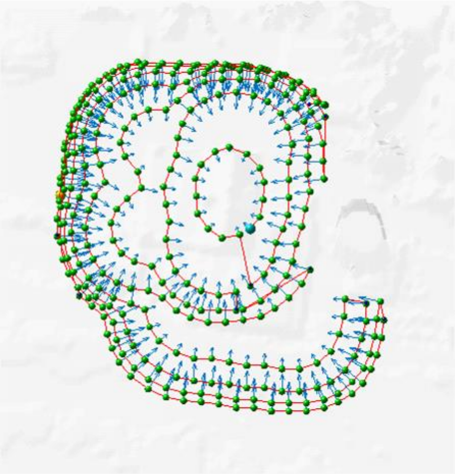
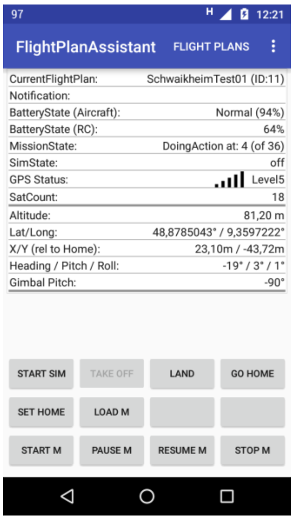


Figure 10. Top view of a flight plan. Green spheres with blue arrows indicate camera stations and corresponding camera alignment.

The final flight plan waypoints (Figure 10) are transformed to WGS84 longitude and latitude. A binary indicator states whether an image needs to be taken at a certain location. Viewing direction vectors are expressed in two angles, azimuth and elevation (0° = horizontal, -90° = nadir). The values are stored in a simple ASCII exchange file and can be copied to a mobile device for execution of the mission using the flight assistant app.

3. FLIGHT PLAN ASSISTANT APP

A custom android app (Figure 11) was developed to particularly satisfy the needs of this project, which are primarily: assembling a flight mission from given waypoints and viewpoints (flight plan), transferring it to an UAV and controlling the execution. Other solutions available on the market have been lacking certain features, many apps being solely designed for nadir flights. We used *DJI Mobile SDK for Android v3.5.1*, a Java library acting as an API for android apps to communicate with their aircrafts and handheld devices.



CurrentFlightPlan:	SchwaikheimTest01 (ID:11)
Notification:	
BatteryState (Aircraft):	Normal (94%)
BatteryState (RC):	64%
MissionState:	DoingAction at: 4 (of 36)
SimState:	off
GPS Status:	Level5
SatCount:	18
Altitude:	81,20 m
Lat/Long:	48,8785043° / 9,3597222°
X/Y (rel to Home):	23,10m / -43,72m
Heading / Pitch / Roll:	-19° / 3° / 1°
Gimbal Pitch:	-90°

Figure 11. Screenshot of our flight assistant app.

Within the SDK, the concept of so called *waypoint missions* has been employed for the current version of the application. Sets of waypoints / viewpoints can be uploaded to the drone as a whole and are executed accordingly. Simple built-in safety and convenience mechanisms as *failsafe* or *goHome* can be assessed and parameterised easily. These functions assume a horizontal plane in the airspace above of which the UAV can fly freely, similarly to the concept of a safe height used during flight planning. In order to manoeuvre the UAV from the home (take-off / landing) position to the first waypoint of the waypoint mission or from the last waypoint back to the home position, the UAV rises to this predefined *feeder zone* (Figure 12) and approaches the desired horizontal location before sinking to its target position.

While executing a flight plan, the progress, i.e. the last processed viewpoint, is stored. This allows picking up the execution of the flight plan at any intermediate station in case of an interruption. These situations may either occur due to technical reasons (e.g. loss of signal, low battery) or to the fact that the maximum number of waypoints is limited to 99 (of which four are used within the feeder zone). In the latter case, the return manoeuvre may be interrupted and the mission can be reinitialised without much delay with a new part of the flight plan, while the drone hovers in the air.

https://doi.org/10.5194/isprs-archives-XLII-2-W6-63-2017 | © Authors 2017. CC BY 4.0 License.

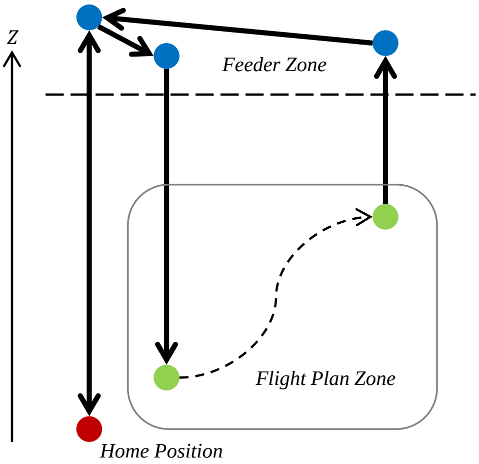


Figure 12. Sketch of the spatial concept of the feeder zone. First and final stations of a mission are horizontally approached at free airspace area.

A manual interruption and reinitialisation also proves helpful in order to compensate for weak self-localization capabilities of drones as the Phantom 4, which has been used in our experiments. Particularly, the lack of RTK-GNSS and unreliable barometric altitude measurements impose the need for a correction mechanism. Here, the app offers the possibility to set offset values, which are applied to the flight plan coordinates before uploading, i.e. the UAV trajectory is shifted as a whole. Preferably, the drone is positioned over a point with known coordinates before take-off to determine the offsets. However, the mission can be interrupted at any time to alter the values. Collision avoidance sensors are only provided in nadir and front direction of the drone. A semi-automatic mode supports the pilot in approaching critical situations. When activated, the pilot can influence the speed at which the waypoints are approached, using one of the remote control's sticks. Thereby, the drone can be manoeuvred back and forth on the trajectory.

In total, these simple yet effective features enable the user to safely operate the UAV in order to efficiently execute rather complex flight plan.

4. EXPERIMENTS AND DATA ANALYSIS

Within the project, the mapping of the UAV is analysed together with the terrestrial data. The terrestrial data consists of terrestrial laser scanning (TLS) and GNSS measurements. These additional measurements are carried out in a static mode using the vehicle shown in Figure 1. For this purpose, the off-road quad is positioned at optimal points around the object. The TLS measurements are initially registered by a plane based method using the software Scantra. Possibilities for a joint bundle adjustment employing airborne images, terrestrial laser scans and GNSS measurements are investigated, though not implemented at present. Further, a quality model for the joint 3D point cloud is currently developed that should provide empirical quality parameters, i.e. for the precision of the fused data. Furthermore, other quality measures, e.g. like accuracy and completeness should be considered in future.

Figure 13 shows the standard deviation as a measure for the precision of the TLS point cloud, i.e. the Helmert point uncertainty, calculated according to (3). An intensity based approach for the TLS distance measurements according to (Wujanz et al., 2017) is used. The angle uncertainties of the TLS are then added according to the manufacturer specifications, here for a Zoller+Fröhlich (Z+F) Imager 5006 (Z+F GmbH, 2017).

$$\sigma_{XYZ} = \sqrt{\sigma_X^2 + \sigma_Y^2 + \sigma_Z^2} \tag{3}$$

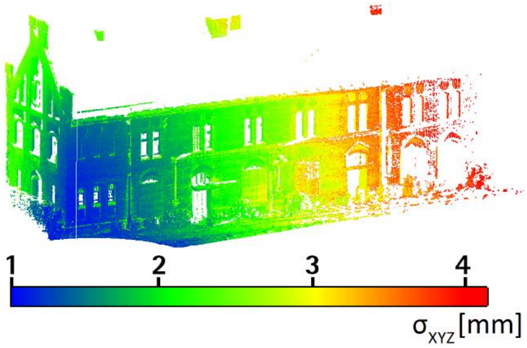


Figure 13. Example of estimated standard deviation of a 3D point (Helmert point uncertainty) for the precision of TLS measurements.

The site depicted in Figure 2 has been revisited in order to test the flight planning tool in conjunction with the flight assistant app and compare the results against TLS measurements. The acquired images have been processed using the software packages PhotoScan (PS) and RealityCapture (RC) for comparison. However, we will not discuss software differences, but present a mixture of the results. For testing reasons, three separate flight plans at different GSD levels (sub-centimetre), overlap settings (80% to 90%) and voxel sizes (0.3 to 0.5m) have been executed, of which one adds an additional side building. All UAV images have been processed jointly (Figure 14).

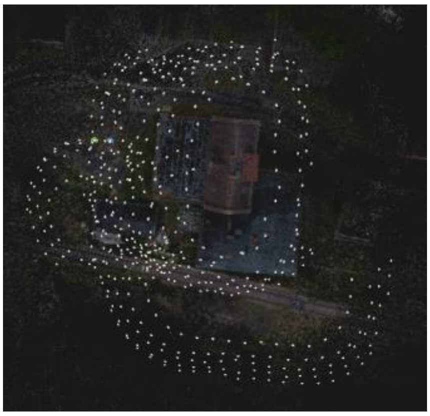


Figure 14: Bird's eye view of the sparse reconstruction (RealityCapture) of the building depicted in Figure 2. Here, the image acquisition was planned and executed using the described software tools.

Figure 17 compares the resulting image distribution and connectivity to the manually acquired data (visualised with RC). The distribution is homogeneous and covers the structures completely, while being well aligned towards the surfaces. Further, though covering a larger area, the number of captured images has been reduced from 776 for the manual flight with time interval triggering to 527 when executing our planned flights. The reduction of more than 30% in the number of images has corresponding effects in processing time, memory consumption and necessary storage volume. Moreover, these image numbers already take into account that images have been removed by the operator for the manual flight, whereas no such measure is necessary for a planned flight with selective camera triggering. A comparison of time efficiency, however, is difficult, as the planned images were acquired in a test situation on a rather windy day. Neglecting interruptions, the average time interval between images is 4.9s in comparison to 2.9s for the manual flight. Extrapolation to the image number results in ~43min compared to ~38min. De facto, a few interruptions are necessary to find good correction values for the trajectory, especially for the altitude, where the drone's self-localization deficiencies are most apparent. However, considering the larger area covered by the planned flights and the superior image distribution, we regard the test to be successful. Figure 15 depicts a meshed reconstruction using RC.

Seven ground control points have been used for georeferencing (PS) exhibiting a horizontal error of 1.6cm and a vertical error of 0.8cm, thereby being in the expected range. The vast majority of the sparse feature points in the area of the buildings have been tracked through ten or more images.

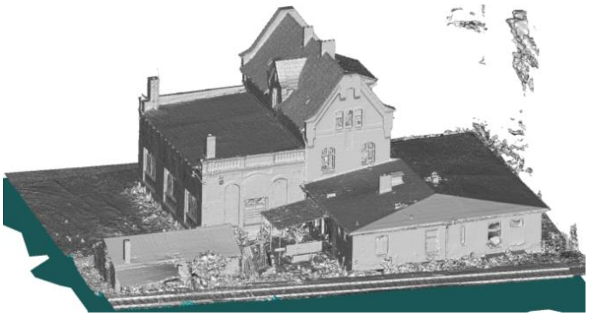


Figure 15. Meshed surface of the building, generated from UAV images, solely. A precise, homogeneous and complete reconstruction could be achieved with the well distributed images of our flight planning and execution concept.

The Helmert point error of the TLS data in the area of the building is of comparable magnitude to the example of Figure 13. In order to empirically judge the results of the image based reconstruction without influence of possible georeferencing errors, the datasets have been fitted using the ICP (iterative closest points algorithm) of the software CloudCompare. The resulting cloud-to-mesh distances for a facade are shown in Figure 16. The standard deviation of roughly 6mm successfully fulfils the demands of the project.

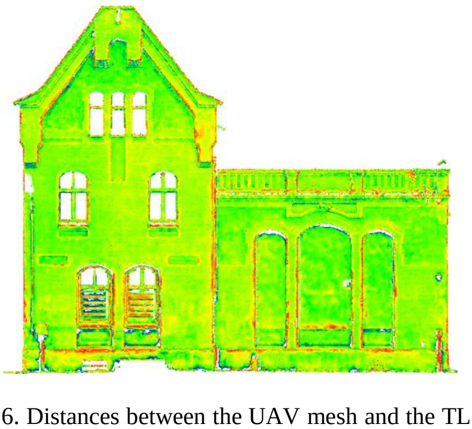


Figure 16. Distances between the UAV mesh and the TLS point cloud (red/blue correspond to +/-2cm) after an ICP fit for one facade. The achieved standard deviation is 6mm.

5. CONLUSIONS

We have presented our work on automated generation of flight plans for architectural inspection and reconstruction within the project MoVEQuaD. The flight planning approach derives well distributed camera stations from a volumetric representation of the building environment. An additional app for mobile devices assists in execution of the corresponding flight missions and thereby allows for rather complex flight patterns with a low-cost UAV, which lacks high-end features as RTK-GNSS or sophisticated obstacle avoidance. We have successfully tested this complementary approach in a real-world situation and evaluated the process by comparing the resulting reconstruction to TLS. The deviations are well below a centimetre and thereby in the desired range. Future developments on the flight planning component will include the input of three-dimensional data, allowing to pass below obstacles as power lines, which is not possible at present. Further, joint orientation and georeferencing of the various data sources employed in such surveying scenarios is topic of our ongoing research.

ACKNOWLEDGEMENTS

The presented research was performed as a part of the collaborative project "A mobile virtual mapping platform and mission management for the quality analysis and quality visualization of objects in 3D (MoVEQuaD)" between the company Geo-Office GmbH from Falkensee as well as the Geodetic Institute of Leibniz Universität Hannover and Institute of Photogrammetry of Universität Stuttgart, which is funded by the German Federal Ministry for Economic Affairs and Energy (BMWi) in the Central Innovation Programme for SMEs (ZIM Kooperationsprojekt, 16KN035280, 16KN035225 and 16KN035226).

REFERENCES

Achille, C., Adami, A., Chiarini, S., Cremonesi, S., Fassi, F., Fregonese, L., & Taffurelli, L., 2015. UAV-based photogrammetry and integrated technologies for architectural applications—Methodological strategies for the after-quake survey of vertical structures in Mantua (Italy). *Sensors*, *15*(7), 15520-15539.

Cefalu, A., Abdel-Wahab, M., Peter, M., Wenzel, K., & Fritsch, D., 2013. Image based 3D Reconstruction in Cultural Heritage Preservation. In *ICINCO (1)* (pp. 201-205).

Daftry, S., Hoppe, C., & Bischof, H., 2015. Building with drones: Accurate 3d facade reconstruction using mavs. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on* (pp. 3487-3494). IEEE.

Deris, A., Trigonis, I., Aravanis, A., & Stathopoulou, E. K., 2017. Depth Cameras on UAVs: a First Approach. *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 231-236.

Eschmann, C., Kuo, C. M., Kuo, C. H., & Boller, C., 2013. High-resolution multisensor infrastructure inspection with unmanned aircraft systems. *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, (2), 125-129.

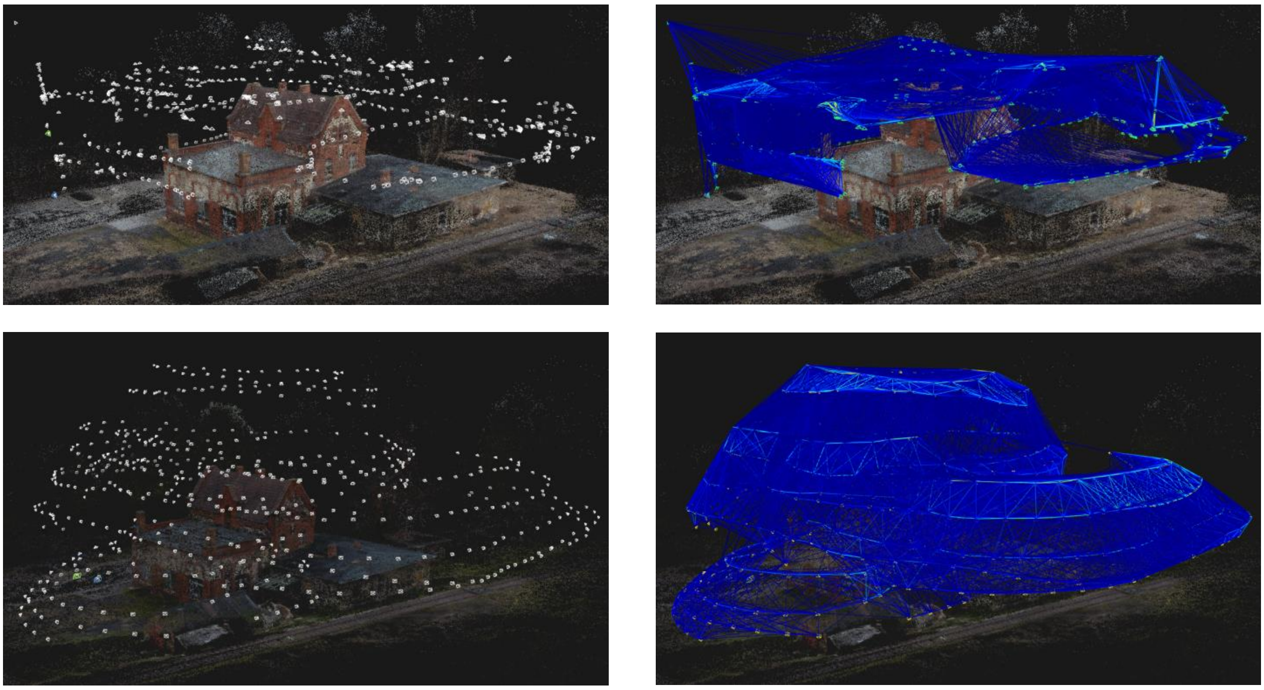


Figure 17: Comparison of image configurations resulting from a manually piloted attempt using time interval camera triggering (top) and from tests using our flight planning and assistance tools (bottom). In the former case, distances and viewing directions were not realised adequately. The distribution of camera stations (left) exhibits holes as well as clusters and results in suboptimal image connectivity (right). Using our approach, the results are significantly improved.

Gandor, F., Rehak, M., & Skaloud, J., 2015. Photogrammetric Mission Planner for RPAS. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, *40*(1), 61.

Grenzdörffer, G. J., Niemeyer, F., & Frank, A., 2015. Symbiosis of Uas Photogrammetry and Tls for Surveying and 3d Modeling of Cultural Heritage Monuments-a Case Study about the Cathedral of ST. Nicholas in the City of Greifswald. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, *40*(1), 91.

Nieuwenhuisen, M., & Behnke, S., 2016. Layered mission and path planning for MAV navigation with partial environment knowledge. In *Intelligent Autonomous Systems 13* (pp. 307-319). Springer, Cham.

Wujanz, D.; Burger, M.; Mettenleiter, M.; Neitzel, F., 2017. An intensity-based stochastic model for terrestrial laser scanners. In *ISPRS Journal of Photogrammetry and Remote Sensing* 125, S. 146–155. DOI: 10.1016/j.isprsjprs.2016.12.006.

Zhang, J., Jung, J., Sohn, G., & Cohen, M., 2015. Thermal Infrared Inspection of Roof Insulation Using Unmanned Aerial Vehicles. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, *40*(1), 381.

Zoller+Fröhlich GmbH, 2007. Technical data sheet for the IMAGER 5006. Version 1.0.5.