Procedings of 13th International Conference on Intelligent Autonomous Systems (IAS), Padova, Italy, July 2014.

Mapping with Micro Aerial Vehicles by Registration of Sparse 3D Laser Scans

Dirk Holz and Sven Behnke

Autonomous Intelligent Systems Group, University of Bonn, Germany
holz@ais.uni-bonn.de, behnke@ais.uni-bonn.de

Abstract. Micro aerial vehicles (MAVs) pose specific constraints on onboard sensing, mainly limited payload and limited processing power. For accurate 3D mapping even in GPS denied environments, we have designed a light-weight 3D laser scanner specifically for the application on MAVs. Similar to other custom-built 3D laser scanners composed of a rotating 2D laser range finder, it exhibits different point densities within and between individual scan lines. When rotated fast, such non-uniform point densities influence neighborhood searches which in turn may negatively affect local feature estimation and scan registration. We present a complete pipeline for 3D mapping including pair-wise registration and global alignment of 3D scans acquired in-flight. For registration, we extend a state-of-the-art registration algorithm to include topological information from approximate surface reconstructions. For global alignment, we use a graph-based approach making use of the same error metric and iteratively refine the complete vehicle trajectory. In experiments, we show that our approach can compensate for the effects caused by different point densities up to very low angular resolutions and that we can build accurate and consistent 3D maps in-flight with a micro aerial vehicle.

1 Introduction

Micro aerial vehicles (MAVs) such as quadrotors have attracted much attention in the field of aerial robotics in recent years. Their size and weight limitations, however, pose a problem in designing sensory systems for environment perception. Most of today's MAVs are equipped with ultrasonic sensors and camera systems due to their minimal size and weight. While these small and lightweight sensors provide valuable information, they suffer from a limited field-of-view and cameras are sensitive to illumination conditions. Only few MAVs [1–4] are equipped with 2D laser range finders (LRF) that are used for navigation. These provide accurate distance measurements to the surroundings but are limited to the two-dimensional scanning plane of the sensor. Objects below or above that plane are not perceived.

This work has been supported partially by grants BE 2556/7-1 and BE 2556/8-1 of German Research Foundation (DFG).

3D laser scanners provide robots with the ability to extract spatial information about their surroundings, detect obstacles in all directions, build 3D maps, and localize. In the course of a larger project on mapping inaccessible areas with autonomous micro aerial vehicles (MAVs), we have developed a light-weight 3D scanner [5] specifically suited for the application on MAVs. It consists of a Hokuyo 2D laser scanner, a rotary actuator and a slip ring to allow continuous rotation. Just as with other rotated scanners, the acquired point clouds (aggregated over one full or half rotation) show the particular characteristic of having non-uniform point densities: usually a high density within each scan line and a larger angle between scan lines (see Fig. 1). Since we use the laser scanner for omnidirectional obstacle detection and collision avoidance, we rotate the scanner quickly with 1 Hz, with the consequence of a particularly low angular resolution of roughly 9°. The resulting non-uniform point densities affect neighborhood searches and cause problems in local feature estimation and registration when keeping track of the MAV movement and building allocentric 3D maps.

In this paper, we present a complete processing pipeline for building globally consistent 3D maps with this sensor on a flying MAV. To compensate for the non-uniform point densities, we approximate the underlying measured surface and use this information in both initial pairwise registration of consecutive 3D scans to track the MAV movement and graph-based optimization for building a consistent and accurate 3D map. For initial registration, we extend the state-of-the-art registration algorithm Generalized-ICP [6] to include topological surface information instead of a point's 3D neighborhood. We represent the resulting trajectory in a pose graph [7] and connect neighboring poses by edges representing point-pair correspondences between scans and encoding the same error metric using topological surface information. This graph is iteratively refined, re-estimating the point correspondences in each iteration, to build a consistent 3D map.

The remainder of this paper is structured as follows. After an overview on related work in Sec. 2, we present our approaches for approximate surface reconstruction, registration and graph optimization in Sec. 3. We present experimental results that show the reliability of our approach in Sec. 4.

2 Related Work

Particularly important for the autonomous application of MAVs is the ability to perceive and avoid obstacles as well as building environment maps for planning paths and goal-directed navigation.

2.1 Mapping with Micro Aerial Vehicles

Scaramuzza et al. [8] present vision-based perception, control and mapping for a swarm of MAVs. In contrast to our work, 3D mapping is done on a ground station gathering visual keypoints from all MAVs, and dense 3D maps are reconstructed from the final trajectories off-line. Moreover, the approach is purely vision-based

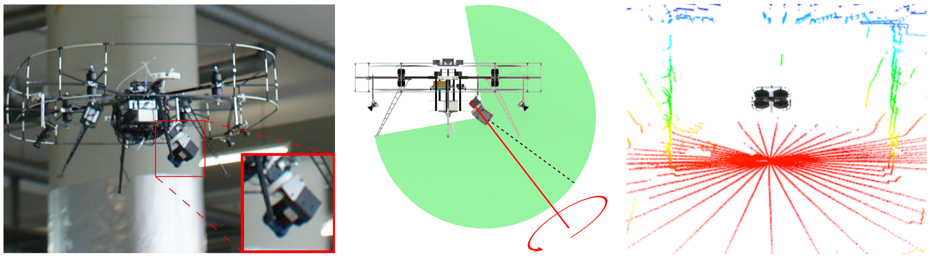

(a) MAV + scanner (b) Field-of-view (c) Example scan

Fig. 1. The laser scanner is mounted slightly below the MAV facing forwards. Continuously rotating it allows for an almost omnidirectional perception of its surroundings. The resulting 3D scans (aggregated over one half rotation using visual odometry) show different point densities within and between individual scan lines.

and restricted to downward-facing cameras whereas our approach aims at omnidirectional perception thereby allowing to map environmental structures that are not below the MAV.

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 field-of-view. For instance, autonomous cars often perceive obstacles by means of a rotating laser scanner with a 360° horizontal field-of-view, allowing for detection of obstacles in every direction [9].

Up to now, such 3D laser scanners are rarely used on lightweight MAVs—due to payload limitations. Instead, two-dimensional laser range finders [1–4, 10, 11] are used. Using a statically mounted 2D laser range finder restricts the field-of-view to the two-dimensional measurement plane of the sensor. This poses a problem especially for reliably perceiving obstacles surrounding the MAV. When moving, however, and in combination with accurate pose estimation, these sensors can very well be used to build 3D maps of the measured surfaces. Fossel et al. [12], for example, use Hector SLAM [13] for registering horizontal 2D laser scans and OctoMap [14] to build a three-dimensional occupancy model of the environment at the measured heights. Morris et al. [15] follow a similar approach and in addition use visual features to aid motion and pose estimation. 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 spinning laser range finder 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. and Cover et al. [16, 17]. 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.

2.2 3D Scan Registration

The fundamental problem in 3D map building is registration in order to align the acquired 3D laser scans and estimate the poses (positions and orientations) where the scans have been acquired. Over the past two decades, many different registration algorithms have been proposed. Prominent examples for estimating the motion of mobile ground robots using 3D scan registration are the works of Segal et al. [6], Nuechter et al. [18], and Magnusson et al. [19].

3D laser scanners built out of an actuated 2D laser range finder are usually (especially on ground robots) rotated comparably slower than ours to gain a higher and more uniform density of points. Most of the approaches to register such scans are derived from the Iterative Closest Points (ICP) algorithm [20]. Generalized-ICP (GICP) [6] unifies the ICP formulation for various error metrics such as point-to-point, point-to-plane, and plane-to-plane. The effect of using this generalized error metric is that corresponding points in two 3D laser scans are not directly dragged onto another, but onto the underlying surfaces. For our non-uniform density point clouds, GICP tends to fail since the local neighborhoods of points do not adequately represent the underlying surface. We adapt the GICP approach here to use extracted information from approximate surface reconstruction in the acquired 3D scans.

Our approach explicitly addresses the non-uniform point densities and tries to compensate for the resulting effects by using the approximated surface information. An alternative for using such sparse data in registration and mapping is to aggregate the point clouds in local maps and thereby increase the point density as is done in another work [21] within the same project on MAV-based mapping as the work at hand. Both ways constitute problems in their own right.

Bosse et al. [22] use a spring to passively articulate a 2D laser range finder and present a registration algorithm for building accurate 3D point cloud maps. Due to the passivity of the spring-based articulation, however, their sensor setup cannot guarantee complete omnidirectional point clouds at fixed controllable intervals as is the case for a continuously rotating scanner. Furthermore, it requires the carrying vehicle to move in order to impose oscillation. In terms of registration, they use a surfel-based approach and efficiently solve both the creation of accurate 3D scans (as captured by a statically mounted 3D scanner) and building globally consistent 3D maps. Since surfels are computed on local neighborhoods, the approach may suffer from the same degradation effects when applied to the non-uniform density data of our sensor.

2.3 Map Building

Simultaneous Localization and Mapping (SLAM) is a key problem in mobile robotics. Registering pairs of consecutive laser scans on its own can only provide estimates about the movement in between the poses where the scans have been acquired but cannot be used for building consistent maps due to inaccuracies and drift (when propagating estimated movements over registrations). Instead, pure pairwise registration algorithms are usually used in the front-end of SLAM systems to obtain a rough initial vehicle trajectory and to detect loop closures, i.e., regions where the robot has been before. For globally aligning all acquired scans and building a consistent map, the registration problem is usually formulated in terms of a graph where poses or landmark positions form the vertices, and view or movement constraints form the edges. For optimizing a graph of poses with initial estimates many different approaches have been proposed [23–26]. Our laser scans are very sparse, however, and, consequently, our estimated transformations are accurate but not as accurate as the individual laser measurements. As a second mean for compensating for the non-uniform densities in our scans, we do not use a single edge between vertices to encode their relative position but estimate point correspondences in between the scans and iteratively refine the resulting system. For each correspondence, we add an edge to the graph that follows the same error metric as our registration algorithm—again using the information extracted from approximate surface reconstruction. To optimize the resulting graph, we use g2o [7], a state-of-the-art open-source general graph optimization framework. In a final optional processing step, we build a 3D map with the optimized poses using OctoMap [14] for being able to plan paths in future missions of the MAV.

3 Approach

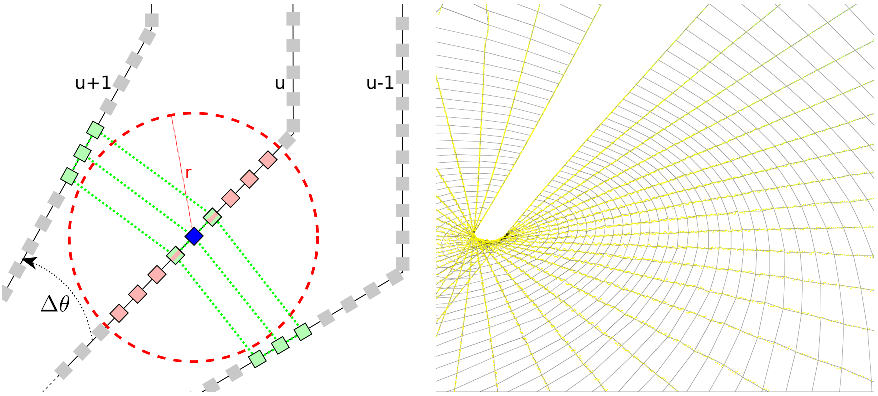
3.1 Registration

Iterative registration algorithms align pairs of 3D point clouds by alternately searching for corresponding points between the clouds and minimizing the distances between matches. A standard algorithm is the original Iterative Closest Point (ICP) algorithm [27]. In order to align a point cloud A with point cloud B, is searches for closest neighbors in B for points $\mathbf{a}_i \in A$ and minimizes the point-to-point distances $d_i^{(\mathbf{T})} = \mathbf{b}_i - \mathbf{T}\mathbf{a}_i$ of found correspondences:

$$\mathbf{T}^\star = \arg\min_{\mathbf{T}} \sum_i \| d_i^{(\mathbf{T})} \|^2. \qquad (1)$$

As a result, points in A are dragged onto their corresponding points in B. Assuming (predominantly) correct correspondences, the ICP algorithm can reliably register regular uniform density point clouds (if the initial alignment is not considerably off). In case of our non-uniform density point clouds, closest points do not correspond to the same physical point in the measured environment and the point-to-point error metric leads to dragging the high-density 2D scan lines onto another—instead of correctly aligning the aggregated 3D scans.

A particularly robust registration algorithm is Generalized-ICP [6] which generalizes over the different available error metrics (point-to-point, point-to-plane, plane-to-plane) and thus takes into account information about the underlying surface. Instead of minimizing the distances $d_i^{(\mathbf{T})}$ between corresponding points



(a) Topology visualization

(b) Example reconstruction

Fig. 2. (a) Classic neighbor searches in non-uniform density clouds may only find points in the same scan line (red), whereas a topological neighborhood (green) may better reflect the underlying surface. (b) Example of an approximate surface reconstruction: edges in the quad mesh connect neighboring points in the same scan and between neighboring scans. Points in the first and last scan of a half rotation are not connected.

$\mathbf{a}_i$ and $\mathbf{b}_i$ as in the ICP algorithm, it minimizes the more general error metric:

$$\mathbf{T}^\star = \arg\min_{\mathbf{T}} \sum_i {d_i^{(\mathbf{T})}}^T \left(C_i^B + \mathbf{T} C_i^A \mathbf{T}^T\right)^{-1} d_i^{(\mathbf{T})}. \tag{2}$$

The effect is that corresponding points are not directly dragged onto another, but the underlying surfaces represented by the covariance matrices C_i^A and C_i^B are aligned. The covariance matrices are computed so that they express the expected uncertainty along the local surface normals at the points. Consequently, the convergence of Generalized-ICP degrades with inaccurate estimates of the covariances with regular neighborhood searches as illustrated in Fig. 2a.

At the heart of our approach is the idea to approximate the underlying surface in the point clouds to compensate for the non-uniform point densities.

3.2 Approximate Surface Reconstruction

In order to get a better estimate of the underlying covariances, we perform an approximate surface reconstruction as done in our previous work [28] in the context of range image segmentation. We traverse an organized point cloud S once and build a simple quad mesh by connecting every point $\mathbf{p} = S(u, v)$ (v-th point in the u-th scan line) to its neighbors $S(u, v + 1)$, $S(u + 1, v + 1)$, and $S(u + 1, v)$ in the same and the subsequent scan line (see Fig. 2). We only add a new quad to the mesh if $S(u, v)$ and its three neighbors are valid measurements, and if all connecting edges between the points are not occluded. The first check accounts for possibly missing or invalid measurements in the organized structure. For the latter occlusion checks, we examine if one of the connecting edges falls into a common line of sight with the viewpoint $\mathbf{v} = \mathbf{0}$ from where the measurements were taken. If so, one of the underlying surfaces occludes the other and the edge is not *valid*:

$$valid = (|\cos \theta_{i,j}| \leq \cos \epsilon_\theta) \wedge \left(d_{i,j} \leq \epsilon_d^2\right), \tag{3}$$

$$\text{with} \quad \theta_{i,j} = \frac{(\mathbf{p}_i - \mathbf{v}) \cdot (\mathbf{p}_i - \mathbf{p_j})}{\|\mathbf{p}_i - \mathbf{v}\| \; \|\mathbf{p}_i - \mathbf{p_j}\|}, \tag{4}$$

$$\text{and} \qquad d_{i,j} = \|\mathbf{p}_i - \mathbf{p_j}\|^2, \tag{5}$$

where ϵ_θ and ϵ_d denote maximum angular and length tolerances, respectively. If all checks pass, we add a new quad. Otherwise, holes arise. After construction, we simplify the resulting mesh by removing all vertices that are not used in any quad. A typical result of applying our approximate surface reconstruction to a 3D scan acquired using our flying MAV is shown in Fig. 2b. For details on the approximate surface reconstruction we refer to [28].

3.3 Approximate Covariance Estimates

To estimate the covariance matrix of a point, we directly extract its local neighborhood from the topology in the mesh—instead of searching for neighbors. Depending on the desired smoothing level (usually controlled with the search radius), we can extend a point's neighborhood to include the neighbors of neighbors and ring neighborhoods farther away from the point.

Instead of computing the empirical covariances as in [6], we approximate them using the local surface normals. We compute the normal $\mathbf{n}_i$ for point $\mathbf{p}_i$ directly on the mesh as the weighted average of the plane normals of the N_T faces surrounding $\mathbf{p}_i$:

$$\mathbf{n}_i = \frac{\sum_{j=0}^{N_T} (\mathbf{p}_{j,a} - \mathbf{p}_{j,b}) \times (\mathbf{p}_{j,a} - \mathbf{p}_{j,c})}{\| \sum_{j=0}^{N_T} (\mathbf{p}_{j,a} - \mathbf{p}_{j,b}) \times (\mathbf{p}_{j,a} - \mathbf{p}_{j,c}) \|}, \tag{6}$$

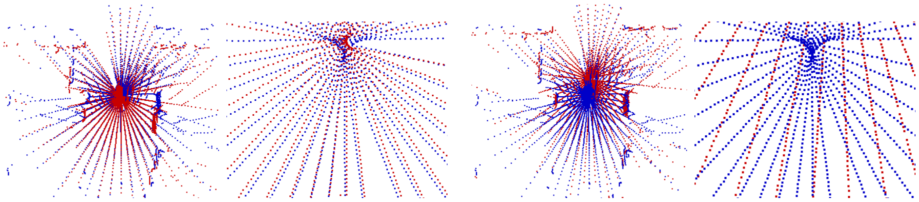
with face vertices $\mathbf{p}_{j,a}$, $\mathbf{p}_{j,b}$ and $\mathbf{p}_{j,c}$. We then compute C_i^A and C_i^B as in [6]:

$$C_i^A = \mathbf{R}_{\mathbf{n}_i}^A \left(\begin{smallmatrix} \epsilon & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{smallmatrix}\right) \mathbf{R}_{\mathbf{n}_i}^{A\;T}, \; C_i^B = \mathbf{R}_{\mathbf{n}_i}^B \left(\begin{smallmatrix} \epsilon & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{smallmatrix}\right) \mathbf{R}_{\mathbf{n}_i}^{B\;T} \tag{7}$$

with rotation matrices $\mathbf{R}_{\mathbf{n}_i}^A$ and $\mathbf{R}_{\mathbf{n}_i}^B$ so that ϵ reflects the uncertainty along the approximated normals $\mathbf{n}_i^A$ and $\mathbf{n}_i^B$.

3.4 Registration with Approximate Covariance Estimates

The actual registration does not deviate from the original Generalized-ICP algorithm or any other ICP variant. Given a source point cloud A and a target point cloud B (usually current and last acquired 3D laser scan), we search for closest points in B for all points $a \in A$. Each match contributes its error (Eq. 2) to the non-linear optimization problem. For finding the optimal transformation



(a) Generalized-ICP (top and detail view)

(b) Ours (top and detail view)

Fig. 3. Registering non-uniform density point clouds. (a) Generalized-ICP suffers from inaccurate covariance estimates and incorrectly aligns the two point cloud origins and the individual scan lines. (b) Our approach correctly aligns the two point clouds.

minimizing Eq. 2, we use the Generalized-ICP implementation in PCL[1] which is based on the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm.

A typical example of registering non-uniform density point clouds using both the original Generalized-ICP and our variant with approximate covariance estimates is shown in Fig. 3. The low angular resolution in these point clouds affects the convergence of the original Generalized-ICP. In effect, it aligns the individuals scan lines and not the sensed environmental structures. Using our approach accurately aligns the two 3D scans.

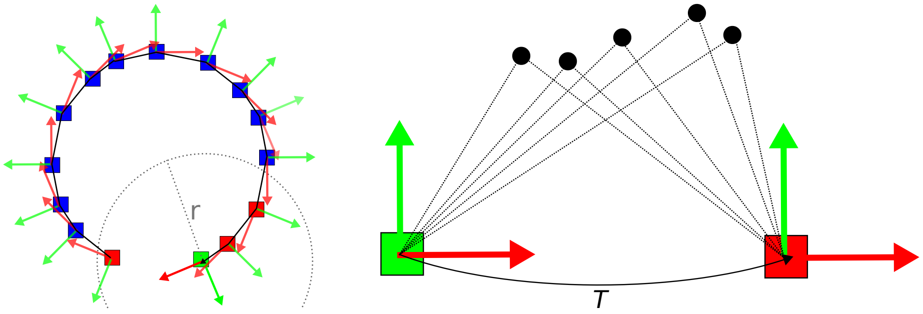
For further details on the registration and an evaluation of its convergence with respect to different angular resolutions and initial conditions, we refer [29].

3.5 Building a pose graph

To get an initial estimate of the MAV's movements, we register pairs of consecutive 3D scans. The resulting trajectory is usually locally accurate and smooth but globally not consistent, e.g., drifts which lead to inconsistencies when returning to a previously visited place (loop closures). In order to obtain a globally consistent vehicle trajectory and map, we build a graph of poses representing the MAV's initial trajectory after registration. For each pose ${}^{\{W\}}T_i$ (represented by the transformation T into the global world coordinate frame W) and 3D scan S_i, we add a vertex v_i to the graph $g = (V, E)$. Referring to Fig. 4, we determine for each vertex $v_i \in V$ the set of neighboring vertices $v_j \in V$ withing search radius r. For every match, we add edges to the graph which constrain the relative poses of the vertex to its neighbors.

A standard procedure for optimizing the trajectory would be to insert a single edge in between all neighboring vertices that encodes the relative pose. Instead, we estimate point correspondences between the scans. For every pair of neighboring vertices (v_i, v_j) we search for point correspondences between the corresponding 3D scans S_i and S_j. The central idea behind this decision is twofold: (1) we gain a second mean for compensating for the non-uniform densities in our scans and (2) using point correspondences as edges allows for iteratively optimizing the graph and re-estimating the updated correspondences.

[1] Point Cloud Library PCL: `http://pointclouds.org`



(a) Graph structure

(b) Connecting edges (one per correspondence)

Fig. 4. Graph construction. (a) For each pose, we add a vertex to the graph. We connect a vertex (green) to all neighboring vertices (red) within search radius r. (b) Instead of adding a single edge (solid line) encoding transformation T between two vertices, we add an edge (dotted line) for every point correspondence between the two scans.

3.6 Pose Graph Optimization

For each point correspondence, we add an edge to the graph that follows the same error metric as our registration algorithm again using the information extracted from approximate surface reconstruction. For modeling and optimizing the graph, we use g2o [7]. Each edge in the graph encodes two entities: a local contribution to the measurement error e and an information matrix H which represents the uncertainty of the measurement error. The information matrix is defined as the inverse of the covariance matrix, i.e., it is symmetric and positive semi-definite. Assuming that we already computed local surface normals and approximate covariance estimates as in Equations 6 and 7, the idea is to use the same error metric as in the pairwise registration (Eq. 2). As a straightforward error measurement between, respectively, two vertices A and B and the i-th correspondence pair $(\mathbf{a}_i, \mathbf{b}_i)$, we use the point-to-point error and approximate its information matrix using the error metric of our registration algorithm:

$$e_{AB,i}(\mathbf{T}) = \mathbf{b}_i - \mathbf{T}\mathbf{a}_i, \tag{8}$$

$$H_{AB,i}(\mathbf{T}) = \left(C_i^B + \mathbf{T}C_i^A\mathbf{T}^T\right)^{-1}. \tag{9}$$

The effect is that every edge contributes its approximate surface-to-surface error term to the system's information matrix—thus automatically giving lower influence on incompatible or false correspondences and quickly leading to alignment even for the sparse non-uniform density point clouds.

For the actual optimization, we follow an iterative procedure by (1) estimating correspondence pairs for all (or a subset of) points $\mathbf{a} \in A$ in B for every two vertices that are to be connected and (2) optimizing the resulting linearized system for five inner iterations. We repeat these two steps for five outer iterations. For a fast initial coarse alignment in early and an accurate refinement in later outer iterations, we use a linearly decaying distance threshold between correspondence pairs, starting with the distance between the vertices.

4 Experiments and Results

In order to assess the robustness and reliability of our approach to registration and graph optimization for allocentric 3D map building, we have recorded different datasets with the continuously spinning laser scanner on both the flying MAV as well as a ground vehicle standing still during scan acquisition.

4.1 Accuracy and Reliability

To assess accuracy and reliability, we have created a dataset of organized point clouds containing ground truth pose information. It was recorded using the same spinning laser scanner but on a mobile ground robot standing still while acquiring 3D scans—thus avoiding inaccuracies in 2D scan plane aggregation. The dataset contains point clouds from eight different poses with a total of 6890 2D laser scans acquired over multiple full rotations at each pose to obtain comparably dense point clouds. The total trajectory length between the eight poses is roughly 50 m. It was recorded by Schadler et al. [30] in the arena of the DLR SpaceBot Cup[2] competition for semi-autonomous exploration and mobile manipulation in rough terrain. For the dataset, we collected all 2D scan lines acquired at each of the poses, sorted them by rotation angle and re-organized the data to obtain eight full resolution (roughly 0.3° angular resolution) organized point clouds. We annotated each point cloud with the ground truth pose estimate obtained from an accurate multi-resolution surfel mapping approach for dense point clouds [30]. For the experimental evaluation, we generated thinned out versions of these eight original point clouds with different angular resolutions (1° to 45°) and added different amounts of noise to the ground truth pose estimates.

For evaluating the accuracy of our pose estimates and the resulting map, we use the error metrics proposed in [31]. In particular, we inspect the relative deviations $\mathbf{E}_i(\varDelta) := \left(\mathbf{Q}_i^{-1}\mathbf{Q}_{i+\varDelta}\right)^{-1}\left(\mathbf{P}_i^{-1}\mathbf{P}_{i+\varDelta}\right)$ between ground truth poses $\mathbf{Q}$ and estimated poses $\mathbf{P}$ at times i and $i+\varDelta$. From the local deviations, we can estimate the accuracy of relative pose estimates (RPE) by computing the root mean square error over all relative deviations and taking the mean over all time intervals $\varDelta$:

$$\mathrm{RPE}\left(\mathbf{E}_{1:n}\right) := \frac{1}{n}\sum_{\varDelta=1}^{n}\left(\frac{1}{m}\sum_{i=1}^{m}\|trans\left(\mathbf{E}_i\left(\varDelta\right)\right)\|^2\right)^{1/2}. \tag{10}$$

This metric averaging over all possible time intervals $\varDelta$ leverages both the errors made in relative pose estimates ($\varDelta = 1$) and global consistency errors, e.g., between start and end pose ($\varDelta = n$). In addition, we use the absolute trajectory error (ATE) which focuses on global consistency by aligning and directly

[2] DLR SpaceBot Cup: `http://www.dlr.de/dlr/en/desktopdefault.aspx/tabid-10212/332_read-8688/year-all/#gallery/12905`

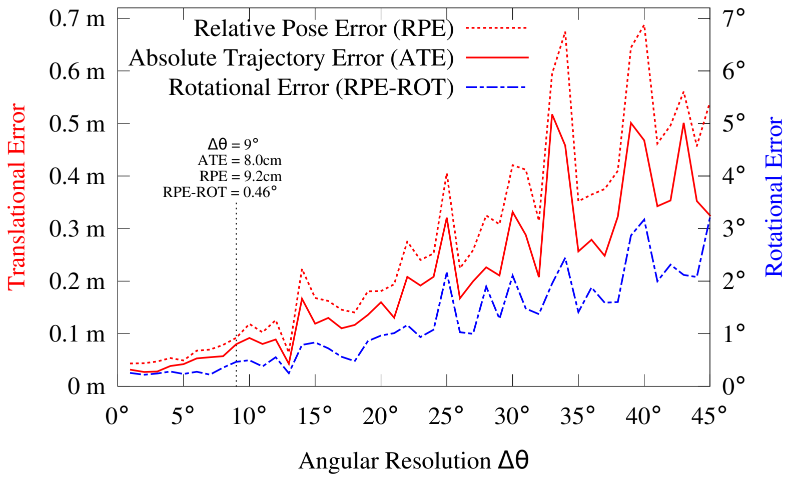


Fig. 5. Evolution of translation and rotation errors with decreasing $\varDelta\theta$.

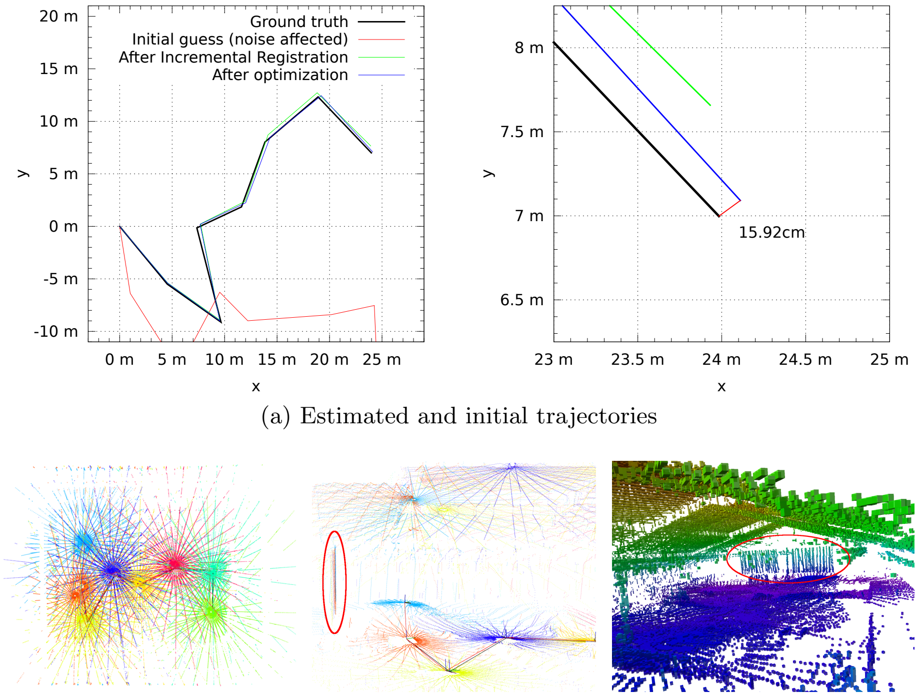
comparing absolute pose estimates (and trajectories):

$$\mathrm{ATE}\left(\mathbf{F}_{i:n}\right) := \left(\frac{1}{m}\sum_{i=1}^{m}\|trans\left(\mathbf{F}_i\left(\varDelta\right)\right)\|^2\right)^{1/2} \tag{11}$$

with $\mathbf{F}_i(\varDelta) := \mathbf{Q}_i^{-1}\mathbf{S}\mathbf{P}_i$, where $\mathbf{S}$ is the rigid-body transformation mapping the estimated trajectory $\mathbf{P}_{i:n}$ to the ground truth trajectory $\mathbf{Q}_{i:n}$.

Fig. 5 shows the estimated relative pose errors and absolute trajectory errors for different angular resolutions $1/\varDelta\theta$. For all experiments, we used a (fixed) initial trajectory estimate obtained from ground truth with added noise (rotations of $\pi/8$ about a random axes, and random translations of up to 1 m along all axes). The estimated errors reflect the expected results of increasing errors with decreasing angular resolution but also show that the obtained pose estimates and trajectories—and thus the built 3D maps—are still quite accurate, even at resolutions considerably lower than ours of 9°. In addition to the metric for translational errors, we have inspected the rotational errors in relative pose estimates (RPE-ROT). It resembles Eq. 10 but uses the rotation angle $angle\left(\mathbf{E}_i\left(\varDelta\right)\right)^2$ (axis-angle representation) instead of the translational error $\|trans\left(\mathbf{E}_i\left(\varDelta\right)\right)\|^2$. For very sparse point clouds $\varDelta\theta \geq 20°$, the translational errors considerably increase while the rotational errors stay comparably small. Overall, we obtain accurate results for $\varDelta\theta = 9°$ for both relative pose estimates (in both translation and rotation) and the overall trajectory.

A typical example for aligned point clouds, estimated trajectories and built maps for $\varDelta\theta = 9°$ is shown in Fig. 6. The initial trajectory estimate is off by several meters. While the trajectory after registration is already quite close to



(a) Estimated and initial trajectories

(b) Scans (top view)

(c) Scans (detail side view)

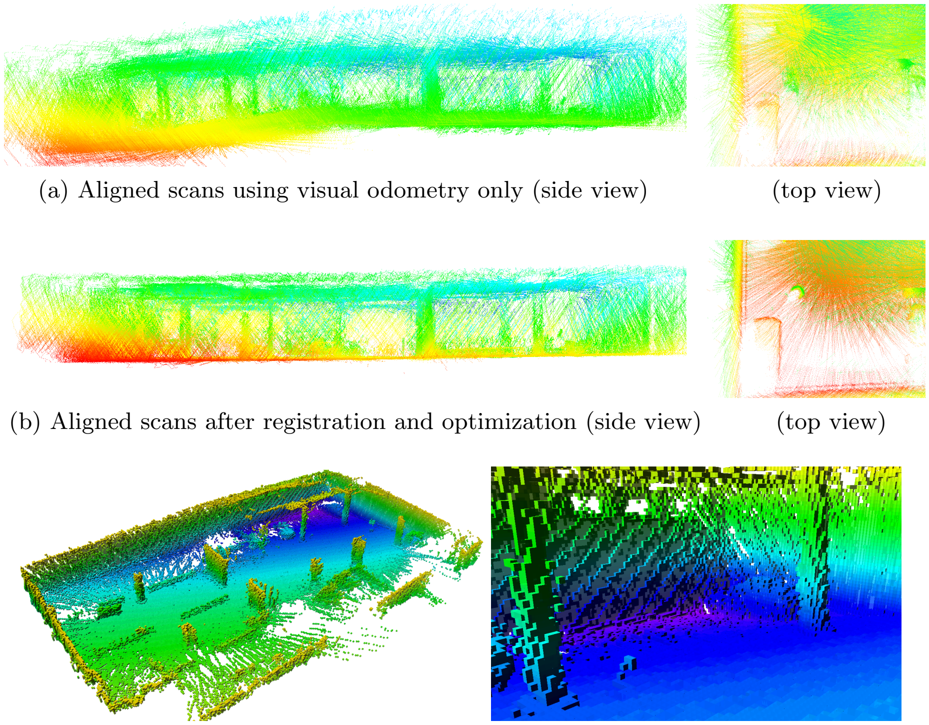
(d) Map (detail view)

Fig. 6. Examples of estimated trajectories, aligned scans and built maps at $\Delta\theta = 9°$. Pairwise registration already comes close to ground truth. The accuracy of the alignment is visible in details such as the planarity of a projector screen in (c) and (d).

ground truth, it still deviates by roughly 1 m. Differences between optimized and ground truth trajectory are barely visible in both robot poses and final map.

4.2 Mapping with the Flying MAV

As a proof-of-concept, we have recorded a dataset with the continuously spinning laser scanner on an MAV which was flying through a parking garage of 40 m × 15 m. The dataset contains a total of 4420 2D scan lines which are aggregated to 200 3D scans (each aggregated over one half rotation of the scanner). The overall trajectory length is 73 m (traveled in 100 s). The measurements cover the complete parking garage and allow for creating a complete model including pillars and other environmental structures as well as parked cars. We used two fish-eye stereo camera pairs on the MAV and visual odometry [32] to obtain an initial pose estimate and to aggregate the individual 2D scan planes to 3D scans. As can be seen in Fig. 7, the visual odometry estimate drifts—leading to an inconsistent map when used on its own, but is accurate enough for scan aggregation, i.e., estimating the movement of the MAV during one half rotation of the spinning laser scanner (500 ms).



(a) Aligned scans using visual odometry only (side view)

(top view)

(b) Aligned scans after registration and optimization (side view)

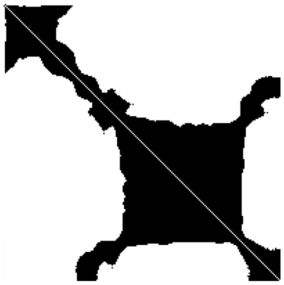
(top view)

(c) Views on the built 3D map (ceiling removed) with pillar and car silhouette

Fig. 7. Scans acquired in the parking garage with the flying MAV and aligned using (a) visual odometry only, and (b) after registration and graph optimization (points colored by height). Using visual odometry only leads to a drift while our approach provides a consistent and accurate map, as can be seen in the detail views of a corner with a cylindrical pillar and the silhouette of a car. Note that axes (and color coding) are not aligned to environmental structures but reflect the orientation of the flying MAV.

In order to obtain a consistent and accurate 3D map out of the acquired data, we register all pairs of consecutive scans (using visual odometry as an initial estimate), and then create a graph where each of the 200 pose vertices is connected to all neighboring vertices within a search radius of 3 m. We iteratively refine the whole graph over five iterations where, in each iteration, the correspondences between connected scans are re-estimated.

Overall, our approach takes roughly 45 s to build a consistent accurate 3D map, including pre-processing, registration, and graph optimization (approx. 6k connections with a total of roughly 580k edges). Fig. 7 shows the aligned scans before (visual odometry only) and after our alignment in a side view to visualize the removal of drift, and in a top view on environmental details (a cylindrical pillar and the silhouette of a car) that show the accuracy in the final model. Using the mean map entropy [33] as a measure of map quality, we obtain an entropy of -2.30 for visual odometry, whereas our approach achieves a lower entropy



(a) Adjacency matrix

Preprocessing (200 frames)	
Approx. surface reconstruction	≈1 ms
Normal and covariance estimation	<1 ms
Registration (199 consecutive pairs)	
GICP w. approx. covariances	60±17 ms
Graph optimization (5 loops)	
Estimating correspondences	1.1±0.21 s
Optimization	2.5±0.47 s

(b) Runtimes

Fig. 8. Optimization of the complete pose graph of the experiment run. Shown are (a) the 200 × 200 adjacency matrix (black = connected) of pose vertices and (b) the average runtimes of the individual processing steps.

(sharper map) of -3.58. An adjacency matrix representing the connected graph in the 5th optimization loop is shown in Fig. 8a. We report runtimes of the individual processing steps in Fig. 8b.

5 Conclusions and Future Work

In this paper, we have addressed the problem of 3D mapping with light-weight laser scanners on micro aerial vehicles (in GPS denied environments). When rotated fast, e.g., when the scanner is also used for collision avoidance, the acquired measurements have different densities in and between scan planes. To compensate for the resulting effects in feature estimation and registration, we have presented a pipeline for 3D map building, including a pairwise registration algorithm and a global alignment approach. Both make use of approximate surface reconstruction in order to extract topological information and to obtain estimates of the underlying surface.

In a real-world experiment, we have shown that we can build accurate and globally consistent 3D maps out of sparse point clouds (angular resolution of roughly 9°) with a flying micro aerial vehicle. In addition, we have evaluated the accuracy of our approach on a dataset with different angular resolutions. The experiments have shown that our method produces accurate results for the used angular resolution of our scanner, and that the translation and rotation errors remain reasonable even with particularly low angular resolutions and sparse point clouds. We have made all datasets that we used in the experiments publicly available at: `http://www.ais.uni-bonn.de/mav_mapping`.

References

1. Tomić, T., Schmid, K., Lutz, P., Domel, A., Kassecker, M., Mair, E., Grixa, I., Ruess, F., Suppa, M., Burschka, D.: Toward a fully autonomous UAV: Research

platform for indoor and outdoor urban search and rescue. IEEE Robotics Automation Magazine **19**(3) (2012) 46–56

2. Grzonka, S., Grisetti, G., Burgard, W.: Towards a navigation system for autonomous indoor flying. In: Proc. of the IEEE Int. Conference on Robotics and Automation (ICRA). (2009) 2878–2883
3. Bachrach, A., He, R., Roy, N.: Autonomous flight in unstructured and unknown indoor environments. In: Proc. of the European Micro Aerial Vehicle Conference (EMAV). (2009) 1–8
4. Shen, S., Michael, N., Kumar, V.: Autonomous multi-floor indoor navigation with a computationally constrained micro aerial vehicle. In: Proc. of the IEEE Int. Conference on Robotics and Automation (ICRA). (2011) 2968–2969
5. Nieuwenhuisen, M., Droeschel, D., Schneider, J., Holz, D., Läbe, T., Behnke, S.: Multimodal obstacle detection and collision avoidance for micro aerial vehicles. In: Proc. of the European Conference on Mobile Robots (ECMR). (2013)
6. Segal, A., Hähnel, D., Thrun, S.: Generalized-ICP. In: Proc. of Robotics: Science and Systems. (2009)
7. Kümmerle, R., Grisetti, G., Strasdat, H., Konolige, K., Burgard, W.: G2o: A general framework for graph optimization. In: Proc. of the IEEE Int. Conference on Robotics and Automation (ICRA). (2011) 3607–3613
8. Scaramuzza, D., Achtelik, M., Doitsidis, L., Fraundorfer, F., Kosmatopoulos, E., Martinelli, A., Achtelik, M., Chli, M., Chatzichristofis, S., Kneip, L., et al.: Vision-controlled micro flying robots: from system design to autonomous navigation and mapping in GPS-denied environments. IEEE Robotics and Automation Magazine (2014)
9. Montemerlo, M., Becker, J., Bhat, S., Dahlkamp, H., Dolgov, D., Ettinger, S., Haehnel, D., Hilden, T., Hoffmann, G., Huhnke, B., et al.: Junior: The stanford entry in the urban challenge. Journal of Field Robotics **25**(9) (2008) 569–597
10. Grzonka, S., Grisetti, G., Burgard, W.: A fully autonomous indoor quadrotor. IEEE Transactions on Robotics **28**(1) (2012) 90–100
11. Huh, S., Shim, D., Kim, J.: 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). (2013) 3158–3163
12. Fossel, J., Hennes, D., Claes, D., Alers, S., Tuyls, K.: Octoslam: A 3d mapping approach to situational awareness of unmanned aerial vehicles. In: Proc. of the Int. Conference on Unmanned Aircraft Systems (ICUAS). (2013) 179–188
13. Kohlbrecher, S., Meyer, J., von Stryk, O., Klingauf, U.: 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). (2011)
14. Hornung, A., Wurm, K.M., Bennewitz, M., Stachniss, C., Burgard, W.: OctoMap: An efficient probabilistic 3D mapping framework based on octrees. Autonomous Robots (2013) Software available at `http://octomap.github.com`.
15. Morris, W., Dryanovski, I., Xiao, J., Member, S.: 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. (2010)
16. Scherer, S., Rehder, J., Achar, S., Cover, H., Chambers, A.D., Nuske, S.T., Singh, S.: River mapping from a flying robot: state estimation, river detection, and obstacle mapping. Autonomous Robots **32**(5) (2012) 1–26
17. Cover, H., Choudhury, S., Scherer, S., Singh, S.: Sparse tangential network (SPARTAN): Motion planning for micro aerial vehicles. In: Proc. of the IEEE Int. Conference on Robotics and Automation (ICRA). (2013) 2820–2825

18. Nuechter, A., Lingemann, K., Hertzberg, J., Surmann, H.: 6D SLAM with approximate data association. In: Proc. of the IEEE Int. Conference on Robotics and Automation (ICRA). (2005) 242–249
19. Magnusson, M., Duckett, T., Lilienthal, A.J.: Scan registration for autonomous mining vehicles using 3D-NDT. Journal of Field Robotics **24**(10) (2007) 803–827
20. Besl, P.J., McKay, N.D.: A method for registration of 3-D shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **14**(2) (1992) 239–256
21. Droeschel, D., Stückler, J., Behnke, S.: Local multi-resolution surfel grids for MAV motion estimation and 3D mapping. In: Proc. of the Int. Conference on Intelligent Autonomous Systems (IAS). (2014)
22. Bosse, M., Zlot, R., Flick, P.: Zebedee: Design of a spring-mounted 3-d range sensor with application to mobile mapping. IEEE Transactions on Robotics **28**(5) (2012) 1104–1119
23. Frese, U., Larsson, P., Duckett, T.: A multilevel relaxation algorithm for simultaneous localization and mapping. IEEE Transactions on Robotics **21**(2) (2005) 196–207
24. Olson, E., Leonard, J., Teller, S.: Fast iterative optimization of pose graphs with poor initial estimates. In: Proc. of the IEEE Int. Conference on Robotics and Automation (ICRA). (2006)
25. Grisetti, G., Kuemmerle, R., Stachniss, C., Frese, U., Hertzberg, C.: Hierarchical optimization on manifolds for online 2d and 3d mapping. In: Proc. of the IEEE Int. Conference on Robotics and Automation (ICRA). (2010)
26. Grisetti, G., Stachniss, C., Burgard, W.: Nonlinear constraint network optimization for efficient map learning. IEEE Transactions on Intelligent Transportation Systems **10**(3) (2009) 428–439
27. Besl, P.J., McKay, N.D.: A Method for Registration of 3-D Shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence **14**(2) (1992) 239–256
28. Holz, D., Behnke, S.: Fast range image segmentation and smoothing using approximate surface reconstruction and region growing. In: Proc. of the Int. Conference on Intelligent Autonomous Systems (IAS). (2012)
29. Holz, D., Behnke, S.: Registration of non-uniform density 3D point clouds using approximate surface reconstruction. In: Proc. of the 45th Int. Symposium on Robotics (ISR) and 8th German Conference on Robotics (ROBOTIK). (2014)
30. Schadler, M., Stückler, J., Behnke, S.: Rough terrain 3D mapping and navigation using a continuously rotating 2D laser scanner. KI - Künstliche Intelligenz (2014) Accepted. `http://dx.doi.org/10.1007/s13218-014-0301-8`.
31. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of rgb-d slam systems. In: Proc. of the IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS). (2012)
32. Schneider, J., Läbe, T., Förstner, W.: 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. (2013)
33. Droeschel, D., Stückler, J., Behnke, S.: 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). (2014)