Journal of Field Robotics, published online 2015, DOI: 10.1002/rob.21603

Multi-Layered Mapping and Navigation for Autonomous Micro Aerial Vehicles

David Droeschel, Matthias Nieuwenhuisen, Marius Beul,
Dirk Holz, Jörg Stückler, and Sven Behnke
Autonomous Intelligent Systems Group
University of Bonn
Bonn, Germany
{droeschel, nieuwenh, mbeul, holz, stueckler, behnke}@ais.uni-bonn.de

Abstract

Micro aerial vehicles, such as multirotors, are particularly well suited for the autonomous monitoring, inspection, and surveillance of buildings, e.g., for maintenance or disaster management. Key prerequisites for the fully autonomous operation of micro aerial vehicles are real-time obstacle detection and planning of collision-free trajectories. In this article, we propose a complete system with a multimodal sensor setup for omnidirectional obstacle perception consisting of a 3D laser scanner, two stereo camera pairs, and ultrasonic distance sensors. Detected obstacles are aggregated in egocentric local multiresolution grid maps. Local maps are efficiently merged in order to simultaneously build global maps of the environment and localize in these. For autonomous navigation, we generate trajectories in a multi-layered approach: from mission planning over global and local trajectory planning to reactive obstacle avoidance. We evaluate our approach and the involved components in simulation and with the real autonomous micro aerial vehicle. Finally, we present the results of a complete mission for autonomously mapping a building and its surroundings.

1 Introduction

Micro aerial vehicles (MAVs) are enjoying increasing popularity. Due to their low cost and flexibility, they are used for aerial photography, inspection, surveillance, and search and rescue (SAR) missions. In most cases, a human operator pilots the MAV remotely to fulfill a specific task or the MAV is following a predefined path of GPS waypoints in an obstacle-free altitude. Especially for search and rescue operations—where first responders need to obtain an overview of a site and further inspect certain regions—not all parts of the environment may be reachable by humans or ground robots. For example, collapsed staircases and elevators being out-of-order may hinder accessing upper stories of a building. Here, MAVs allow to quickly inspect areas otherwise inaccessible. However, especially in collapsed buildings or larger sites, a constant connection to the MAV may not be maintainable. Also, especially within buildings, passages may be narrow and surrounding (collapsed) environmental structures may be hard to perceive for a human operator. In order to safely navigate in such surroundings, an alternative is to have a (semi-)autonomous MAV that can on its own—and without interaction with the operator—solve a well-defined sub-task. For example, the operator may specify a set of regions within or around a building and the MAV autonomously approaches all regions and collects (and/or transmits) sensor information.

Key prerequisites for the autonomous operation of micro aerial vehicles are real-time obstacle detection and planning of collision-free trajectories. Fundamental aspects in the implementation of such a system

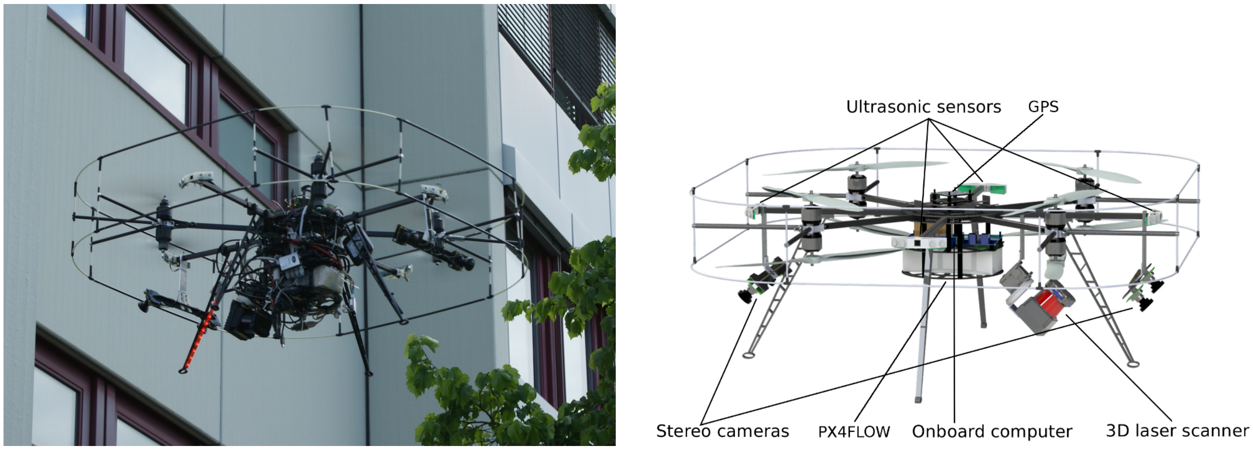


Figure 1: Our MAV is equipped with eight co-axial rotors and a plurality of sensors, including a continuously rotating 3D laser scanner, two stereo camera pairs, ultrasonic sensors, and GPS.

are robustness (all obstacles need to be reliably detected and mapped while avoiding false positives) and real-time applicability (everything needs to be sensed and processed on-board). In this article, we present a complete integrated system consisting of an MAV with multimodal omnidirectional sensor setup (see photo and labeled CAD-model in Figure 1) and a navigation approach tailored to the special needs of MAVs. We follow a multi-layered navigation approach: from slower deliberative to fast reactive layers, including mission planning, global and local path planning, fast local obstacle avoidance, and robust motion controllers. Each layer uses and builds its own environment representation: allocentric maps and city models for global path and mission planning and local egocentric obstacle maps for local trajectory planning and reactive collision avoidance.

Although we aim at mapping the environment during execution, prior knowledge can aid our mission planning—in contrast to fully autonomous exploration of unknown space. We incorporate 3D city models as acquired by land surveying authorities, i.e., a Level-of-Detail 2 (LoD2) model containing footprint, height, and roof-shape of buildings (Gröger et al., 2008) and a digital elevation model (DEM) of the environment. These models do not include smaller structures, which constitute a collision hazard for the MAV, however. Previously acquired maps can be outdated, especially in disaster situations and other dynamic environments. Hence, the initial mission plans need to be adjusted on the fly, whenever more information becomes available during flight. Nevertheless, buildings are often the largest obstacles and might inhibit local path planners to find a feasible path towards the global goal. Other obstacles, e.g., power poles, vegetation, debris, or building attachments, are likely to be small enough such that ways around them are included in our local free space and obstacle map, built by means of efficient multiresolution scan registration. Hence, a globally consistent path enables a local planner to navigate towards a global goal.

Designing sensory systems and perception algorithms is challenging for MAVs due to their size and weight constraints and their limited computing power. In order to enable navigation in difficult 3D environments for autonomous MAVs, we developed a small and lightweight continuously rotating 3D laser scanner that measures distances of up to 30 m in almost all directions. It consists of a Hokuyo UTM-30LX-EW 2D laser range finder (LRF), which is rotated by a servo actuator to gain a 3D field-of-view (FoV), as shown in Figure 3. Additionally, our MAV is equipped with two stereo camera pairs and ultrasonic sensors, covering the volume around the MAV up to 6 m range (Holz et al., 2013). All these sensors have only local precision. This is reflected in the local multiresolution property of our MAV-centric obstacle map, which has constant memory and computational requirements, independent of the environment size. The local navigation planner operates directly on this representation. We employ 3D local multiresolution path planning, extending ideas from our prior work (Behnke, 2004). This efficient planning technique allows for frequent replanning, which makes 3D navigation in dynamic, unpredictable environments possible.

The MAV-centric local maps are used to build a globally consistent map of the environment. While we assure real-time in the processing pipeline for the local maps, the global map can be built asynchronously or even on a different computer (e.g., a ground station).

In this article, we present results of complete autonomous mapping missions and give a complete system overview including the sensor setup and the involved perception and planning components. Building a fully integrated system with the given requirements led to two main advances over the state-of-the-art:

1. We introduce a local multiresolution map structure and a scan registration method to efficiently build accurate 3D maps. These maps are used for navigation and to build globally consistent environment representations.
2. We present an efficient multi-layered planning approach for MAV navigation that generates collision free trajectories based on coarse prior knowledge and the aforementioned global and local maps.

The remainder of this article is organized as follows: after a discussion of related work in the next section, we introduce our MAV in Section 3. Section 4 describes our approach to local and global mapping. Our hierarchical control architecture from global mission and path planning to low-level obstacle avoidance is detailed in Section 5. Finally, we present the results of complete mapping missions and of experimental evaluations of the involved components in Section 6.

2 Related Work

The application of MAVs varies especially in the level of autonomy—ranging from basic hovering and position holding (Bouabdallah et al., 2004) over trajectory tracking and waypoint navigation (Puls et al., 2009) to fully autonomous navigation (Grzonka et al., 2012).

2.1 Obstacle Perception

Particularly important for fully autonomous operation is the ability to perceive obstacles and to avoid collisions. Obstacle avoidance is often neglected, e.g., by flying in a sufficient height when autonomously flying between waypoints.

Due to the limited payload of MAVs, most approaches to obstacle avoidance are camera-based (Mori and Scherer, 2013; Ross et al., 2013; Schmid et al., 2014; Magree et al., 2014; Tripathi et al., 2014; Flores et al., 2014; Schauwecker and Zell, 2014; Park and Kim, 2014). Approaches using monocular cameras to detect obstacles require movement in order to perceive the same surface points from different perspectives for being able to triangulate depth. In order to estimate depth of object points instantaneously, stereo camera pairs are used on MAVs, e.g., in the works of Schmid et al. (2014) and Park and Kim (2014). Tripathi et al. (2014) use stereo cameras for reactive collision avoidance. A particularly small and lightweight (only 4 g) stereo system was developed in the DelFly project (De Wagter et al., 2014). Indoors, a popular approach for depth perception is to use RGB-D cameras that can measure depth even on textureless surfaces by projecting an infrared pattern (Bachrach et al., 2012; Flores et al., 2014).

The limited field of view of cameras poses a problem when flying in constrained spaces, where close obstacles can surround the MAV. To overcome these limitations, some MAVs are equipped with multiple (stereo) cameras. Schauwecker and Zell (2014) use two pairs of stereo cameras, one oriented forward, the other backward. Moore et al. (2014) use a ring of small cameras to achieve an omnidirectional view in the horizontal plane, but rely on optical flow for velocity control, centering, and heading stabilization only.

Other groups use 2D laser range finders (LRF) to localize the MAV and to avoid obstacles (Grzonka et al., 2012), limiting obstacle avoidance to the measurement plane of the LRF, or combine LRFs and visual obstacle detection (Tomić et al., 2012; Huh et al., 2013; Jutzi et al., 2014). Still, their perceptual field is limited to the apex angle of the stereo camera pair (facing forward), and the 2D measurement plane of the scanner when flying sideways. They do not perceive obstacles outside of this region or behind the vehicle. We allow omnidirectional 4D movements of our MAV, thus we have to take obstacles in all directions into account. Another MAV with a sensor setup that allows omnidirectional obstacle perception is described by Chambers et al. (2011).

Cameras and LRFs have problems detecting transparent obstacles such as windows. In contrast, ultrasonic distance sensors can reliably detect such obstacles. On MAVs, ultrasonic sensors are often used to measure height (Honegger et al., 2013). Ultrasonic sensors have a wide measurement cone. Thus only few of these cheap and lightweight sensors are required to span a complete ring or sphere around the MAV in order to detect obstacles in the close vicinity of the MAV, as done by Becker et al. (2012).

For mobile ground robots, 3D laser scanning sensors are widely used—due to their accurate distance measurements, even in bad lighting conditions, and their large FoV. For instance, autonomous cars often perceive obstacles by means of a rotating laser scanner with a 360° horizontal FoV, allowing for detection of obstacles in every direction (Montemerlo et al., 2008). Up to now, such 3D laser scanners are rarely used on lightweight MAVs—due to payload limitations. Instead, two-dimensional LRFs (Tomić et al., 2012; Grzonka et al., 2009; Bachrach et al., 2009; Shen et al., 2011; Grzonka et al., 2012; Huh et al., 2013) are used. Statically mounted 2D LRFs restrict the FoV to the two-dimensional measurement plane of the sensor. However, this poses a problem especially for reliably perceiving obstacles surrounding the MAV. When moving and in combination with accurate pose estimation, these sensors can very well be used to build 3D maps of the measured surfaces, though. Fossel et al., for example, use Hector SLAM (Kohlbrecher et al., 2011) for registering horizontal 2D laser scans and OctoMap (Hornung et al., 2013a) to build a three-dimensional occupancy model of the environment at the measured heights (Fossel et al., 2013). Morris et al. follow a similar approach and in addition use visual features to aid motion and pose estimation (Morris et al., 2010). Still, perceived information about environmental structures is constrained to lie on the 2D measurement planes of the moved scanner.

In contrast, we use a continuously rotating LRF, that does not only allow for capturing 3D measurements without moving, but also provides omnidirectional sensing at comparably high frame rates (2 Hz in our setup). A similar sensor is described by Scherer et al. (2012) and Cover et al. (2013). Their MAV is used to autonomously explore rivers using visual localization and laser-based 3D obstacle perception. In contrast to their work, we use the 3D laser scanner for both omnidirectional obstacle perception and mapping the environment in 3D.

To robustly detect different types of obstacles in all directions, we combine multiple modalities. Our sensor setup consists of a lightweight 3D laser scanner, two stereo camera pairs, and ultrasonic sensors. It contains only lightweight sensors and is particularly well suited for MAVs (Holz et al., 2013).

2.2 Navigation Planning

A good survey on approaches to motion planning for MAVs is given by Goerzen et al. (2010). Due to the limited computational power onboard the MAV, low computational costs are crucial for the applicability of these algorithms. Consequently, layered planning algorithms are often used. Our approach uses four layers operating at different time scales and different layers of abstraction.

A two-level approach to collision-free navigation, using artificial potential fields on the lower layer is proposed by Ok et al. (2013). Similar to our work, completeness of the path planner is guaranteed by an allocentric layer on top of local collision avoidance. Andert et al. (2010) use a three-level hierarchical behavior control algorithm to fly a helicopter through a gate. Whalley et al. (2014) employ five navigation layers to fly 230 km with a helicopter. Obstacles are detected and avoided with an onboard laser scanner. While in their work sensing and consequently planning is limited to a narrow FoV in flight direction, we employ full 3D planning, including flying sideways and backwards.

Some reactive collision avoidance methods for MAVs are based on optical flow (Green and Oh, 2008) or a combination of flow and stereo vision (Hrabar et al., 2005). However, solely optical flow-based solutions cannot cope well with frontal obstacles and these methods are not well suited for omnidirectional obstacle avoidance as needed for our scenario. Johnson and Mooney (2014) use reactive obstacle avoidance on a small helicopter for velocities up to 12 m/s. Heng et al. (2014) use a multiresolution grid map to represent the surroundings of a quadrotor. A feasible plan is generated with a vector field histogram (Ulrich and Borenstein, 1998). We employ reactive obstacle avoidance as a fast safetety layer, and frequent replanning in a multiresolution grid to generate collision free local paths.

Schmid et al. (2014) autonomously navigate to user-specified waypoints in a mine. The map used for planning is created by an onboard stereo camera system. By using rapidly exploring random belief trees (RRBT), Achtelik et al. (2014) plan paths that do not only avoid obstacles, but also minimize the variability of the state estimation. Recent search-based methods for obstacle-free navigation include work of MacAllister et al. (2013). They use A*-search to find a feasible path in a four-dimensional grid map—accounting for the asymmetric shape of their MAV. With our sensor setup, we cover most of the space around the approximately circular robot, hence we control view directions independent from path planning according to mission-defined points of interest. Cover et al. (2013) use a search-based planning method as well. It assumes complete knowledge of the scene geometry—an assumption that we do not make here.

2.3 Simultaneous Localization and Mapping

The Simultaneous Localization and Mapping (SLAM) problem has attracted attention especially in the field of autonomous navigation with mobile ground robots (Nuechter et al., 2005; Magnusson et al., 2007). LRFs provide accurate distance measurements in a large FoV, with only minor dependencies on lighting conditions. A common research topic in SLAM with 3D laser scanners is how to maintain high run-time performance and low memory consumption simultaneously.

Hornung et al. (2013b) implement a multiresolution map based on octrees (OctoMap). Ryde and Hu (2010) use voxel lists for efficient neighbor queries. Both of these approaches consider mapping in 3D with a voxel being the smallest map element. Similar to our approach, the 3D-NDT (Magnusson et al., 2007) represents point clouds as Gaussian distributions in voxels at multiple resolutions. Our local multiresolution surfel grids adapt the maximum resolution with distance to the sensor to incorporate measurement characteristics.

Moreover, our registration method matches 3D scans on all resolutions concurrently—utilizing the finest common resolution available between both maps, which also makes registration efficient. In previous own work (Stückler and Behnke, 2014; Schadler et al., 2013), we used this concept within an octree voxel representation. We model up to six view directions, such that multiple 3D scans from different view points can be integrated in a single multiresolution surfel map. Compared to the dense RGB-D images used in our previous work, the 3D scans obtained from our LRF are much sparser. We overcome this sparsity through probabilistic assignments of surfels during the registration process.

While many methods assume the robot to stand still during 3D scan acquisition, some approaches also integrate scan lines of a continuously rotating laser scanner into 3D maps while the robot is moving (Bosse and Zlot, 2009; Elseberg et al., 2012; Stoyanov and Lilienthal, 2009; Maddern et al., 2012; Anderson and Barfoot, 2013).

Above mentioned Scherer et al. (2012) and Cover et al. (2013) use a 3D laser scanner for obstacle perception for autonomous river exploration. They approach MAV localization with a vision sensor. In contrast, we combine visual odometry with 3D scan registration in a 3D multiresolution map to localize the MAV.

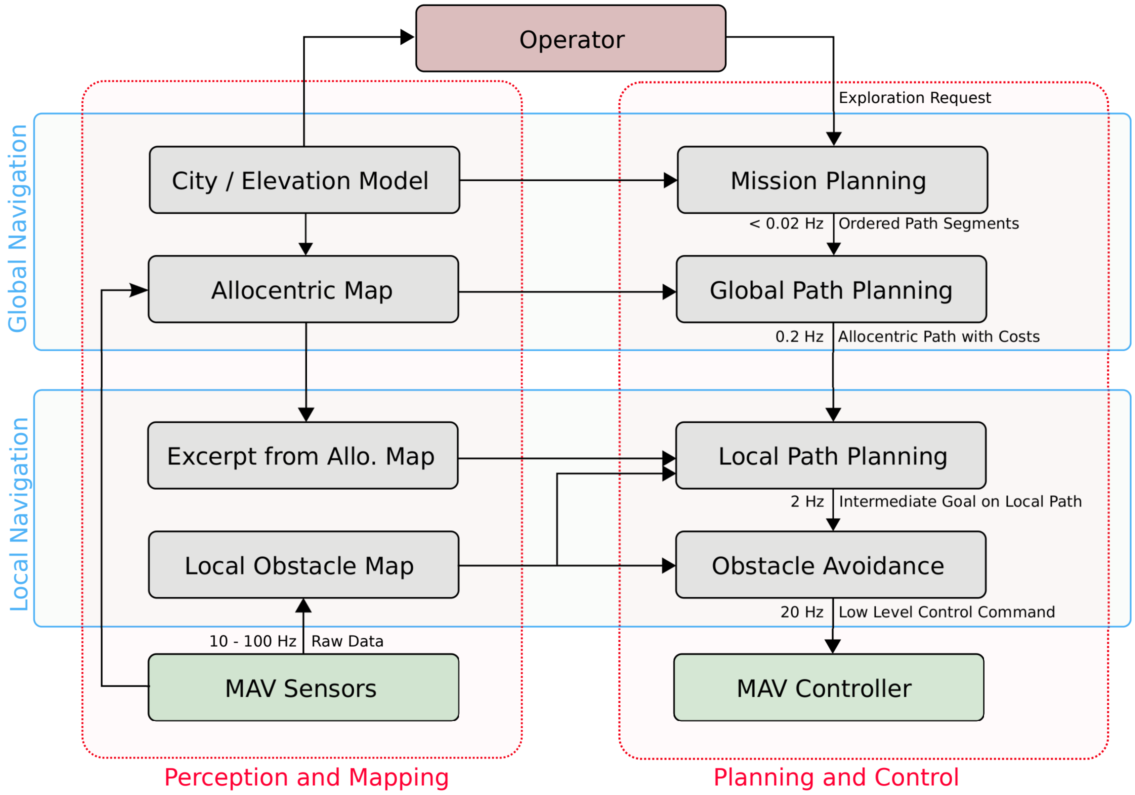


Figure 2: Multi-layered navigation approach: slow planners on the top yield coarse trajectories which are refined on faster lower layers. The layers build and use different environment representations: allocentric maps from city models and local egocentric obstacle maps.

Takahashi et al. (2008) also build environment maps with a 3D laser scanner. They localize the robot using GPS and IMU sensors. Thrun et al. (2003) propose a 3D mapping system with a rigidly mounted 2D laser scanner on a helicopter. The laser scanner measures in a vertical plane perpendicular to the flight direction. In order to localize the helicopter, measurements from GPS and IMU are fused and consecutive 2D scans are registered, assuming scan consistency in flight direction. In our approach, we do not make such an assumption on scan consistency.

3 System Setup and Overview

3.1 Approach Overview and Data Flow

To allow for prompt reactions on obstacle perceptions on the one end, and consistent mapping and complex planning on the other end, our system architecture is layer-based (see Figure 2). Accordingly, we have slower global layers on the top (allocentric mapping, deliberative planning) and faster local layers on the bottom (egocentric obstacle maps, reactive obstacle avoidance). From top to bottom, the abstraction level of planning and mapping is reduced and the processing frequency approaches the sensor measurement frequency.

3.2 Platform and Sensor Setup

Our MAV platform is an octorotor multicopter with a co-axial arrangement of rotors (see Figure 1). This yields a compact flying platform that is able to carry a plurality of sensors and an onboard computer

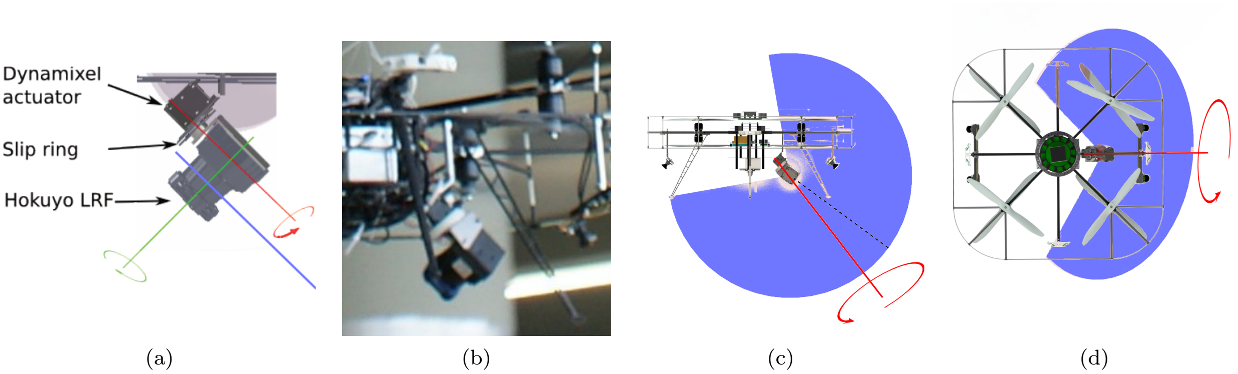


Figure 3: 3D scanner setup: (a) CAD drawing of our continuously rotating laser scanner. The Hokuyo 2D LRF is mounted on a bearing and rotated around the red axis. Its mirror is rotated around the green axis, resulting in a 2D measurement plane (blue). (b) Photo of the sensor. (c + d) CAD drawings illustrating the FoV of individual scans of the laser scanner (blue) from side and top view. The black dashed line illustrates the centre of the measurement plane.

with sufficient computing power (Intel Core i7-3820QM 2.7 GHz). For sensor data processing and navigation planning, we employ the Robot Operating System (ROS), Quigley et al. (2009), as middleware. For low-level velocity and attitude control, the MAV is equipped with a PIXHAWK Autopilot flight control unit (Meier et al., 2012). To allow for safe omnidirectional operation in challenging environments, our MAV is equipped with a multimodal sensor setup:

- Our main sensor for obstacle perception is a continuously rotating 3D laser scanner (Figure 3). The scanning plane of the Hokuyo UTM-30LX-EW 2D LRF is parallel to the axis of rotation, but the heading direction of the scanner is twisted slightly away from the direction of the rotational axis—in order to enlarge its FoV (Figure 3a). The measurement density of the 3D laser scanner varies and has its maximum in a forward-facing cone. The sensor is pitched downward by 45° in forward direction, so that only a small conical spot above the back of the MAV is occluded by its core.
- Two monochrome stereo camera pairs (pointing slightly downward in forward and backward direction) are used for visual odometry and obstacle perception. Equipped with fish-eye lenses, they cover a large area around the MAV.
- Eight ultrasonic sensors around the MAV complete the perception setup. Despite their limited accuracy and range, they aid the perception of transparent obstacles in the vicinity of the MAV, such as windows.
- Additionally, we use GPS and an optical flow camera (Honegger et al., 2013) for localization and state estimation. The flow camera is pointing vertically to the ground and can—given suitable lighting conditions—measure velocities relative to the ground-plane with more than 100 Hz update rate.

We detail our sensor setup in (Holz et al., 2013; Droeschel et al., 2013). The fusion of these sensors allows the reliable detection and avoidance of obstacles.

4 Perception and Mapping

In order to accumulate laser range measurements, we construct an MAV-centric multiresolution grid map (Section 4.2). The map is used by our path planning and obstacle avoidance algorithms described in subsequent sections.

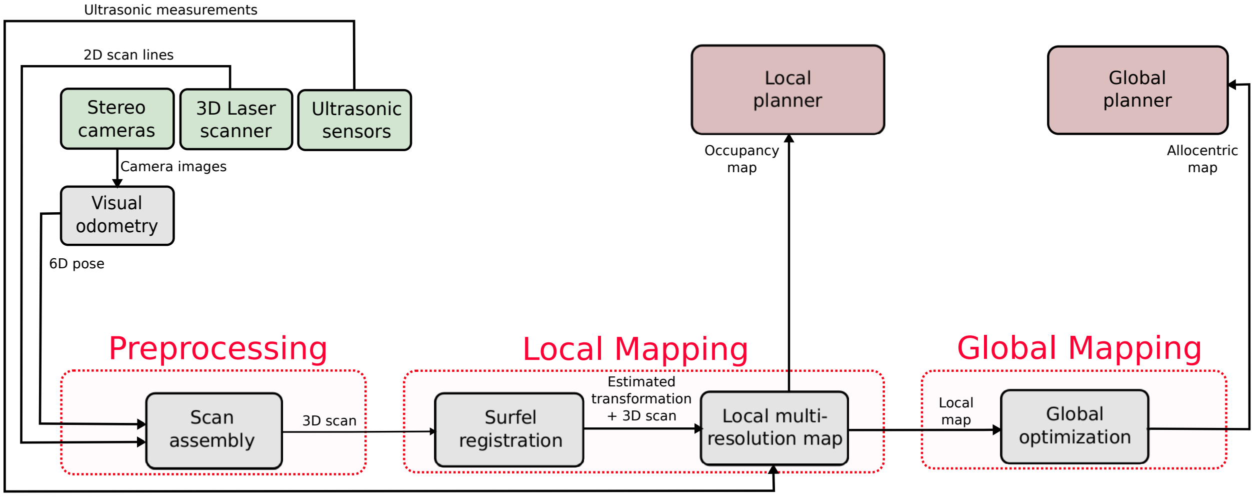


Figure 4: Overview of our mapping system. The LRF measurements are processed in preprocessing steps described in Section 4.1. The resulting 3D point cloud is used to estimate the transformation between the current scan and the map as described in Section 4.4. Registered scans are stored in a local multiresolution map. These maps are aligned to build an allocentric map as described in Section 4.5.

We register consecutive 3D laser scans with our local multiresolution surfel grid map to estimate the motion of the MAV. 3D scans are acquired in each half rotation of the laser. Since the scans are taken in-flight in a sensor sweep, the motion of the MAV needs to be compensated for when assembling the scan measurements into 3D scans (Section 4.1). Furthermore, we aggregate 3D scans in a local map of the environment. We first register newly acquired 3D scans with the so far accumulated map and then update the map with the registered 3D scan. We concurrently build an allocentric map, where local multiresolution maps acquired from different view poses are aligned (Section 4.5). The architecture of our perception and mapping system is outlined in Figure 4.

4.1 3D Scan Assembly

Assembling raw laser scans to 3D scans must not only account for the rotation of the scanner w.r.t. to the MAV, but also for MAV motion during acquisition. We undistort 3D scans in two steps.

First, measurements of individual scan lines are undistorted with regards to the rotation of the 2D LRF around the servo rotation axis (red axis in Figure 3). Here, the rotation between the acquisition of two scan lines is distributed over the measurements by using spherical linear interpolation.

Second, we compensate for the motion of the MAV during acquisition of a full 3D scan. To this end, we incorporate a visual odometry estimate from the two stereo cameras. Here, a keyframe-based bundle adjustment is performed (Schneider et al., 2013) on the synchronized images with 18 Hz update rate. Since the update rate of the 2D LRF is 40 Hz, we linearly interpolate between the estimates of the visual odometry.

The 6D motion estimate is used to assemble the individual 2D scan lines of each half rotation to a 3D scan. Figure 5 illustrates the effect of scan undistortion.

Since the dynamics of MAVs are restricted, we found linear undistorsion sufficient for preprocessing. Especially the lack of jerk and the bounded acceleration that are occurring during normal flight conditions do not require more sophisticated algorithms. The integration of IMU measurements could possibly further improve the result, though.

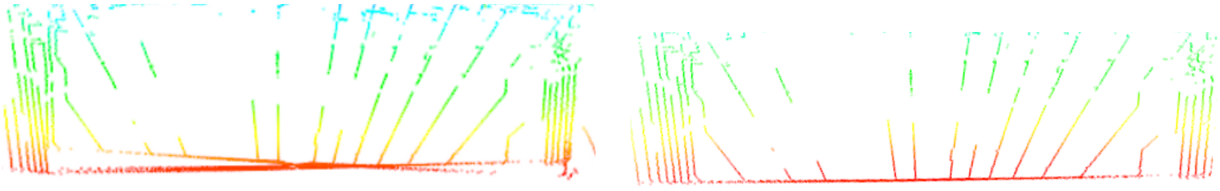


Figure 5: Side view of an indoor 3D scan with flat ground. Left: assembled 3D scan without considering sensor movement during scan acquisition. Right: We incorporate visual odometry to correct for the sensor movement.

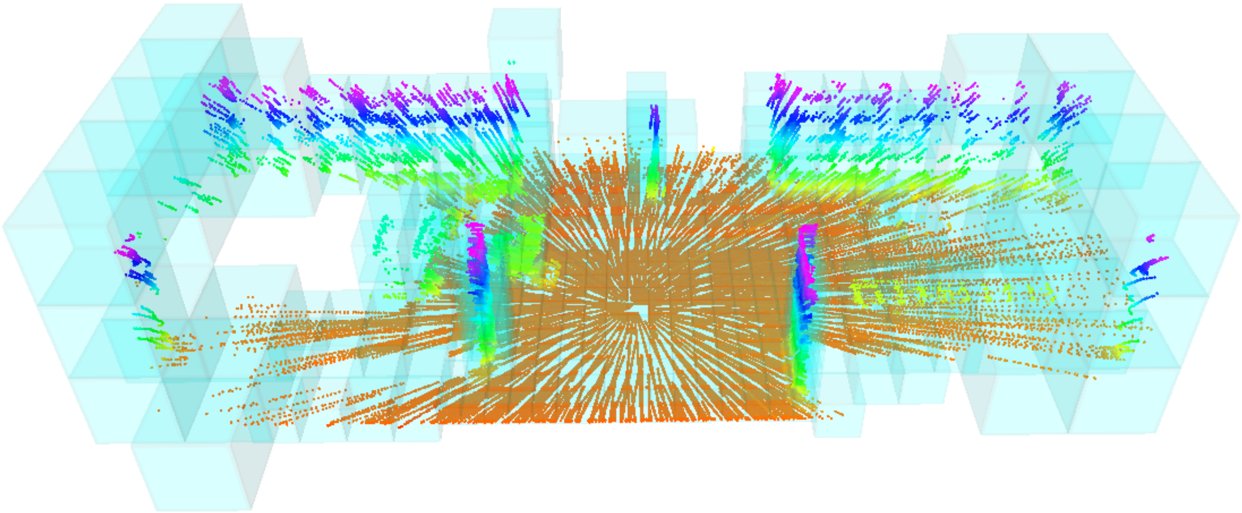
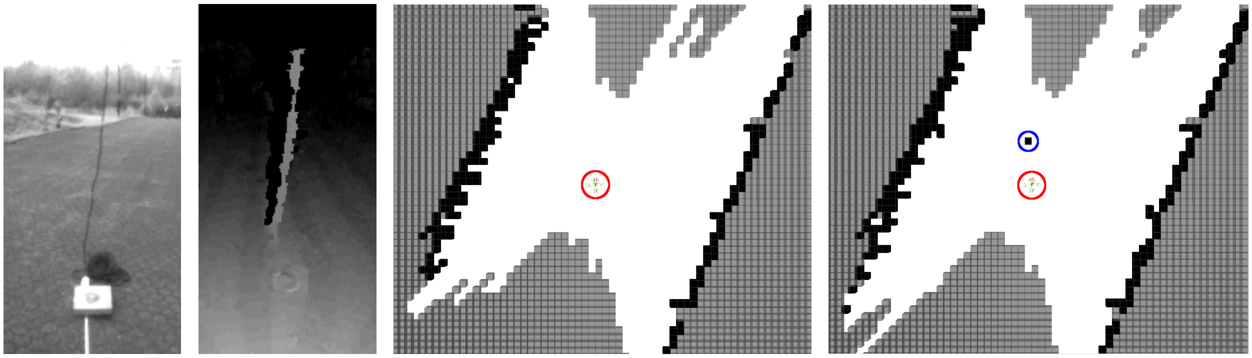


Figure 6: Grid-based local multiresolution map with a higher resolution in proximity to the sensor and a lower resolution with increasing distance. Color encodes height.

4.2 Local Multiresolution Map

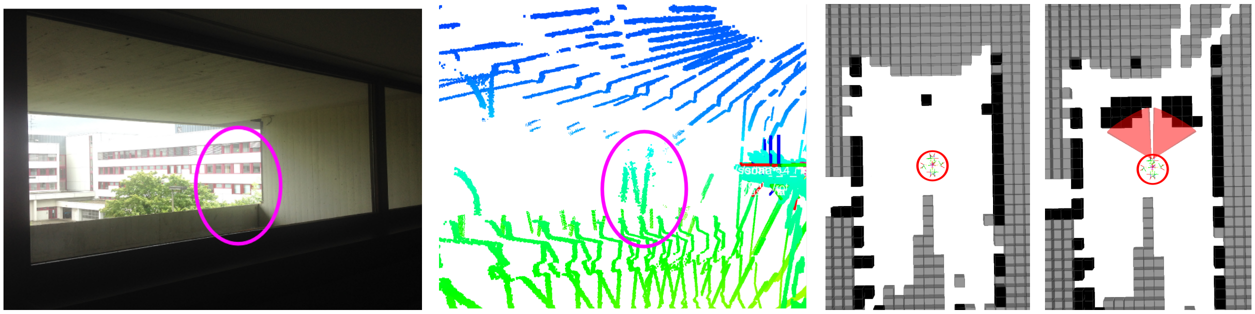
We use a hybrid local multiresolution map that represents both occupancy information and the individual distance measurements. The most recent measurements are stored in ring buffers within grid cells that increase in size with distance from the robot center. Thus, we gain a high resolution in the close proximity to the sensor and a lower resolution far away from our robot, which correlates with the sensor characteristics in relative distance accuracy and measurement density. Compared to uniform grid-based maps, multiresolution leads to the use of fewer grid cells, without losing relevant information and consequently results in lower computational costs. Figure 6 shows an example of our multiresolution grid-based map.

We aim for efficient map management for translation and rotation. To this end, individual grid cells are stored in a ring buffer to allow shifting of elements in constant time. We interlace multiple ring buffers to obtain a map with three dimensions. The length of the ring buffers depends on the resolution and the size of the map. In case of a translation of the MAV, the ring buffers are shifted whenever necessary to maintain the egocentric property of the map. For sub-cell-length translations, the translational parts are accumulated and shifted if they exceed the length of a cell.



(a) (b) (c) (d)

Figure 7: Obstacle perception by fusing laser range measurements of our 3D laser scanner with dense stereo measurements. (a) A loose-hanging cable at 3 m distance, which is not perceived with the 3D laser scanner of the MAV; (b) Dense stereo allows for detecting the cable; (c) The resulting occupancy grid map with measurements from the laser scanner solely; (d) The resulting occupancy grid map with laser and dense stereo. The fused map shows the obstacle circled in blue. The MAV position is circled red. We chose a single layer of a uniform grid for visualization.



(a) (b) (c) (d)

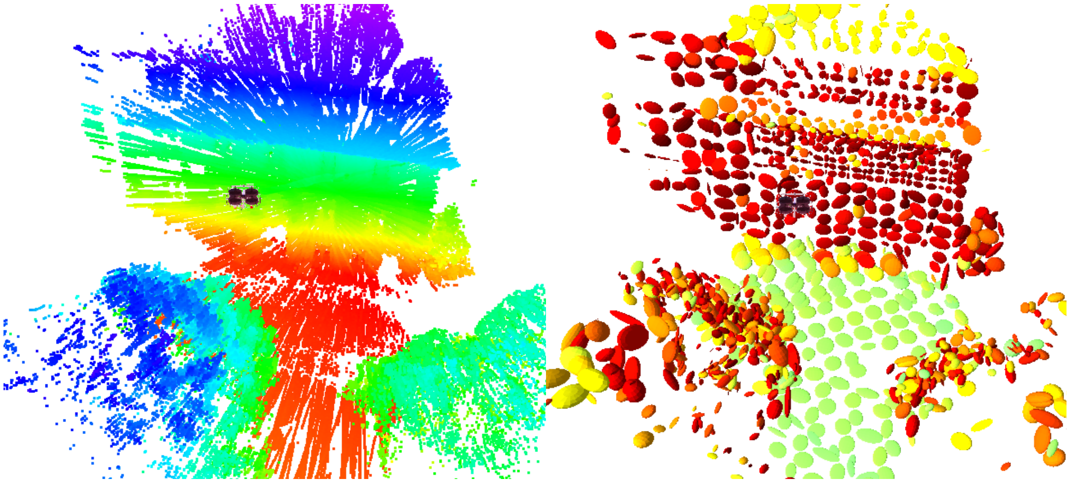
Figure 8: Obstacle perception by fusing laser range measurements of our 3D scanner with ultrasonic measurements: (a) A window in 2 m distance; (b) The window is only partially perceived by the 3D laser scanner; (c) The resulting occupancy grid map with measurements from the laser scanner solely; (d) The resulting occupancy grid map with laser and ultrasonic measurements (red cones). The position of the MAV is circled red. We chose a single layer of a uniform grid for visualization.

4.3 Occupancy Mapping

The individual sensors of our MAV have different strengths and weaknesses. In order to perceive as many obstacles as possible, it is necessary to fuse the measurements adequately into a single map. We collect these measurements in an occupancy grid maintaining occupancy probabilities. We fuse measurements from the 3D laser scanner, from the wide-angle stereo cameras, and from the ultrasonic sensors. Figure 7 shows an example of an outdoor scenario where fusing laser range measurements with dense stereo (Geiger et al., 2010) allows for perception of challenging obstacles. Besides very thin obstacles such as a cables, transparent objects are demanding for reliable obstacle perception. Figure 8 shows how fusing measurements from our 3D laser scanner with ultrasonic measurements allows for detecting transparent obstacles, like windows.

4.4 Registration Approach

We register the points $\mathcal{P} = \{p_1, \ldots, p_P\}$ in a 3D scan with the points $\mathcal{Q} = \{q_1, \ldots, q_Q\}$ in the local grid map of the environment (Droeschel et al., 2014). Instead of considering each point individually, we map the 3D



(a) Aggregated point cloud (colored by height).

(b) Multiresolution surfel map (colored by surfel orientation).

Figure 9: Example of a multiresolution surfel map.

scan into a local multiresolution grid and match surfels. In each voxel in the grid, we maintain one surfel that summarizes the $P_{x,i}$ points that lie within the voxel. A surfel is defined by the sample mean $\mu_{x,i}$ and the sample covariance $\Sigma_{x,i}$ of these points.

We denote the set of surfels in the scene (the 3D scan) by $X = \{x_1, \ldots, x_N\}$ and write $Y = \{y_1, \ldots, y_M\}$ for the set of model surfels in the environment map. We assume that scene and model can be aligned by a rigid 6 degree-of-freedom (DoF) transformation $T(\theta)$ from scene to model. The observation likelihood of the scene surfels in the environment map, given the pose estimate θ,

$$p(\mathcal{P} \mid \theta, \mathcal{Q}) \approx \prod_{i=1}^{N} p(x_i \mid \theta, Y)^{P_{x,i}}. \tag{1}$$

now considers the number of points in each surfel in order to approximate the point-to-point observation likelihood. By this, several orders of magnitudes less map elements are used for registration than if the points would have been used individually. Similarly, the registration of two local maps is treated as the registration of their point sets.

4.4.1 Gaussian Mixture Observation Model

We explain each transformed scene surfel as an observation from a mixture model, similar as in the coherent point drift (CPD) method (Myronenko and Song, 2010). A surfel x_i is observed under the mixture distribution defined by the model surfels and an additional uniform component that explains outliers, i.e.,

$$p(x_i \mid \theta, Y) = \sum_{j=1}^{M+1} p(c_{i,j})\ p(x_i \mid c_{i,j}, \theta, Y). \tag{2}$$

The binary variable $c_{i,j}$ indicates the association of x_i to one of the $M + 1$ mixture components (defined by surfel y_j or the uniform component if $j = M + 1$). The mixture components for the M model surfels correspond to the matching likelihood between the scene surfel x_i and each of the model surfels through

$$p(x_i \mid c_{i,j}, \theta, Y) := \mathcal{N}\left[T(\theta)\mu_{x,i}; \mu_{y,j}, \Sigma_{y,j} + R(\theta)\Sigma_{x,i}R(\theta)^T + \sigma_j^2 I\right], \tag{3}$$

where $\sigma_j = \frac{1}{2}\rho_{y,j}^{-1}$ is a standard deviation that we adapt to the resolution $\rho_{y,j}$ of the model surfel. We set the likelihood of the uniform mixture component to a constant. This way, we do not make a hard association decision for each surfel, but a scene surfel is associated to many model surfels.

4.4.2 Registration through Expectation-Maximization

The alignment pose θ is estimated through maximization of the logarithm of the joint data-likelihood

$$\ln p(\mathcal{P} \mid \theta, \mathcal{Q}) \approx \sum_{i=1}^{N} P_{x,i} \ln \sum_{j=1}^{M+1} p(c_{i,j})\; p(x_i \mid c_{i,j}, \theta, Y). \tag{4}$$

We optimize this objective function through expectation-maximization (EM) (Bishop, 2006). In the M-step, the latest estimate $\overline{q}$ for the distribution over component associations is held fixed to optimize for the pose θ

$$\widehat{\theta} = \underset{\theta}{\operatorname{argmax}} \;\; const. + \sum_{i=1}^{N} P_{x,i} \sum_{j=1}^{M+1} \overline{q}(c_{i,j}) \ln p(x_i \mid c_{i,j}, \theta, Y). \tag{5}$$

Writing out the mixture component distributions and using the fact the the uniform component is constant in θ, we find that this optimization problem is a non-linear least squares problem,

$$\widehat{\theta} = \underset{\theta}{\operatorname{argmax}} \;\; const. + \sum_{i=1}^{N} P_{x,i} \sum_{j=1}^{M} \overline{q}(c_{i,j})\, e_{i,j}^{T}\, W_{i,j}\, e_{i,j}, \tag{6}$$

where we defined $e_{i,j}(\theta) := \mu_{y,j} - T(\theta)\mu_{x,i}$ and $W_{i,j}(\theta) := \left(\Sigma_{y,j} + R(\theta)\Sigma_{x,i}R(\theta)^T\right)^{-1}$.

This optimization is efficiently performed using the Levenberg-Marquardt (LM) method as in (Stückler and Behnke, 2014). We stack the residuals $e_{i,j}(\theta)$ in a single vector $e(\theta)$. Similarly, we combine the information matrices $W_{i,j}$ in a block-diagonal weighting matrix W, but for which we treat the pose as being constant for the non-linear least-squares optimization. Additionally, according to Eq. (5), each association needs to be weighted by a factor $w_{i,j} := P_{x,i}\, \overline{q}(c_{i,j})$. These weights are additional factors to the information matrices in the individual blocks $(w_{i,j}W_{i,j})$ of W.

The steps taken by LM optimization are

$$\Delta\theta := (J^T W J + \lambda I)^{-1} J^T W e(\theta), \tag{7}$$

where $J := \nabla_\theta e(\theta)$ is the Jacobian of the stacked residuals, and λ is adjusted by LM to trade between Gauss-Newton and gradient descent steps. Note that due to the block-diagonal structure of W, this update decomposes into sums over individual terms per association. The covariance of the LM estimate is readily obtained by $\Sigma(\theta) := (J^T W J)^{-1}$.

The E-step obtains a new optimum $\widehat{q}$ for the distribution q by the conditional likelihood of the cluster associations given the latest pose estimate $\overline{\theta}$

$$\widehat{q}(c_{i,j}) = \frac{p(c_{i,j})\; p(x_i \mid c_{i,j}, \overline{\theta}, Y)}{\sum_{j'=1}^{M+1} p(c_{i,j'})\; p(x_i \mid c_{i,j'}, \overline{\theta}, Y)}. \tag{8}$$

In order to evaluate these soft assignments, we perform a local search in the local multiresolution surfel grid of the model. We first look up the grid cell with a surfel available on the finest resolution in the model map at the transformed mean position of the scene surfel. We consider the surfels in this cell and its direct neighbors for soft association.

4.5 Simultaneous Localization and Mapping

Our map representation and registration method is able to track the pose of the MAV in a local region, since we decrease the resolution in the map with distance to the MAV. In order to localize the robot in a fixed frame towards its environment and to concurrently build an allocentric map, we align local multiresolution maps acquired from different view poses.

We register the current local multiresolution map towards a reference key view to keep track of the MAV motion. A new key view is generated for the current map, if the robot moved sufficiently far. The new key view is set as the reference for further tracking. The registration result x_i^j between a new key view v_i and its reference v_j is a spatial constraint that we maintain as values of edges $e_{ij} \in \mathcal{E}$ in a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ of key views.

To overcome pure time-sequential pose tracking by registration, we add spatial constraints between close-by key views that are not in temporal sequence. On-line SLAM is enabled by establishing up to one spatial constraint per 3D scan update.

4.5.1 Constraint Detection

On each scan update, we check for one new constraint between the current reference v_{ref} and other key views v_{cmp}. We determine a probability

$$p_{\text{chk}}(v_{\text{cmp}}) = \mathcal{N}\left(d(x_{\text{ref}}, x_{\text{cmp}}); 0, \sigma_d^2\right) \tag{9}$$

that depends on the translational distance $d(x_{\text{ref}}, x_{\text{cmp}})$ between the key view poses x_{ref} and x_{cmp}. We sample a key view v according to $p_{\text{chk}}(v)$ and determine a spatial constraint between the key views using our registration method.

4.5.2 Pose Graph Optimization

From the graph of spatial constraints, we infer the probability of the trajectory estimate given all relative pose observations

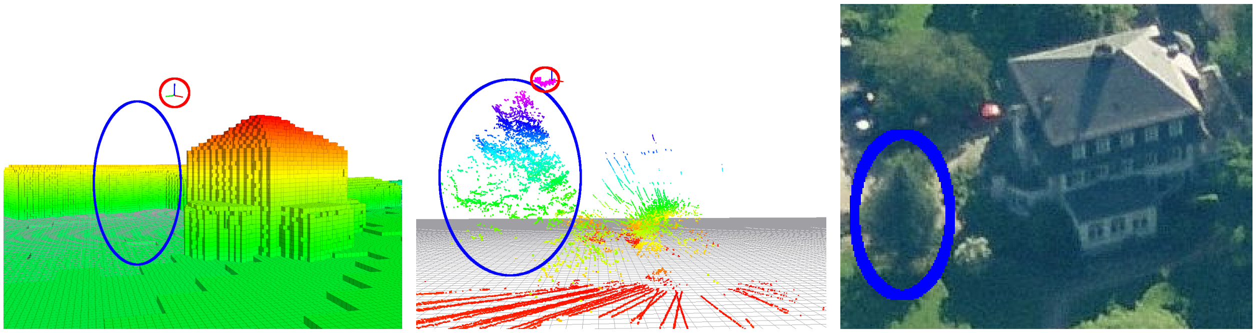
$$p(\mathcal{V} \mid \mathcal{E}) \propto \prod_{e_{ij} \in \mathcal{E}} p(x_i^j \mid x_i, x_j). \tag{10}$$

Each spatial constraint is a normally distributed estimate with mean and covariance determined by our probabilistic registration method. This pose graph optimization is efficiently solved using the g^{2}o framework (Kuemmerle et al., 2011).

5 Planning

For successful fulfillment of an exploration and mapping mission, safe navigation is key. Multiple planning and navigation tasks have to be executed: from mission planning to low-level motion control. These tasks require different abstractions of the environment, as illustrated in Figure 2.

To plan a mapping mission, we employ a coarse (semantic) model of the environment; to plan collision-free paths, we need a finer and up-to-date consistent geometric model; and to avoid obstacles, we need a non-aggregated local representation of the close vicinity of the MAV. The planned actions also have different granularity, which is represented by the planning frequency: from once per mission to multiple times per second. The higher-layer planners set goals for the lower-level planners which produce more concrete action sequences based on more local and up-to date environment representations.



(a) OctoMap derived from city model. (b) Laser scan of the environment. (c) Aerial photo of the environment.

Figure 10: (a) For the more abstract high-level planning layers, we employ coarse models of the environment, i.e., a 3D city model and a digital elevation model as provided by land surveying authorities. (b) The planned paths are refined during a mission by means of the local planning and obstacle avoidance layers operating with onboard sensor measurements, e.g., 3D laser scans. Including a building (right) and vegetation (blue circle). Measurements on the MAV itself are circled red. The color in both figures depicts the height. The (approximate) position of the scanned tree is circled blue in all figures. (c) An aerial photo of the scanned environment (Image credit: www.bing.com/maps).

5.1 Mission Planning

The topmost layers are a mission planner and a global path planner. Both use a static representation of the environment that is derived from a 3D city model and a digital elevation model, depicted in Figure 10(a). These models are available from land surveying authorities and aid as initial information for planning a mission to inspect buildings or search for persons. The combined model is stored efficiently in an OctoMap (Hornung et al., 2013a).

Input to the mission planner is a set of view poses defined by the user. The mission planner employs a global path planner on a coarse uniform grid map to determine the approximate costs between every pair of mission goals. In order to speed up the process, we reuse already calculated information, e.g., the obstacle costs per grid cell stay the same for every combination of view poses. Furthermore, the costs of reaching grid cells from one start pose stay constant in this offline processing step. After calculating all pair-wise edge weights, the cost-optimal sequence of view poses is determined by means of Concorde (Applegate et al., 2006), a fast solver for the traveling salesman problem (TSP). Please note, that the instances of the TSP for one mission are sufficiently small, so that exact solutions are tractable.

The result of mission planning is a flight plan composed of a list of waypoints, the MAV should pass approximately or reach exactly, depending on the mission objectives. Figure 11 shows an example solution for a mission to build the map in Figure 24.

5.2 Global Path Planning

The next layer in the planning hierarchy is a global path planner. This layer plans globally consistent plans, based on I) the static environment model, discretized to grid cells with 1 m edge length, II) the current pose estimate of the MAV, and III) a robot-centric local grid map representing the vicinity of the MAV. Planning frequency is 0.2 Hz and we use the A* algorithm to find cost-optimal paths.

We assume, that in our application domain most obstacles that are not known in advance can be surrounded locally, without the need for global replanning. Hence, it is sufficient to replan globally on a more long-term time scale to keep the local deviations of the planner synchronized to the global plan and to avoid the MAV to get stuck in a local minimum that the local planner cannot solve due to its restricted view of the

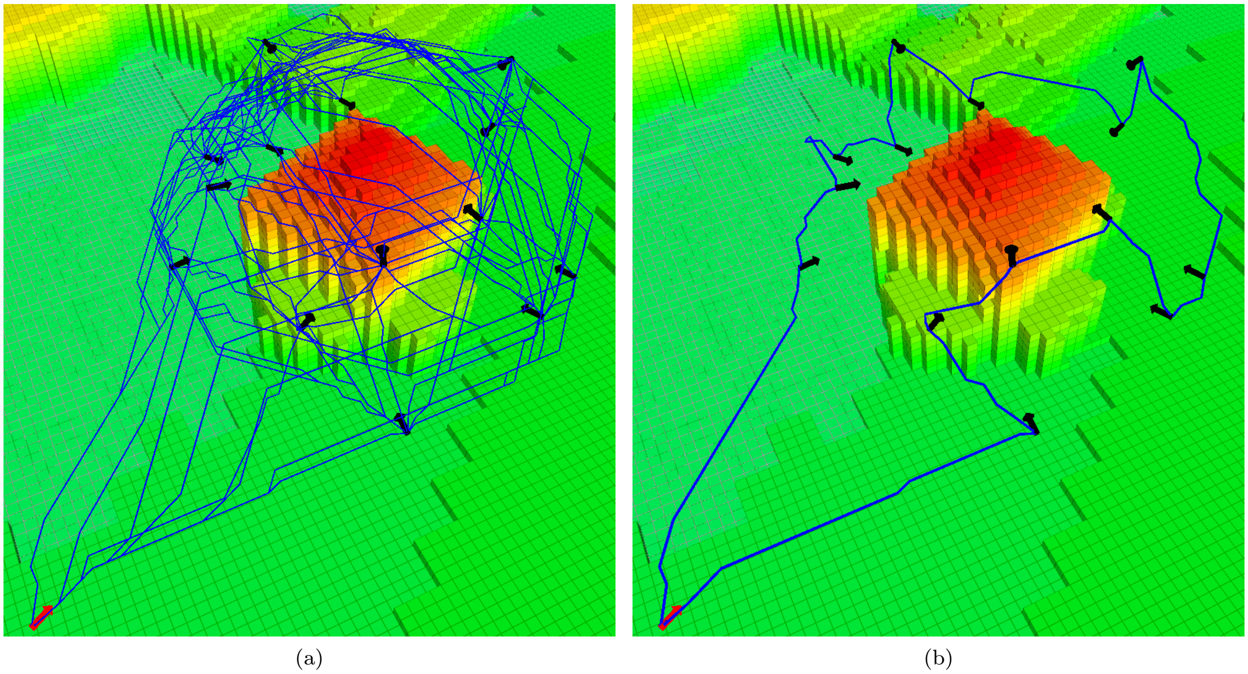


Figure 11: On the top layer, a mission planner evaluates the best execution order of mission poses (black arrows). (a) All cost-optimal trajectories between each pair from the set of mission waypoints, including the current robot pose (red arrow). (b) The optimal flight plan to reach all waypoints and return to the start pose. The cost function allows for positions close to the building but penalizes these more than paths farther away.

environment.

As via-points that are not mission critical can be blocked by locally perceived obstacles, it is not sufficient to send the next waypoint of the global path to the local planning layers. Instead, the input to the local planner is the complete global plan. The global path is cost-optimal with respect to the allocentric map. Hence, the path costs of the global path are a lower bound to path costs for refined plans, based on newly acquired sensor information—mostly dynamic and static previously unknown obstacles—and a local path deviating from the global plan cannot be shorter in terms of path costs. Locally shorter plans on lower layers with a local view on the map may yield globally suboptimal paths. Because of that, we add the estimated path costs between waypoints of the global plan to its edges to facilitate efficient exploration of the search space on the next lower layer. The optimal costs between every two waypoints of the global path can be determined by the local planner and there exists no shorter way between these. Also, mission goals are marked as the local planner has to reach these exactly. If this is not possible, the mission planning has to resolve this failure condition.

5.3 Local Multiresolution Planning

On the local path planning layer, we employ a 3D local multiresolution path planner. This planner uses as input the solution by the allocentric planning layer, a local excerpt of the global map, and local distance measurements which have been aggregated in a 3D local multiresolution map (Sec. 4.2). It refines the global path according to the actual situation. The more detailed trajectory is fed to the potential field-based reactive obstacle avoidance layer on the next level (cf. Sec. 5.4).

To resemble the relative accuracy of onboard sensors—i.e., they measure the vicinity of the robot more accurate and with higher density than distant space—we plan with a higher resolution close to the robot

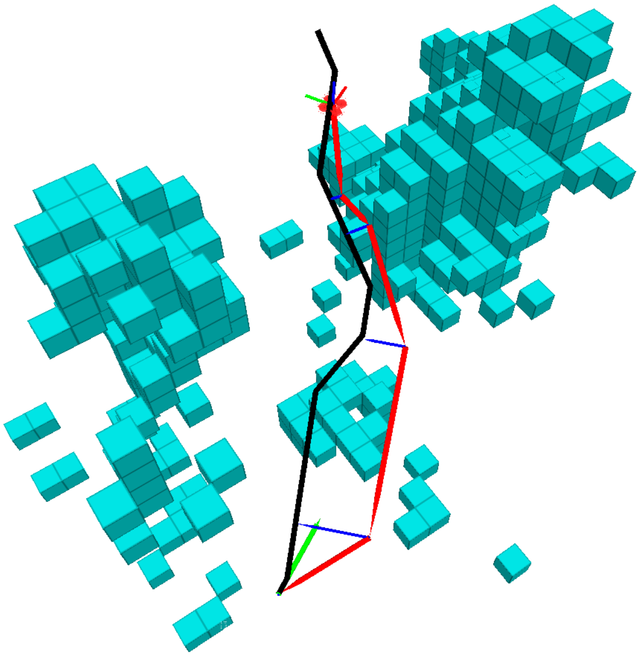


Figure 12: The local plan (red) is coupled with the allocentric plan (black) by a cost term that penalizes deviations from the allocentric plan. The blue lines depict the deviation vectors at example points, the green arrow indicates the goal for local planning.

and with coarser resolutions with increasing distance from the robot.

This local multiresolution approach to path planning is very efficient. Since parts of the plan, that are farther away from the MAV are more likely to change, e.g., due to newly acquired sensor data, it is efficient to spend more planning effort in the close vicinity of the robot. Compared to uniform resolution planning our approach reduces the planning time drastically and makes frequent replanning feasible. Our planner operates on grid-based robot-centric obstacle maps with higher resolution in the center and decreasing resolution in the distance, similar to the representation in Sec. 4.2. We embed an undirected graph into this grid (Figure 13a) and perform A* graph search (Hart et al., 1968) from the center of the MAV-centered grid to the goal. The edge costs are determined using the obstacle costs of the cells it is connecting and edge length.

An obstacle is modeled as a core with maximum costs, determined by obstacle radius r_F and enlarged by the approximate robot radius r_R. The obstacle costs are multiplied by the fraction of the edge length within the respective cells. Figure 13b shows our obstacle model: a core of the perceived obstacle enlarged by the approximate robot radius r_C and a distance-dependent part r_S that models the uncertainty of farther away perceptions and motions with high costs. Added is a part with linearly decreasing costs with increasing distance to the obstacle r_A that the MAV shall avoid if possible. The integral of the obstacle stays constant by reducing its maximum costs h_{max} with increasing radius. For a distance d between a grid cell center and the obstacle center, the obstacle costs h_c are given by

$$h_c(d) = \begin{cases} h_{max} & \text{if } d \leq (r_F + r_D) \\ h_{max} \frac{1-d-(r_F+r_D)}{2*(r_F+r_D)} & \text{if } (r_F + r_D) < d < 3*(r_F + r_D) \\ 0 & \text{otherwise} \end{cases} .$$

The local planner is coupled to the solution of the allocentric path planner by a cost term h_a penalizing deviations. For every grid cell visited in the planning process, we calculate the point-line distance between the grid cell's center and the closest segment of the allocentric path (see Figure 12). The distance contributes

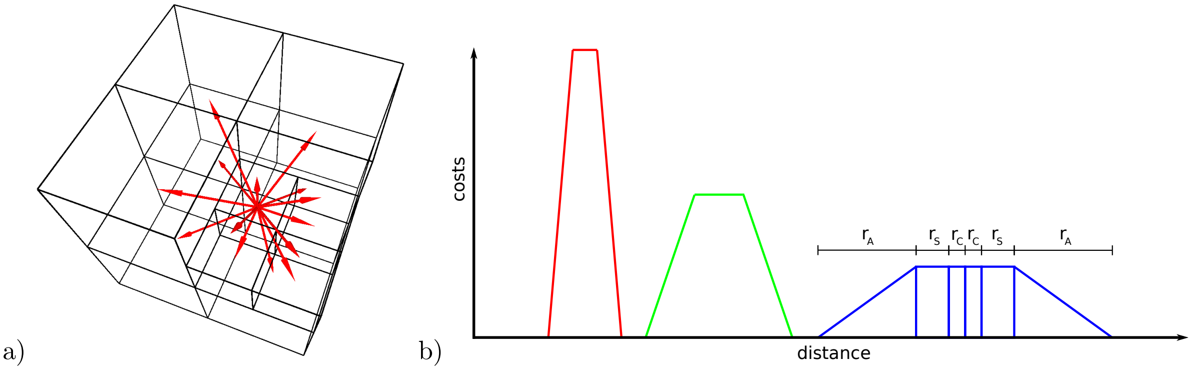


Figure 13: Local multiresolution path planning. (a) Connectivity in the robot-centered multiresolution planning grid. Red lines depict edges from a cell at the corner of an inner (high resolution) grid to its neighbors. (b) We model obstacles in the local multiresolution grid as a fixed core r_C, a safety area with maximum costs r_S, and an avoidance zone with linear decreasing costs r_A. With increasing distance to the grid's origin the radii of these areas increase and their maximum cost decreases to account for the uncertainty in measurements.

to the overall cost for traversing the cell.

A situation where purely reactive obstacle avoidance fails but local planning is sufficient to find a solution is shown in Figure 14.

5.4 Local Obstacle Avoidance

On the next lower layer, we employ a fast reactive collision avoidance module based on artificial potential fields (Ge and Cui, 2002) as a safety measure reacting directly on the available sensor inputs.

The robot-centered local multiresolution occupancy grid, the current motion state x_t, and a target velocity v_t, serve as input to our algorithm. The obstacle map induces repulsive forces on particles on the MAV with magnitude

$$F_r^p = \text{costs}\left(argmin_o\left(\|o-p\|\right)\right) \tag{11}$$

for an obstacle at position o and a particle at position p. For efficiency reasons, we calculate the force affecting one particle by selecting the closest obstacle. The effects caused by this simplification are mitigated by extending the standard potential field-based approach by relaxing the assumption that the robot is one idealized particle.

We account for the shape of the MAV by discretizing it into cells (blue grid cells in Figure 15). Every cell is considered as one particle in the algorithm. This leads to a robot model containing particles in the center of the cells p_i. The force affecting this model is the average of all individual forces

$$F_r^p = \frac{1}{N}\sum_i^N F_r^{p_i}. \tag{12}$$

The MAV velocity is modified according to the accumulated obstacle-repelling potentials of its parts and a target velocity to follow the local plan.

Our formulation allows to evaluate the effects of the potential field on the robot's orientation according to Eq. 13. As the robot is represented as a discretized 3D model, we can calculate the angular momentum on the MAV's center by all artificial forces $\vec{F_i}$ applied to the individual robot cells i and their respective relative

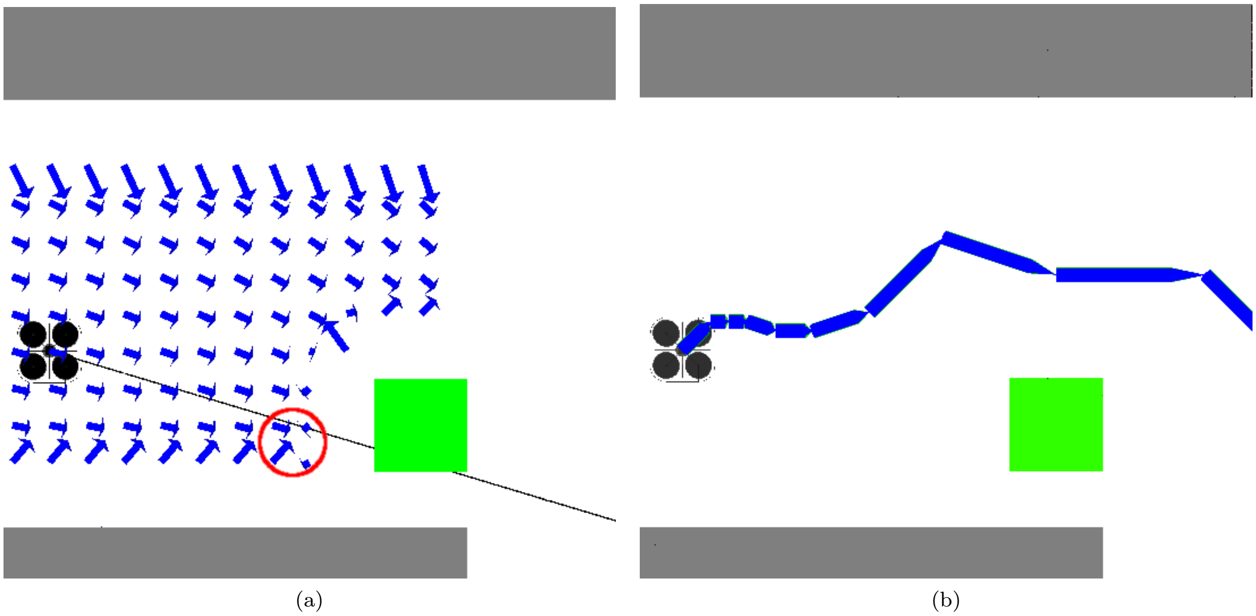


Figure 14: Our reactive collision avoidance layer is a fast mean to react on a priori unknown obstacles (green rectangle in this example). (a) The arrows depict the artificial potential forces pushing the MAV away from all static and dynamic obstacles, but the MAV will get stuck in a local minimum (red circle) while approaching a waypoint in the direction of the black line. (b) The local path planning layer is necessary to proceed without planning a new global path.

positions $\vec{p_i}$:

$$\vec{M} = \sum_i \vec{p_i} \times \vec{F_i}. \tag{13}$$

From the three rotational velocities, only the yaw velocity can be chosen independently from the linear velocities of the MAV. Hence, we project all p_i and F_i to a plane parallel to the ground. Thus, we get an acceleration around the z-axis resulting in an angular velocity. This can be used to orient less circular robots (e.g., an MAV with a sensor pole) away from obstacles. As our MAV is appriximative circular, we give precedence to the orientation commanded by higher layers.

To take the MAV's dynamic state into account, we predict its future trajectory T_t by predicting the probable sequence of motion commands $u_{t:t+n}$ for a fixed discrete-time horizon n (Figure 15). The prediction of the future trajectory for the next n time steps, initial motion command u_0, initial position p_0, and a nonlinear discrete time-invariant motion model $f(x, u)$ is then given by

$$\begin{aligned} T_t &= p_{t:t+n} = (p_t, p_t + 1, \ldots, p_{t+n}), && (14)\\ p_{i+1} &= f(x_i, u_i) + p_i \qquad i \in [t : t+n-1], && (15)\\ u_i &= C\vec{F_{p_i}}. && (16) \end{aligned}$$

The future control commands u_i are predicted by mapping the estimated forces $\vec{F_{p_i}}$ at a position p_i to a control command with matrix C.

Due to the input bounding of the model, the motion state cannot be changed instantaneously. Analysis of the model pledges a maximum lateral acceleration of $a_{max} = 1.988 \frac{m}{s^2}$ which leads to a maximum stopping distance of 0.754 m in 0.503 s under ideal conditions. Considering unmodeled effects like wind, a prediction horizon of $t = 1$ s is found to be sufficient. For more details regarding the motion model, see Beul et al. (2014).

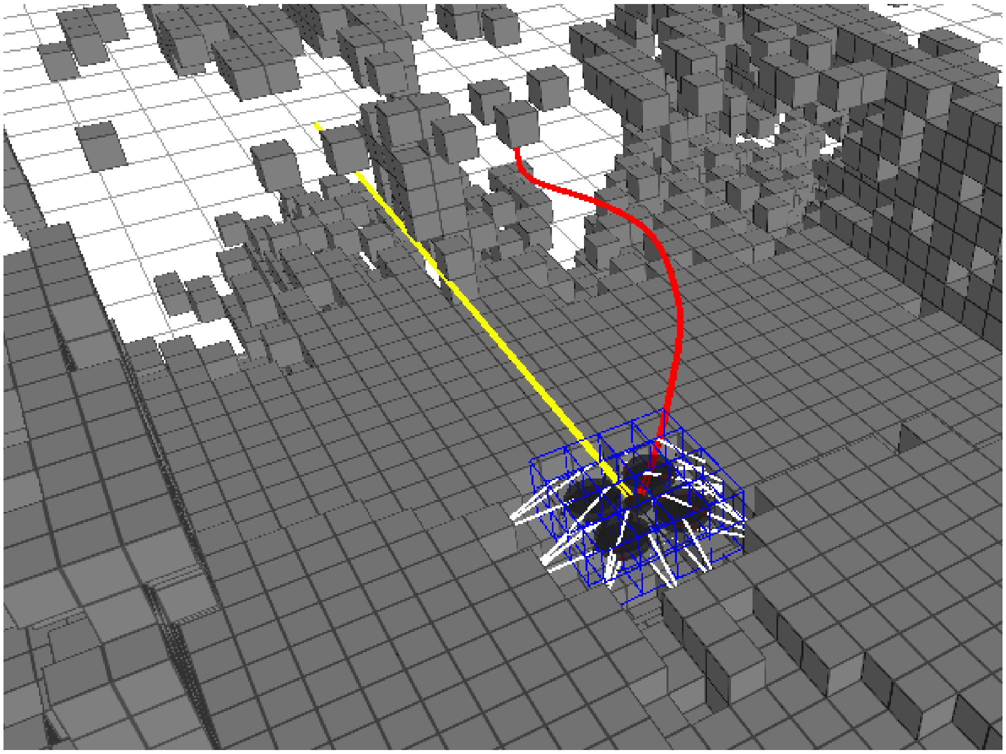


Figure 15: We predict the influence of a motion command by rolling out the robot's trajectory (red) using a learned motion model. The direct line towards the next specified waypoint is depicted in yellow. The white lines connect obstacles to the parts of the robot model, their artificial force is applied to.

If a given force threshold is exceeded at any point p_i of the trajectory, we reduce the velocity v of the MAV to

$$v_{new} = \left(\frac{1}{2} + \frac{i}{2n}\right) v_{max}. \tag{17}$$

For more details, see Nieuwenhuisen et al. (2013).

6 Evaluation

In order to assess the performance and the reliability of our system as well as the involved components, we conducted a set of experiments. These range from testing individual components in isolation to reporting the results of an integrated mission where the complete system accomplishes the mapping of a building and its surroundings.

Going through the different layers of our architecture, we first report the results of experiments on local and allocentric mapping and then continue with global and local path planning as well as reactive collision avoidance. Finally, we showcase a typical mission of the integrated experiment and present the achievable results.

6.1 Scan Registration and Local Multiresolution Mapping

In order to assess the performance of our local multiresolution mapping approach, we have recorded two datasets in-flight with our MAV. We register the point set of each acquired 3D scan with an iteratively updated local multiresolution map using I) our surfel-based registration method, II) the Iterative Closest Point algorithm (ICP) by Besl and McKay (1992) as a baseline registration algorithm, and III) the state-of-the-art Generalized ICP (GICP) by Segal et al. (2009). Throughout the experiments, we use four levels for the local multiresolution map with a cell length of 0.25 m at the finest level, which yields a cell length of 2 m at the coarsest level. The different approaches are compared in terms of runtime and accuracy. The latter is evaluated using the accuracy of the determined poses along the trajectory of the MAV and the quality of the resulting map. We make both datasets publicly available[1].

6.1.1 Indoor Flight in the Motion Caption System

The first dataset provides ground-truth pose information from an indoor motion capture (MoCap) system. The MoCap system provides accurate pose information of the MAV at high frame rates (100 Hz), but is restricted to a small capture volume of approximately 2×2×3 m. During the 46 s flight, visual odometry and laser data for 92 3D scans have been recorded. In order to assess the accuracy of our approach, we compare the estimated trajectories by the different registration algorithms with ground truth. In addition, we propose an entropy-based metric to measure the quality of the resulting map.

We quantify the accuracy of the estimated trajectories using the measure of the Absolute Trajectory Error (ATE) proposed by Sturm et al. (2012). It is based on determining relative and absolute differences between estimated and ground-truth poses. Global consistency is measured by first aligning and then directly comparing absolute pose estimates (and trajectories):

$$\mathrm{ATE}\left(F_{i:n}\right) := \left(\frac{1}{m}\sum_{i=1}^{m} \|trans\left(F_i\left(\Delta\right)\right)\|^2\right)^{1/2} \tag{18}$$

with $F_i(\Delta) := Q_i^{-1} S P_i$, where S is the rigid-body transformation mapping the estimated trajectory $P_{i:n}$ to the ground truth trajectory $Q_{i:n}$.

In order to measure the quality of the resulting map, we calculate the *mean map entropy*, a quantitative measure which evaluates the sharpness of a map. The entropy h for a map point q_k is calculated by

$$h(q_k) = \frac{1}{2}\ln\ |2\pi e\,\Sigma(q_k)|, \tag{19}$$

where $\Sigma(q_k)$ is the sample covariance of mapped points in a local radius r around q_k. We select $r = 0.3$ m in our evaluation. The mean map entropy $H(\mathcal{Q})$ is averaged over all map points

$$H(\mathcal{Q}) = \frac{1}{Q}\sum_{k=1}^{Q} h(q_k). \tag{20}$$

Table 1 summarizes the ATE of our method together with the estimated map entropy and measured runtimes, and comparing them to visual odometry and registration using ICP and GICP. The results indicate that all scan registration methods improve the motion estimate produced by visual odometry. Our method results in a lower ATE compared to ICP and GICP. Furthermore, a lower map entropy indicates a higher quality of the resulting map in terms of sharper planes, edges, corners and other environmental structures. In addition, the run-times reported in Table 1 demonstrate that our method is computationally more efficient. In Figure 16, we show the trajectory estimates obtained from the two best registration methods (ours and GICP) as well as the deviations of both estimates from the ground-truth trajectory.

6.1.2 Flight in a Parking Garage

A second dataset has been acquired in a parking garage which allows for larger flight distances. The dataset consists of 200 3D scans. The overall trajectory length is 73 m, covering the complete floor of the garage.

[1] Datasets recorded in-flight with our MAV are available at: `http://www.ais.uni-bonn.de/mav_mapping`.

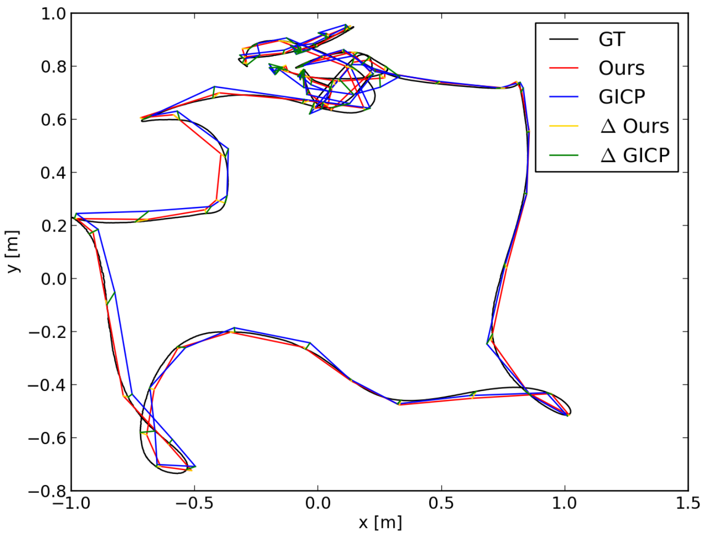


Figure 16: Absolute trajectory error of scan registration using the multiresolution map. Poses of the trajectory are projected on the xy-plane. While both approaches adequateley reconstruct the trajectory, our surfel-based methods is closer to groung-truth (GT) than Generalized-ICP (GICP).

Table 1: Scan registration results. Trajectory error, map entropy, and runtimes.

	ATE (m)						map entropy	run-time (ms)		
	RMSE	mean	median	std	min	max	mean	mean	std	max
VO	0.151	0.134	0.129	0.059	0.024	0.324	-3.112			
ICP	0.040	0.035	0.034	0.019	0.006	0.117	-3.411	290.31	108.72	521
GICP	0.034	0.031	0.030	0.014	**0.005**	0.088	-3.363	1769.52	813.92	5805
ours	**0.021**	**0.019**	**0.016**	**0.010**	**0.005**	**0.061**	**-3.572**	**51.06**	**27.30**	**121**

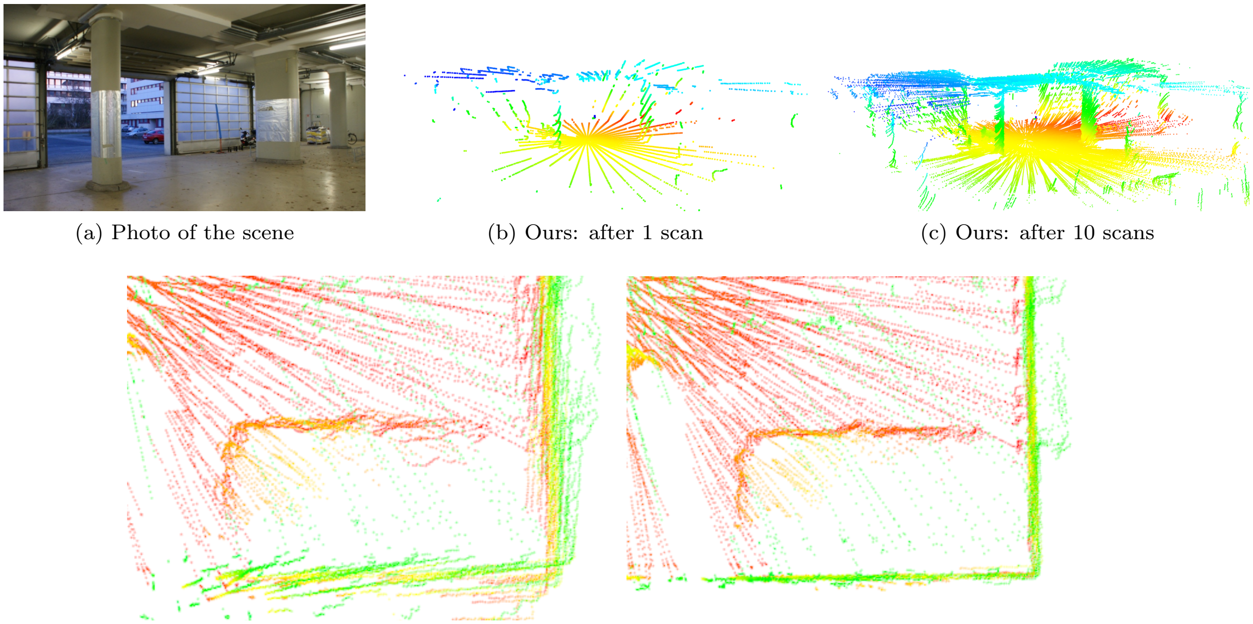
Visual Odometry (VO), Iterative Closest Point (ICP) and Generalized-ICP (GICP)

Ground truth information is not available in this environment due to the lack of a MoCap system and no availability of Global Navigation Satellite Systems (GNSS) in indoor environments. Instead, we only use the entropy of the resulting map to measure its quality. In addition, we visually inspect the resulting maps and present qualitative results. Using GICP to estimate the motion and building the map, results in a map entropy of -3.438, whereas using our method results in a lower entropy of -3.696. Figure 17 illustrates the increase of measurement density through the aggregation of measurements as well as the higher map accuracy of our approaches, compared to GICP.

6.2 Global Registration and Allocentric Mapping

In order to assess the performance of our global registration and allocentric mapping approach, we tested our approach on the complete dataset of the parking garage (Section 6.1.2).

Figure 18 shows the resulting allocentric maps and trajectories after pose graph optimization using the different registration approaches. For this figure, we chose an orthogonal top-down perspective to get an indication about the consistency of the aligned 3D scans by the parallel walls. Without pose graph optimization, the trajectory aggregates drift which results in inconsistencies, indicated by the misalignment of the walls.



(a) Photo of the scene

(b) Ours: after 1 scan

(c) Ours: after 10 scans

(d) GICP: map detail of one corner

(e) Ours: map detail of one corner

Figure 17: (a) Registration and local multiresolution surfel mapping in a parking garage. Ground truth or GNSS-based pose estimates are not available in this environment. (b+c) Registering and aggregating scans increases the density of points without loosing accuracy. (d+e) Our approach obtains higher quality maps as indicated by lower map entropy and sharper environmental structures.

Similar to the local registration experiments, we also compare our method to GICP (Segal et al., 2009). Note that the pipeline for assembling scans by visual odometry and pose graph optimization to globally align the local dense 3D maps is used in the same way, to have a fair comparison. For GICP, the resulting 3D map is less accurate and smeared. Our method is computationally more efficient with a runtime of $145 \pm 50\,\mathrm{ms}$ averaged over the complete data set, compared to $1555 \pm 613\,\mathrm{ms}$ for GICP.

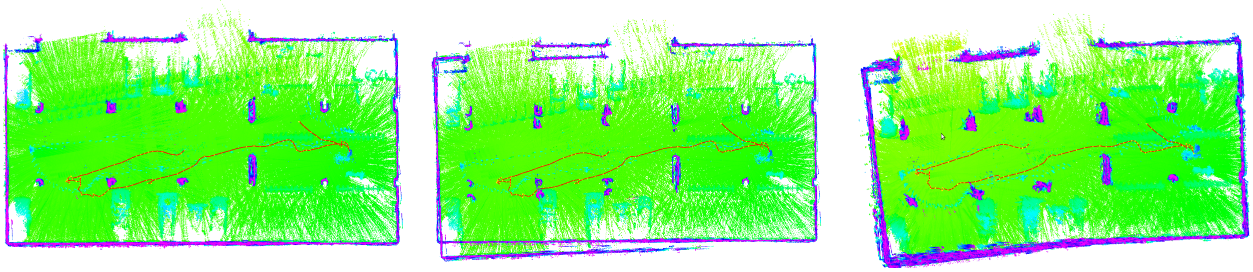
Figure 19 shows the resulting map from different perspectives, which allows for a better interpretation of the scene. Here, cars and pillars in the parking garage can be identified in the globally aligned 3D scans. It can also be seen that even lamps hanging from the ceiling are modeled by the extracted allocentric 3D point cloud.

6.3 Global Path Planning

We tested our global path planner in simulation and with the real MAV. In the real robot experiments, our MAV had to follow the planned paths employing a position controller and GPS. We extended the allocentric map derived from city and elevation model with obstacles (lanterns) for these experiments. Replanning was performed at 0.2 Hz and all components were running on the onboard computer. Our MAV was able to follow the planned paths collision free. Figure 20 shows an example from the test runs where the MAV was pushed from the planned path by a gust of wind—resulting in a qualitatively different path after replanning.

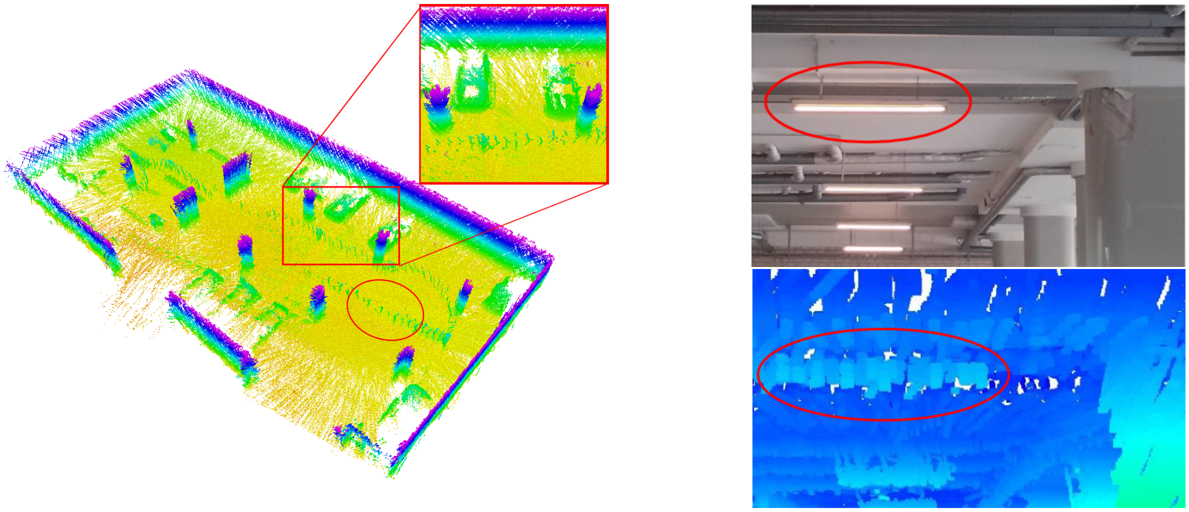
6.4 Local Path Planning

We evaluate the computing time and the resulting flight trajectories in simulation. The MAV follows a globally planned path and has to avoid obstacles that are not in the a priori known world model (Figure 21). The experiments were performed with two different uniform grids with cell size 0.25 m and 1 m, respectively.



(a) Ours with graph optimization.
(b) Ours without graph optimization.
(c) GICP with graph optimization.

Figure 18: Top-down views of the resulting maps: our surfel registration method with graph optimization (a) yields accurate results, whereas the maps of the other methods show inconsistencies (b) and drift (c).



(a) Accurately mapped environmental structures and cars.
(b) Ceiling structure with mapped lights.

Figure 19: Impressions of the quality of the built 3D map. Environmental structures are consistently mapped. Even details such as car silhouettes (a) and lamps hanging from the ceiling (b) are accurately modeled.

These were compared to our local multiresolution grid with a minimum cell size of 0.25 m and 8 cells per grid. The timings in Table 2 are measured with the MAV onboard computer.

All planning representations perform equally well if the globally planned path can be followed. In the case newly perceived obstacles have to be avoided, the planning time for a uniform grid with high resolution substantially exceeds the time window for replanning. In contrast, the local multiresolution planning is always fast enough for continuous replanning.

Table 2 summarizes the resulting path lengths for a case where the MAV has to locally plan a detour around an obstacle, not represented in the static environment model using the three planning representations. The path lengths are normalized for comparability between test runs. Local obstacle avoidance without global replanning results in 3% longer paths by means of our proposed multiresolution grid instead of the fine uniform grid. The coarse uniform grid results in 9% longer paths.

6.5 Local Obstacle Avoidance

We evaluate the performance and reliability of our predictive collision avoidance module in simulation and on the real system.

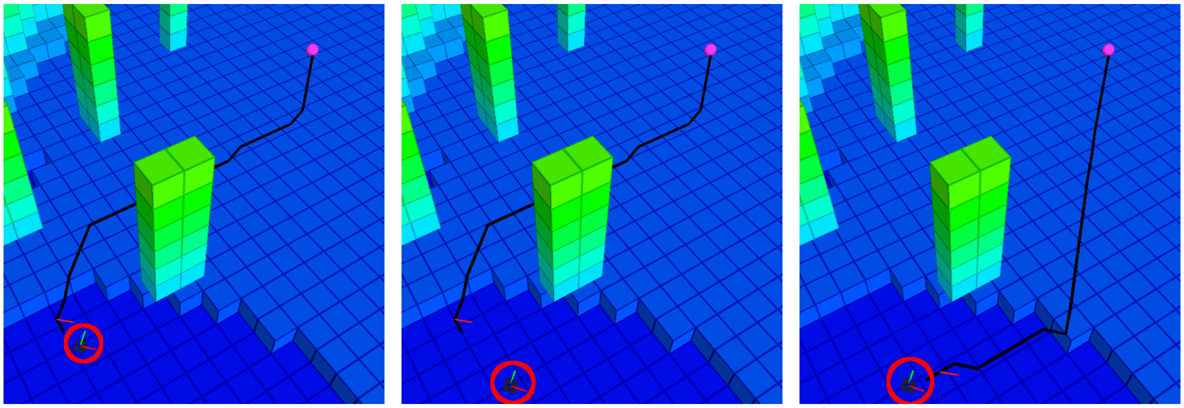

(a) (b) (c)

Figure 20: (a) The MAV follows an allocentric planned path (black lines) given a static environment map. (b) The MAV was disturbed by a strong gust of wind. (c) By continuous replanning, a new cost-optimal path to the goal waypoint could be found (c) and the MAV reached the goal collision free. The red circle highlights the position of the MAV. The magenta dot indicates the goal, and the red arrow indicates the next intermediate waypoint to reach.

Table 2: Planning time of local path planner and normalized lengths of resulting trajectories.

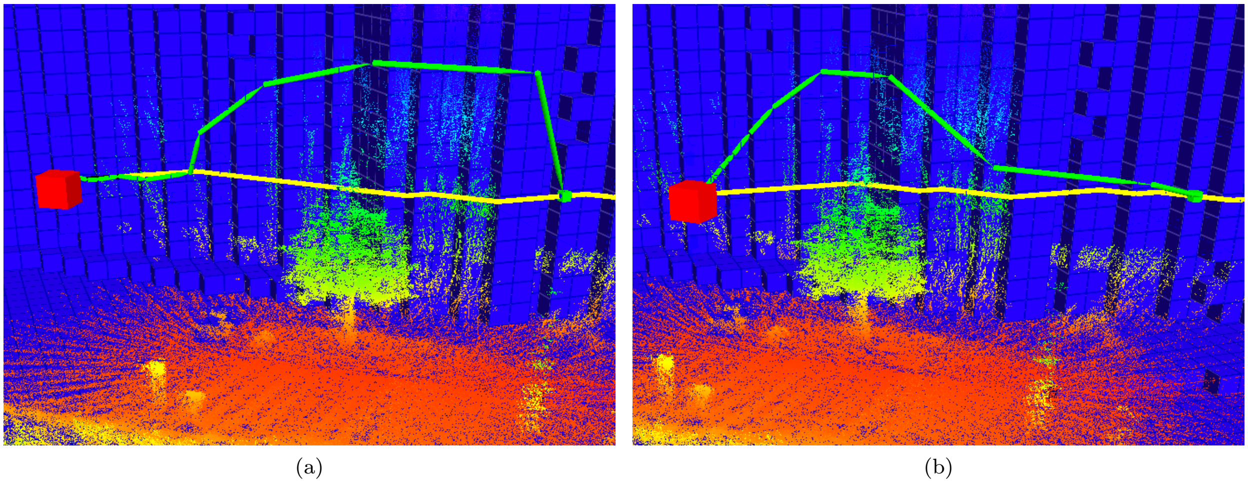
grid representation	**cell size**	**planning time**		**length**
		min.	max.	
multiresolution	0.25 m	12 ms	35 ms	1.03 m
uniform	0.25 m	26 ms	3395 ms	1.00 m
uniform	1.00 m	4 ms	20 ms	1.09 m

We tested our approach on a simulated waypoint following scenario. In this scenario, the robot had to follow a path through three walls with window-like openings of different size. We measured the time the MAV needed and the forces repelling the MAV from obstacles during the flight. The forces are a measure on how close the MAV comes to the obstacles. We compared our approach with the classical potential field approach. Furthermore, we implemented a fixed slow down of the MAV. Here, the MAVs maximum speed is reduced by a fixed factor if the forces along the predicted trajectory cross a threshold at any time. In our evaluation, we tested this approach without prediction, i.e., just the forces in the current state are estimated, and with a 1 s trajectory rollout. We show the results in Table 3.

These experiments show that our predictive collision avoidance leads to smoother trajectories, keeping the MAV further away from obstacles than the same potential field approach without trajectory prediction. No collisions occurred during these test runs. The simulated MAV was able to fly through passageways of its size plus a safety margin. We also evaluated our approach with the real robot. Our collision avoidance approach runs at approximately 100 Hz on a single core of an Intel Core 2 processor, which includes data acquisition and map building. Figure 22 shows an experiment where the hovering MAV avoided approaching or static obstacles.

Table 3: Evaluation of flight durations and repelling forces applied to the MAV by the reactive obstacle avoidance.

	Time (s)	Avg. Force
Std. Potential Fields	11.90 (0.5)	0.44 (0.06)
Fixed Slow Down	12.56 (0.8)	0.43 (0.04)
Fixed Slow Down (1 s look-ahead)	14.30 (1.7)	0.28 (0.04)
Adaptive Velocities (1 s look-ahead)	12.90 (0.8)	0.30 (0.01)


(a)

(b)

Figure 21: A global path planner plans allocentric paths (yellow lines) based on an environment model known in advance at a rate of 0.2 Hz. A local multiresolution path planner refines these plans locally, based on onboard maps incorporating sensor information (colored measurements) at 2 Hz. The green lines depict the local deviation from the global path to avoid a tree not represented in the static environment model (blue boxes). From (a) to (b) the MAV (red box) approaches the tree and the detour is planned at a finer resolution. The movement direction is from left to right in these figures.

6.6 Evaluation of the Integrated System

We evaluated the integration of our components into one working MAV mapping system. Here, we describe one exemplary mapping session of our MAV. The main goal of this experiment was to autonomously map an old manor house as shown in Figure 24a. The scenario involves vegetation such as trees and bushes and is difficult to traverse from all sides by humans. Therefore, a manual flight by a human pilot controlling the MAV was not possible in this scenario.

The user manually defined a set of mission-relevant view poses given the coarse LoD 2 world model. These view poses where roughly specified to cover the building facade from all sides. Then, they are processed by a mission control layer, incorporating the mission planner. Figure 11 shows a solution for the mission described in this section. After takeoff, the global planner begins to continuously plan paths to the next mission relevant pose. Figure 12 shows a local deviation from the allocentric path while returning to the start position. The allocentric world model is updated during the mission.

The overall flight duration was approximately 8 minutes and the traversed trajectory is shown in Figure 23. The MAV successfully traversed the building in all experiments without colliding with an obstacle.

Resulting 3D maps of the global registration are shown in Figure 24. It can be seen that our method is able to successfully map the building. Even details of the structure of the facade, such as windows and doors are represented in the map.

7 Conclusions

In this article, we presented an integrated system to autonomously operate MAVs safely in the vicinity of obstacles. We approached this challenge by employing local multiresolution mapping and planning techniques that facilitate frequent updates and replanning.

We showed that by incorporating multimodal sensor information we are able to detect and avoid diverse obstacles. Pose estimation based on camera and laser data enables robust motion control in GPS-denied

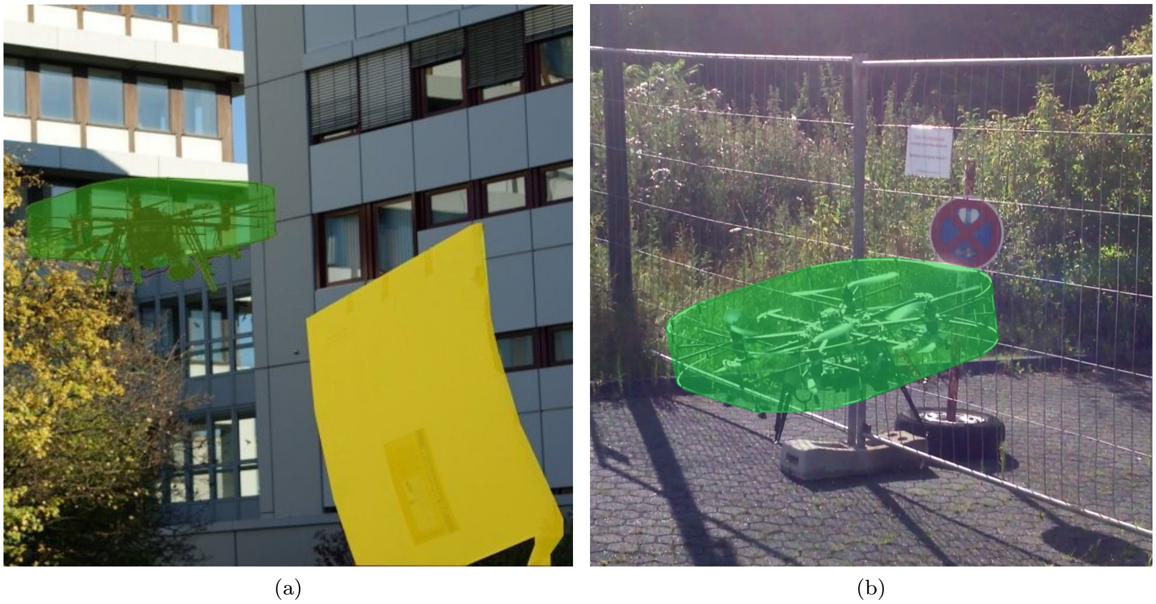


Figure 22: (a) The hovering MAV (green) avoids collisions with approaching obstacles (yellow). (b) The MAV is repelled by a static obstacle (fence) while a human operator commands it towards it.

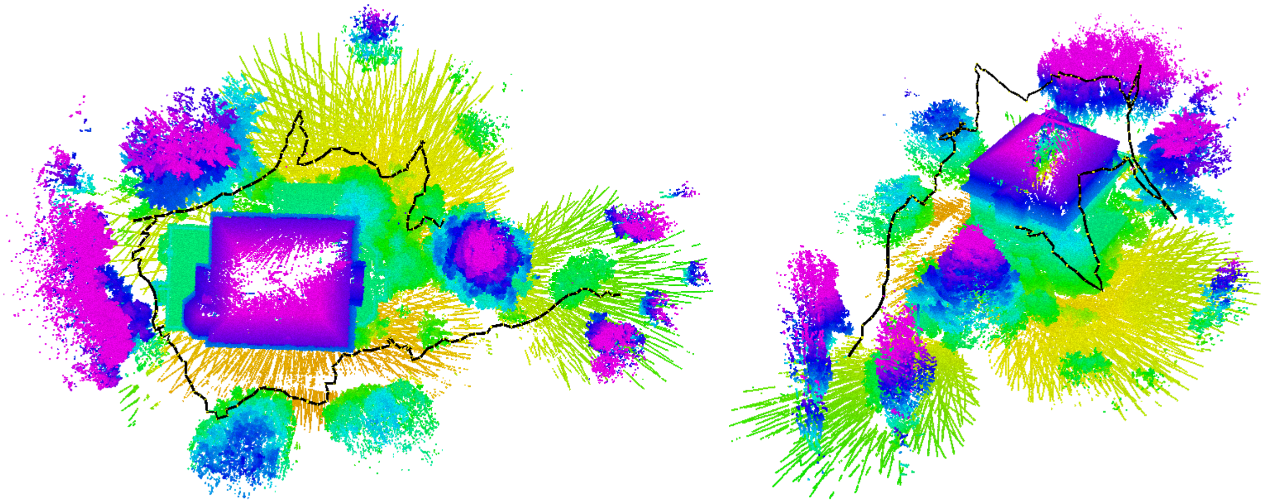
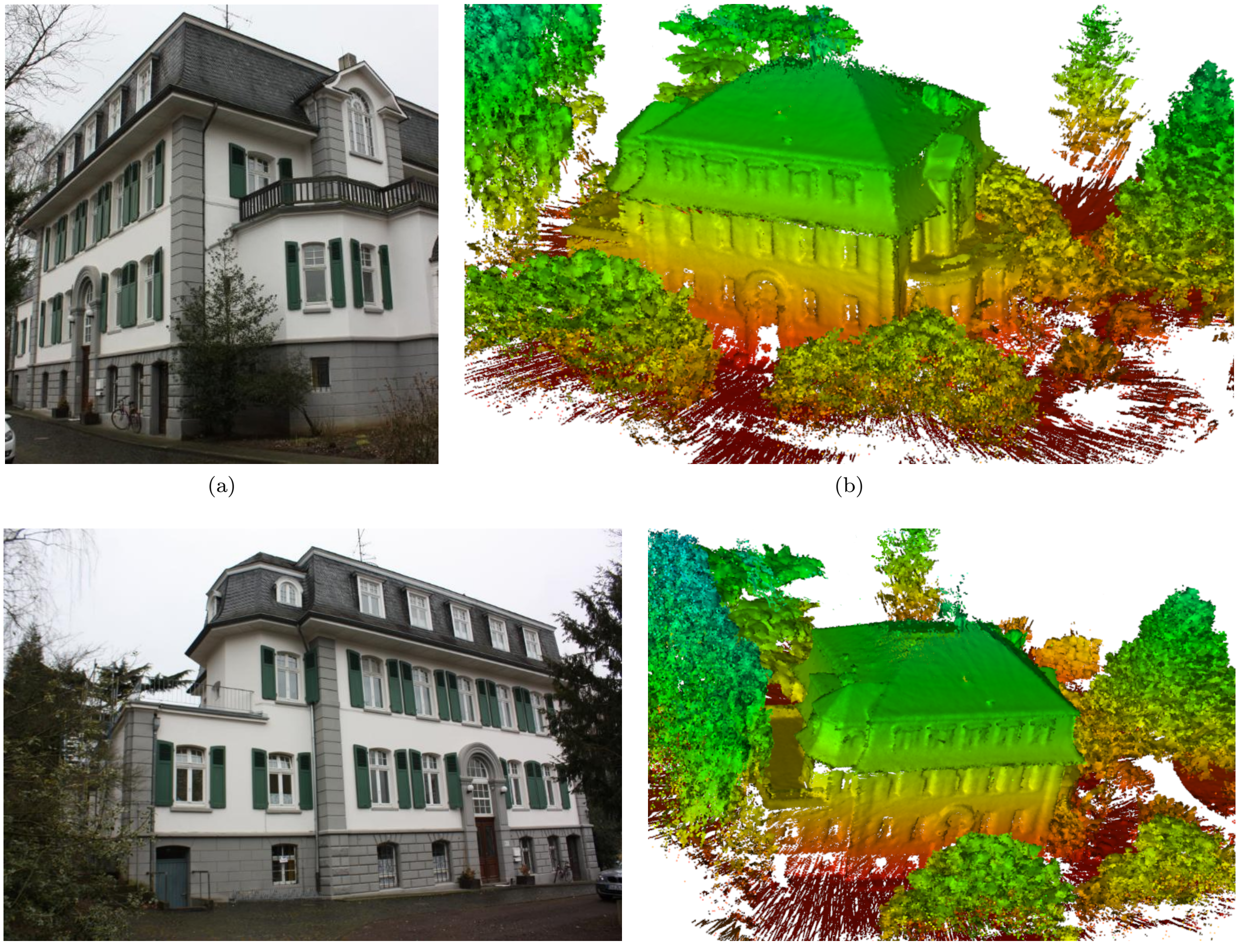


Figure 23: Evaluation of the integrated system by mapping an old manor house. The resulting trajectory (black) traversed by the MAV from top and side view. Color encodes the height from the ground.

environments.

Laser-range measurements are aggregated by registering sparse 3D scans with a local multiresolution surfel map. Modeling measurement distributions within voxels by surface elements allows for efficient and accurate registration of 3D scans with the local map. The incrementally built local dense 3D maps of nearby key poses are registered globally by graph optimization. This yields a globally consistent dense 3D map of the environment. We demonstrate accuracy and efficiency of our approach by showing consistent allocentric 3D maps, recorded by our MAV during flight.

Furthermore, we show that multilayered navigation planning yields a high potential to cope with dynamically changing environments and perpetually new obstacle perceptions. By employing a global-to-local approach in



(a)

(b)

(c)

(d)

Figure 24: Evaluation of the integrated system by mapping an old manor house. (a + c) Photos of the scene. (b + d) the resulting global maps after the mission. Color encodes the height from the ground.

our navigation planning pipeline, we achieve replanning frequencies that match the rate of expected changes in the environment model. A reactive collision avoidance layer accounts for fast MAV and environment dynamics and refines higher-level mission plans based on onboard sensing and a priori information.

We could demonstrate both the efficiency of the involved components as well as the reliability of the overall system by autonomously accomplishing a mission to map a building and its surroundings while flying in the vicinity of buildings, trees, cables and other potential obstacles and sources for collision.

Videos showing the performance of the integrated system can be seen at `www.ais.uni-bonn.de/MoD`.

Acknowledgments

This work has been supported by grants BE 2556/7 and BE 2556/8 of German Research Foundation (DFG).

References

Achtelik, M. W., Lynen, S., Weiss, S., Chli, M., and Siegwart, R. (2014). Motion- and uncertainty-aware path planning for micro aerial vehicles. *Journal of Field Robotics*, 31(4):676–698.

Anderson, S. and Barfoot, T. D. (2013). Towards relative continuous-time SLAM. In *Proc. of the IEEE Int. Conference on Robotics and Automation (ICRA)*, pages 1033–1040.

Andert, F., Adolf, F.-M., Goormann, L., and Dittrich, J. S. (2010). Autonomous vision-based helicopter flights through obstacle gates. In *Selected papers from the 2nd International Symposium on UAVs*, pages 259–280. Springer.

Applegate, D., Bixby, R., Chvatal, V., and Cook, W. (2006). Concorde TSP solver.

Bachrach, A., He, R., and Roy, N. (2009). Autonomous flight in unstructured and unknown indoor environments. In *Proc. of European Micro Aerial Vehicle Conf. (EMAV)*.

Bachrach, A., Prentice, S., He, R., Henry, P., Huang, A. S., Krainin, M., Maturana, D., Fox, D., and Roy, N. (2012). Estimation, planning, and mapping for autonomous flight using an RGB-D camera in GPS-denied environments. *The International Journal of Robotic Research*, 31(11):1320–1343.

Becker, M., Sampaio, R. C. B., Bouabdallah, S., Perrot, V., and Siegwart, R. (2012). In flight collision avoidance for a mini-uav robot based on onboard sensors. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*.

Behnke, S. (2004). Local multiresolution path planning. *Robocup 2003: Robot Soccer World Cup VII*, pages 332–343.

Besl, P. J. and McKay, N. D. (1992). Method for registration of 3-d shapes. In *Robotics-DL tentative*, pages 586–606. International Society for Optics and Photonics.

Beul, M., Behnke, S., and Worst, R. (2014). Nonlinear model-based 2D-position control for quadrotor UAVs. In *Proceedings of the Joint Int. Symposium on Robotics (ISR) and the German Conference on Robotics (ROBOTIK)*.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Bosse, M. and Zlot, R. (2009). Continuous 3D scan-matching with a spinning 2D laser. In *Proc. of the IEEE Int. Conference on Robotics and Automation (ICRA)*, pages 4312–4319.

Bouabdallah, S., Murrieri, P., and Siegwart, R. (2004). Design and control of an indoor micro quadrotor. In *Proc. of the IEEE Int. Conference on Robotics and Automation (ICRA)*.

Chambers, A., Achar, S., Nuske, S., Rehder, J., Kitt, B., Chamberlain, L., Haines, J., Scherer, S., , and Singh, S. (2011). Perception for a river mapping robot. In *Proc. of the IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS)*.

Cover, H., Choudhury, S., Scherer, S., and Singh, S. (2013). Sparse tangential network (SPARTAN): Motion planning for micro aerial vehicles. In *Proc. of the IEEE Int. Conference on Robotics and Automation (ICRA)*.

De Wagter, C., Tijmons, S., Remes, B., and de Croon, G. (2014). Autonomous flight of a 20-gram flapping wing MAV with a 4-gram onboard stereo vision system. In *IEEE International Conference on Robotics and Automation (ICRA)*.

Droeschel, D., Schreiber, M., and Behnke, S. (2013). Omnidirectional perception for lightweight uavs using a continuous rotating laser scanner. In *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci. (ISPRS)*, volume XL-1/W2, pages 107–112.

Droeschel, D., Stückler, J., and Behnke, S. (2014). Local multi-resolution representation for 6D motion estimation and mapping with a continuously rotating 3D laser scanner. In *Proc. of the IEEE Int. Conference on Robotics and Automation (ICRA)*.

Elseberg, J., Borrmann, D., and Nuechter, A. (2012). 6DOF semi-rigid SLAM for mobile scanning. In *Proc. of the IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS)*, pages 1865–1870.

Flores, G., Zhou, S., Lozano, R., and Castillo, P. (2014). A vision and GPS-based real-time trajectory planning for a MAV in unknown and low-sunlight environments. *Journal of Intelligent & Robotic Systems*, 74(1-2):59–67.

Fossel, J., Hennes, D., Claes, D., Alers, S., and Tuyls, K. (2013). Octoslam: A 3d mapping approach to situational awareness of unmanned aerial vehicles. In *Proc. of the Int. Conference on Unmanned Aircraft Systems (ICUAS)*, pages 179–188.

Ge, S. and Cui, Y. (2002). Dynamic motion planning for mobile robots using potential field method. *Autonomous Robots*, 13(3):207–222.

Geiger, A., Roser, M., and Urtasun, R. (2010). Efficient large-scale stereo matching. In *Asian Conference on Computer Vision (ACCV)*.

Goerzen, C., Kong, Z., and Mettler, B. (2010). A survey of motion planning algorithms from the perspective of autonomous uav guidance. *Journal of Intelligent & Robotic Systems*, 57(1-4):65–100.

Green, W. and Oh, P. (2008). Optic-flow-based collision avoidance. *Robotics Automation Magazine, IEEE*, 15(1):96–103.

Gröger, G., Kolbe, T. H., Czerwinski, A., and Nagel, C. (2008). Opengis city geography markup language (citygml) encoding standard. *Open Geospatial Consortium Inc. Reference number of this OGC® project document: OGC*.

Grzonka, S., Grisetti, G., and Burgard, W. (2009). Towards a navigation system for autonomous indoor flying. In *Proc. of the IEEE Int. Conference on Robotics and Automation (ICRA)*.

Grzonka, S., Grisetti, G., and Burgard, W. (2012). A fully autonomous indoor quadrotor. *IEEE Trans. on Robotics*, 28(1):90–100.

Hart, P. E., Nilsson, N. J., and Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. on Systems Science and Cybernetics*, 4(2):100–107.

Heng, L., Honegger, D., Lee, G. H., Meier, L., Tanskanen, P., Fraundorfer, F., and Pollefeys, M. (2014). Autonomous visual mapping and exploration with a micro aerial vehicle. *Journal of Field Robotics*, 31(4):654–675.

Holz, D., Nieuwenhuisen, M., Droeschel, D., Schreiber, M., and Behnke, S. (2013). Towards mulimodal omnidirectional obstacle detection for autonomous unmanned aerial vehicles. In *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci. (ISPRS)*, volume XL-1/W2, pages 201–206.

Honegger, D., Meier, L., Tanskanen, P., and Pollefeys, M. (2013). An open source and open hardware embedded metric optical flow cmos camera for indoor and outdoor applications. In *Proc. of the IEEE Int. Conference on Robotics and Automation (ICRA)*.

Hornung, A., Wurm, K. M., Bennewitz, M., Stachniss, C., and Burgard, W. (2013a). OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*.

Hornung, A., Wurm, K. M., Bennewitz, M., Stachniss, C., and Burgard, W. (2013b). OctoMap: an efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 34:189–206.

Hrabar, S., Sukhatme, G., Corke, P., Usher, K., and Roberts, J. (2005). Combined optic-flow and stereo-based navigation of urban canyons for a uav. In *Proc. of the IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS)*.

Huh, S., Shim, D., and Kim, J. (2013). Integrated navigation system using camera and gimbaled laser scanner for indoor and outdoor autonomous flight of UAVs. In *Proc. of the IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS)*, pages 3158–3163.

Johnson, E. N. and Mooney, J. G. (2014). A comparison of automatic nap-of-the-earth guidance strategies for helicopters. *Journal of Field Robotics*, 31(4):637–653.

Jutzi, B., Weinmann, M., and Meidow, J. (2014). Weighted data fusion for UAV-borne 3D mapping with camera and line laser scanner. *International Journal of Image and Data Fusion*.

Kohlbrecher, S., Meyer, J., von Stryk, O., and Klingauf, U. (2011). A flexible and scalable slam system with full 3d motion estimation. In *Proc. of the IEEE Int. Symposium on Safety, Security and Rescue Robotics (SSRR)*.

Kuemmerle, R., Grisetti, G., Strasdat, H., Konolige, K., and Burgard, W. (2011). G2o: A general framework for graph optimization. In *Proc. of the IEEE Int. Conference on Robotics and Automation (ICRA)*, pages 3607–3613.

MacAllister, B., Butzke, J., Kushleyev, A., Pandey, H., and Likhachev, M. (2013). Path planning for non-circular micro aerial vehicles in constrained environments. In *Proc. of the IEEE Int. Conference on Robotics and Automation (ICRA)*.

Maddern, W., Harrison, A., and Newman, P. (2012). Lost in translation (and rotation): Fast extrinsic calibration for 2D and 3D LIDARs. In *Proc. of the IEEE Int. Conference on Robotics and Automation (ICRA)*.

Magnusson, M., Duckett, T., and Lilienthal, A. J. (2007). Scan registration for autonomous mining vehicles using 3D-NDT. *Journal of Field Robotics*, 24(10):803–827.

Magree, D., Mooney, J. G., and Johnson, E. N. (2014). Monocular visual mapping for obstacle avoidance on UAVs. *Journal of Intelligent & Robotic Systems*, 74(1-2):17–26.

Meier, L., Tanskanen, P., Heng, L., Lee, G., Fraundorfer, F., and Pollefeys, M. (2012). PIXHAWK: A micro aerial vehicle design for autonomous flight using onboard computer vision. *Autonomous Robots*, 33(1-2):21–39.

Montemerlo, M., Becker, J., Bhat, S., Dahlkamp, H., Dolgov, D., Ettinger, S., Haehnel, D., Hilden, T., Hoffmann, G., Huhnke, B., et al. (2008). Junior: The stanford entry in the urban challenge. *Journal of Field Robotics*, 25(9):569–597.

Moore, R., Dantu, K., Barrows, G., and Nagpal, R. (2014). Autonomous MAV guidance with a lightweight omnidirectional vision sensor. In *Proc. of the IEEE Int. Conference on Robotics and Automation (ICRA)*.

Mori, T. and Scherer, S. (2013). First results in detecting and avoiding frontal obstacles from a monocular camera for micro unmanned aerial vehicles. In *Proc. of the IEEE Int. Conference on Robotics and Automation (ICRA)*.

Morris, W., Dryanovski, I., Xiao, J., and Member, S. (2010). 3d indoor mapping for micro-uavs using hybrid range finders and multi-volume occupancy grids. In *In RSS 2010 workshop on RGB-D: Advanced Reasoning with Depth Cameras*.

Myronenko, A. and Song, X. (2010). Point set registration: Coherent point drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(12):2262–2275.

Nieuwenhuisen, M., Droeschel, D., Schneider, J., Holz, D., Läbe, T., and Behnke, S. (2013). Multimodal obstacle detection and collision avoidance for micro aerial vehicles. In *Proc. of the European Conference on Mobile Robots (ECMR)*.

Nuechter, A., Lingemann, K., Hertzberg, J., and Surmann, H. (2005). 6D SLAM with approximate data association. In *Int. Conf. on Advanced Robotics*, pages 242 – 249.

Ok, K., Ansari, S., Gallagher, B., Sica, W., Dellaert, F., and Stilman, M. (2013). Path planning with uncertainty: Voronoi uncertainty fields. In *Proc. of the IEEE Int. Conference on Robotics and Automation (ICRA)*.

Park, J. and Kim, Y. (2014). 3d shape mapping of obstacle using stereo vision sensor on quadrotor uav. In *AIAA Guidance, Navigation, and Control Conference*.

Puls, T., Kemper, M., Kuke, R., and Hein, A. (2009). GPS-based position control and waypoint navigation system for quadrocopters. In *Proc. of the IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS)*.

Quigley, M., Conley, K., Gerkey, B. P., Faust, J., Foote, T., Leibs, J., Wheeler, R., and Ng, A. Y. (2009). Ros: an open-source robot operating system. In *ICRA Workshop on Open Source Software*.

Ross, S., Melik-Barkhudarov, N., Shankar, K. S., Wendel, A., Dey, D., Bagnell, J. A., and Hebert, M. (2013). Learning monocular reactive uav control in cluttered natural environments. In *Proc. of the IEEE Int. Conference on Robotics and Automation (ICRA)*.

Ryde, J. and Hu, H. (2010). 3D mapping with multi-resolution occupied voxel lists. *Autonomous Robots*, 28:169 – 185.

Schadler, M., Stückler, J., and Behnke, S. (2013). Multi-resolution surfel mapping and real-time pose tracking using a continuously rotating 2D laser scanner. In *Proceedings of 11th IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*.

Schauwecker, K. and Zell, A. (2014). On-board dual-stereo-vision for the navigation of an autonomous MAV. *Journal of Intelligent & Robotic Systems*, 74(1-2):1–16.

Scherer, S., Rehder, J., Achar, S., Cover, H., Chambers, A. D., Nuske, S. T., and Singh, S. (2012). River mapping from a flying robot: State estimation, river detection, and obstacle mapping. *Autonomous Robots*, 32(5):1–26.

Schmid, K., Lutz, P., Tomic, T., Mair, E., and Hirschmüller, H. (2014). Autonomous vision-based micro air vehicle for indoor and outdoor navigation. *Journal of Field Robotics*, 31(4):537–570.

Schneider, J., Läbe, T., and Förstner, W. (2013). Incremental real-time bundle adjustment for multi-camera systems with points at infinity. In *ISPRS Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume XL-1/W2.

Segal, A., Haehnel, D., and Thrun, S. (2009). Generalized-ICP. In *Proc. of Robotics: Science and Systems (RSS)*.

Shen, S., Michael, N., and Kumar, V. (2011). Autonomous multi-floor indoor navigation with a computationally constrained micro aerial vehicle. In *Proc. of the IEEE Int. Conference on Robotics and Automation (ICRA)*.

Stoyanov, T. and Lilienthal, A. (2009). Maximum likelihood point cloud acquisition from a mobile platform. In *Proc. of the Int. Conf. on Advanced Robotics (ICAR)*, pages 1–6.

Stückler, J. and Behnke, S. (2014). Multi-resolution surfel maps for efficient dense 3D modeling and tracking. *Journal of Visual Communication and Image Representation*, 25(1):137–147.

Sturm, J., Engelhard, N., Endres, F., Burgard, W., and Cremers, D. (2012). A benchmark for the evaluation of RGB-D SLAM systems. In *Proc. of the IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS)*.

Takahashi, M., Schulein, G., and Whalley, M. (2008). Flight control law design and development for an autonomous rotorcraft. In *Proceedings of the 64th Annual Forum of the American Helicopter Society*.

Thrun, S., Diel, M., and Hähnel, D. (2003). Scan alignment and 3-d surface modeling with a helicopter platform. In Yuta, S., Asama, H., Thrun, S., Prassler, E., and Tsubouchi, T., editors, *Int. Conf. on Field and Service Robotics (FSR)*, volume 24 of *Springer Tracts in Advanced Robotics*, pages 287–297. Springer.

Tomić, T., Schmid, K., Lutz, P., Domel, A., Kassecker, M., Mair, E., Grixa, I., Ruess, F., Suppa, M., and Burschka, D. (2012). Toward a fully autonomous UAV: Research platform for indoor and outdoor urban search and rescue. *Robotics Automation Magazine, IEEE*, 19(3):46–56.

Tripathi, A., G Raja, R., and Padhi, R. (2014). Reactive collision avoidance of UAVs with stereovision camera sensors using UKF. In *Advances in Control and Optimization of Dynamical Systems*, pages 1119–1125.

Ulrich, I. and Borenstein, J. (1998). Vfh+: Reliable obstacle avoidance for fast mobile robots. In *Proc. of the IEEE Int. Conference on Robotics and Automation (ICRA)*, volume 2, pages 1572–1577.

Whalley, M. S., Takahashi, M. D., Fletcher, J. W., Moralez, E., Ott, L. C. R., Olmstead, L. M. G., Savage, J. C., Goerzen, C. L., Schulein, G. J., Burns, H. N., and Conrad, B. (2014). Autonomous black hawk in flight: Obstacle field navigation and landing-site selection on the rascal juh-60a. *Journal of Field Robotics*, 31(4):591–616.