Robotics and Autonomous Systems, vol. 74, pp. 318-330, Elsevier, December 2015.

Registration of Non-Uniform Density 3D Laser Scans for Mapping with Micro Aerial Vehicles

Dirk Holz and Sven Behnke

University of Bonn, Autonomous Intelligent Systems Group
Friedrich-Ebert-Allee 144, 53113 Bonn, Germany

Abstract

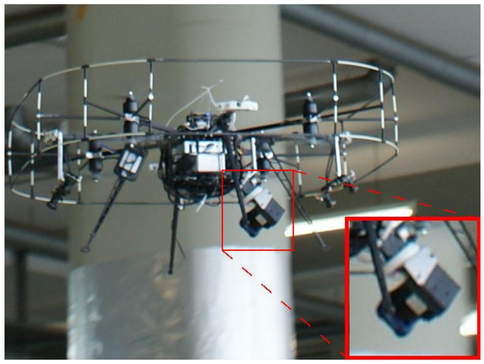
Micro aerial vehicles (MAVs) pose specific constraints on onboard sensing, mainly limited payload and limited processing power. For accurate 3D mapping even in GPS-denied environments, we have designed a lightweight 3D laser scanner specifically for the application on MAVs. Similar to other custom-built 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 such non-uniform density 3D point clouds 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.

Keywords: Mapping, registration, micro aerial vehicles, approximate surface reconstruction, Generalized-ICP

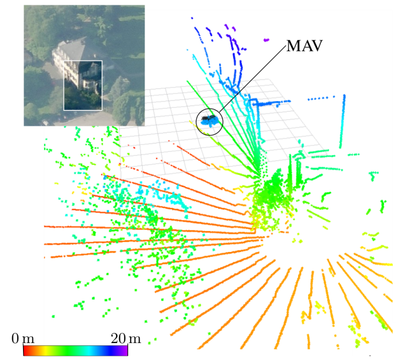
1. Introduction

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

Email address: `{holz, behnke}@ais.uni-bonn.de` (Dirk Holz and Sven Behnke)



(a) Flying micro aerial vehicle with scanner



(b) Example of a scan acquired in-flight

Figure 1: The laser scanner is mounted slightly below the MAV facing forwards. Continuously rotating it allows 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 lightweight sensors provide valuable information, they suffer from a limited field-of-view and cameras are sensitive to illumination conditions. Only few MAVs [1, 2, 3, 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.

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, we have developed a lightweight 3D scanner [5] specifically suited for the application on MAVs. It consists of a Hokuyo 2D laser range scanner, a rotary actuator and a slip ring to allow continuous rotation. Just as with other rotated scanners, the acquired point clouds (aggregated over a half rotation of the scanner) 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 Figure 1). Since we use the laser scanner for omnidirectional obstacle detection and collision avoidance, we rotate it quickly with 1 Hz, resulting in a particularly low angular resolution of roughly 9° to 10°. 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 by means of simulatenous localization and mapping (SLAM).

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 (GICP) [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.

This paper is an extended version of our previous works on registration and mapping with such sparse 3D laser scans [8, 9]. It is organized as follows. After a discussion of related work in Section 2, we present our registration approach including the approximate surface reconstruction and the approximate feature estimation in Section 3. In Section 4, we discuss the extension to a complete SLAM system: we extend the registration approach to also estimate the pose uncertainty and use this information for graph-based SLAM (single edge between connected nodes) as a baseline system. We then introduce our SLAM approach using multiple edges per connection where every edge encodes a point-to-point correspondence (in terms of the GICP error metric). In Section 5, we present the results of a thorough experimental evaluation of both the plain registration approach and the two SLAM variants. Finally, we summarize the main conclusions and discuss future work in Section 6.

2. Related Work

Particularly important for the autonomous application of MAVs is the ability to perceive and avoid obstacles. Building environment maps is necessary for goal-directed navigation planning and executing the planned trajectories. In the following, we discuss related works with a focus on 1) perception, 2) registration and 3) mapping. The former two allow sensing environmental structures, keeping track of the motion of the MAV, and aggregating measurements in local egocentric maps in order to be able to reliably avoid collisions. The latter aims for building allocentric 3D environments for being able to plan paths and missions.

2.1. Perception and Mapping with Micro Aerial Vehicles

Scaramuzza et al. [10] 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 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 the detection of obstacles in every direction [11, 12]. Up to now, such 3D laser scanners are rarely used on lightweight MAVs—due to payload limitations. Instead, two-dimensional laser range finders are often used [1, 2, 3, 4, 13, 14]. 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. [15], for example, use Hector SLAM [16] for registering horizontal 2D laser scans and OctoMap [17] to build a three-dimensional occupancy model of the environment at the measured heights.

Morris et al. [18] 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 rotating laser range finder that does not only allow capturing 3D measurements without moving, but also provides omnidirectional sensing at comparably high frame rates (2 Hz in our setup by aggregating scans over one half rotation).

A similar sensor is described by Scherer et al. and Cover et al. [19, 20]. 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.

For building maps with a hand-held rotating 2D laser range finder, Zhang et al. [21] compute edge points and planar points in the acquired range scans. They split the SLAM task in two problems: matching range scans to obtain motion estimates at a high frame rate and accurate registration for mapping at a lower frame rate. The method produces accurate 3D maps of smaller environments but does not detect loop closures, i.e., entering previously mapped regions.

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], Nüchter et al. [22], and Magnusson et al. [23].

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 [24]. 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, however, 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 reconstructions 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 [25] within the same project on MAV-based mapping as the work at hand. Both ways constitute problems in their own right.

Bosse et al. [26] 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 induce oscillation. For registration, Bosse et al. use a surfel-based approach and efficiently solve both aggregating point clouds and building globally consistent 3D maps. Since surfels are computed on local neighborhoods, the approach may suffer from the same degradation effects as GICP when applied to the non-uniform density data of our sensor setup.

2.3. Multi-View Scan Registration and SLAM

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. A standard approach is to encode relative pose estimates between connected view poses (vertices) in a single edge, e.g., a homogeneous transformation matrix, together with a covariance estimate. We present such a system making use of the proposed registration algorithm as a baseline system for comparison in Section 4.2. For optimizing a graph of poses with initial estimates many different approaches have been proposed [27, 28, 29, 30]. For a survey on different mapping problems and their relation to graph-based SLAM, we refer the interested reader to the survey of Agarwal et al. [31]. The difficulty in our case is that our laser scans are particularly sparse. Consequently, our estimated transformations are accurate but not as accurate as each individual laser measurement (see the results of our experimental evaluation of pairwise registration in Section 5.1).

As a second mean for compensating for the non-uniform densities in our scans, we do not use a single edge between 3D scans 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 graph optimization framework. In a final optional processing step, we build a 3D map with the optimized poses using OctoMap [17] for being able to plan paths in future missions of the MAV.

In multi-view scan matching, multiple poses from which scans have been taken are determined simultaneously by aligning all scans in a single error function or optimization framework. In the 2D domain, a popular multi-view scan registration approach is the algorithm proposed by Lu and Milios (LUM) [32]. Borrmann et al. extend this approach to six degrees of freedom for the alignment of 3D scans and present methods to efficiently deal with the resulting nonlinearities [33]. Several further extensions and optimizations have been proposed by the same and other authors since then. The resulting SLAM approach first applies the ICP algorithm to align consecutive point clouds and then builds a graph based on the determined connectivity of view poses similar to our approach. Both the determined relative transformations between view poses and the sets of point correspondences are represented in the edges. From both transformation and correspondences a measurement vector and its covariance matrix are computed which are then fed as one block into a large linear system. The linear system is then solved for the optimal relative transformations and view poses. In contrast, in the proposed multi-edge approach, every correspondence pair forms a block in the final non-linear error function. Its simplification is thereby left to g2o, e.g., using sparse Cholesky decomposition. Furthermore, LUM uses a point-to-point error metric as in the original ICP algorithm which conflicts with the particularly sparse nature of our point clouds. Instead, we approximate the underlying surface and use a probabilistic surface-to-surface error metric in both the initial pairwise registration and the point correspondence edges of the graph.

Using multiple edges constraining the relative transformation between two view poses also forms the underlying idea of landmark-based SLAM and bundle adjustment. In landmark-based SLAM, features are extracted from the data acquired by the moving sensor and used as landmarks in the graph. In order to form constraints in the graph of poses and landmarks, the extracted features are matched. Matching is usually performed in a higher-dimensional descriptor space to ease the involved data association problem. Prominent examples include using 3D features such as FPFH by Rusu et al. [34] or appearance-based features such as SIFT, SURF or ORB as in the evaluation of Endres et al. [35]. Repeatable features are not easily extractable from our 3D laser scans, especially since the different scans are likely to not include the same parts of environmental structures due to the low angular resolution between individual scan lines. Instead, our matching is purely based on proximity of the raw points. Due to the low angular resolution, it is very likely that none of the matching pairs is formed by two measurements of the same point, but by measureing two points in close vicinity, possibly on different surfaces. By using a robust surface-to-surface error metric, we compensate for this inaccuracy. To compensate for false correspondences in the initial matching steps, we iteratively refine both transformation and matches in the initial registration and the global alignment with a decreasing distance threshold in an ICP-like fashion. Hence, our approach can be categorized as being somewhere between multi-view scan matching and landmark-based SLAM.

Similar to our multi-edge global alignment step is the approach of Ruhnke et al. [36]. They also use raw point matches as constraints in the graph and apply a surfel-based error metric to iteratively refine both the sensor poses and the positions of the points. Their approach can build highly accurate object models but requires a rough initial alignment of the dense RGB-D data. In contrast, we present a complete pipeline that is tailored for the challenging non-uniform density point clouds instead of dense RGB-D data and that can cope with unavailable and erroneous initial pose estimates. Both the initial registration and the multi-edge global alignment make use of approximate surface reconstructions and the same surface-to-surface error metric.

Recently, Zlot and Bosse [37] presented a 3D mapping system for mines that uses a continuously spinning SICK scanner. They use non-rigid surfel registration and graph optimization for aggregating point clouds and building consistent maps. Compared to our work, their scanner is rotated slower, equipped with an accurate inertial measurement unit, and mounted on a slowly driving truck. Moreover, the aggregation is performed in larger local windows to increase the density of the data in a similar fashion as Droeschel et al. who build local egocentric maps [25]. Instead, we address the problem of registration and mapping directly using the sparse non-uniform density point clouds.

3. Registration of Sparse 3D Laser Scans

Under the assumption of good motion estimates (e.g., GPS, visual odometry, or inertial measurement units) at least over short periods of time, acquired range scans can be aggregated to form locally consistent 3D point clouds. Throughout this paper, we will assume such an estimate as given (robust visual odometry) and process point clouds aggregated over one half rotation of the laser range scanner. For details on scan aggregation, initial motion estimate and sensor characteristics, we refer to [5].

In contrast to the motion estimate being reliable over short periods of time for aggregating point clouds, we do not assume a good pose estimate between aggregated point clouds and good motion estimates over longer periods of time in general. Instead, we design our approach to be robust against noisy, erroneous and no initial pose estimates so as to estimate the motion of the MAV between the acquisition of aggregated point clouds. This allows mapping and localization purely based on point cloud registration even in case of sensor outages and other localization errors.

3.1. Registration of 3D Point Clouds

A point cloud is a data structure P used to represent a collection of multi-dimensional points $\boldsymbol{p} \in P$. In a 3D point cloud, the elements usually represent the X, Y, and Z geometric coordinates of an underlying sampled surface. When more information is available (such as color information) or information about local surface normal $\boldsymbol{n}$ or curvature κ, the points $\boldsymbol{p} \in P$ become n-dimensional.

Given a *source* point cloud A with points $\boldsymbol{a} \in A$, and a *target* point cloud B with points $\boldsymbol{b} \in B$, the problem of registration is to find *correspondences* between A and B, and estimate a transformation $\boldsymbol{T}$ that, when applied to A, aligns all pairs of corresponding points $(\boldsymbol{a}_i \in A, \boldsymbol{b}_j \in B)$. One fundamental problem of registration is that these correspondences are usually not known and need to be determined by the registration algorithm.

Iterative registration algorithms align pairs of 3D point clouds by alternately searching for correspondences between the clouds and minimizing the distances between matches. A standard algorithm is the Iterative Closest Point (ICP) algorithm [24]. In order to align a point cloud A with a point cloud B, it searches for closest neighbors in B for points $\boldsymbol{a}_i \in A$ and minimizes the point-to-point distances $\boldsymbol{d}_{ij}^{(\boldsymbol{T})}=\boldsymbol{b}_j-\boldsymbol{T}\boldsymbol{a}_i$ of the set of found correspondences C in order to find the optimal transformation $\boldsymbol{T}^\star$:

$$\boldsymbol{T}^\star = \arg\min_{\boldsymbol{T}} \sum_{(ij)\in C} \|\boldsymbol{d}_{ij}^{(\boldsymbol{T})}\|^2. \tag{1}$$

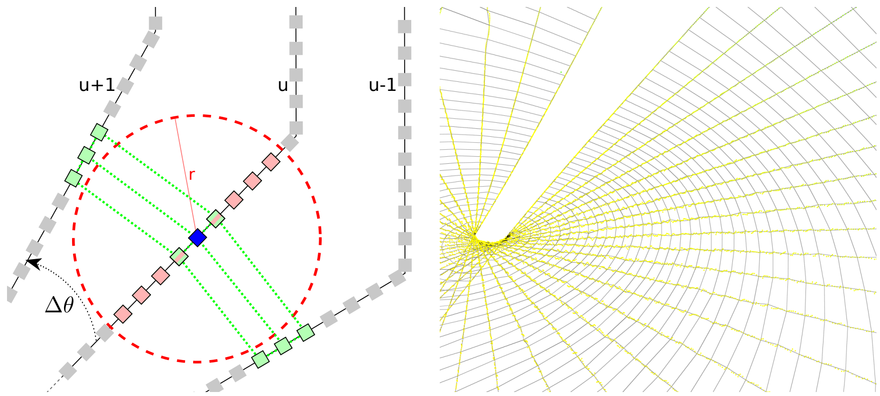
As a result, points in A are dragged onto their corresponding points in B. Assuming (predominantly) correct correspondences, the ICP algorithm can reliably register regular uniform density point clouds (if the initial alignment is not considerably off). In case of our non-uniform density point clouds,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

3.2. Generalized Iterative Point Cloud Registration

A particularly robust registration algorithm is Generalized-ICP (GICP) [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 $\boldsymbol{d}_{ij}^{(\boldsymbol{T})}$ between corresponding points $\boldsymbol{a}_i$ and $\boldsymbol{b}_j$ as in the ICP algorithm, it inspects the distribution

$$\boldsymbol{d}_{ij}^{(\boldsymbol{T})} \sim \mathcal{N}\left(\boldsymbol{b}_j - \boldsymbol{T}\boldsymbol{a}_i,\ \boldsymbol{\Sigma}_j^B + \boldsymbol{R}\boldsymbol{\Sigma}_i^A\boldsymbol{R}^T\right) \tag{2}$$

where $\boldsymbol{R}$ is the rotation matrix of $\boldsymbol{T}$. The underlying assumption is that both points in A and points in B are drawn from independent normal distributions, i.e., $\boldsymbol{a}_i \sim \mathcal{N}(\hat{\boldsymbol{a}}_i, \boldsymbol{\Sigma}_i^A)$ and $\boldsymbol{b}_j \sim \mathcal{N}(\hat{\boldsymbol{b}}_j, \boldsymbol{\Sigma}_j^B)$. The optimal transformation $\boldsymbol{T}^\star$ best



(a) Topology visualization

(b) Example reconstruction

Figure 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.

aligning A to B can be found using maximum likelihood estimation (MLE):

$$\begin{aligned} \boldsymbol{T}^{\star} &= \arg\max_{\boldsymbol{T}} \prod_{ij \in C} p\left(d_{ij}^{(\boldsymbol{T})}\right) = \arg\max_{\boldsymbol{T}} \sum_{ij \in C} \log\left(p\left(d_{ij}^{(\boldsymbol{T})}\right)\right) & (3) \\ &\simeq \arg\min_{\boldsymbol{T}} \underbrace{\sum_{ij \in C} {d_{ij}^{(\boldsymbol{T})}}^{T} \left(\boldsymbol{\Sigma}_j^B + \boldsymbol{R}\boldsymbol{\Sigma}_i^A \boldsymbol{R}^T\right)^{-1} d_{ij}^{(\boldsymbol{T})}}_{=\text{ simplified likelihood } L(\boldsymbol{T})}. & (4) \end{aligned}$$

The effect of minimizing (4) is that corresponding points are not directly dragged onto another, but the underlying surfaces represented by the local covariance matrices $\boldsymbol{\Sigma}_i^A$ and $\boldsymbol{\Sigma}_j^B$. 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 Figure 2a. If the neighborhood radius is too small, the covariance only reflects a single scan line and not the surface. If it is too large, the covariance can become inaccurate compared to the underlying surface.

At the heart of our approach is the idea to approximate the surfaces in the point clouds in order to compensate for the non-uniform point densities and to compute accurate covariances that better reflect the underlying surfaces.

3.3. 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 [38] in the context of range image segmentation. We traverse an organized point cloud

P once and build a simple quad mesh by connecting every point $\boldsymbol{p} = P(u, v)$ (v-th point in the u-th scan line) to its neighbors $P(u, v+1)$, $P(u+1, v+1)$, and $P(u+1, v)$ in the same and the subsequent scan line (see Figure 2). We only add a new quad to the mesh if $P(u, v)$ 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 $\boldsymbol{v} = \mathbf{0}$ from where the measurements were taken. If so, one of the underlying surfaces occludes the other and the edge is not *valid*:

$$\begin{aligned} valid &= \left(|\cos\theta_{i,j}| \leq \cos\epsilon_\theta\right) \wedge \left(d_{i,j} \leq \epsilon_d^2\right), && (5) \\ \text{with} \quad \theta_{i,j} &= \frac{(\boldsymbol{p}_i - \boldsymbol{v}) \cdot (\boldsymbol{p}_i - \boldsymbol{p_j})}{\|\boldsymbol{p}_i - \boldsymbol{v}\| \, \|\boldsymbol{p}_i - \boldsymbol{p_j}\|}, && (6) \\ \text{and} \quad d_{i,j} &= \|\boldsymbol{p}_i - \boldsymbol{p_j}\|^2, && (7) \end{aligned}$$

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 by our MAV is shown in Figure 2b.

For the sparse point clouds acquired by the MAV the occlusion check is, however, inaccurate since the larger angle $\Delta\theta$ between scan lines causes that occluding edges may not get scanned as in dense point clouds. That is, the scanner may sample the surfaces farther away from the occluding boundaries. Hence, it is often not possible to directly deduce an occlusion from the raw measurements. Instead, we adapt the threshold ϵ_d for the maximum edge length to capture the expected distance between neighboring points on the same surface and the expected angular resolution within and between scan lines:

$$\epsilon_d(d_i) = \begin{cases} \sqrt{2}\ d_i\ \tan\Delta\theta & \text{between scan lines, and} \\ \sqrt{2}\ d_i\ \tan\Delta\phi & \text{within scan lines.} \end{cases} \qquad (8)$$

The threshold $\epsilon_d(d_i)$ depends on the measured distance to point $\boldsymbol{p}_i$, i.e., $d_i = \|\boldsymbol{p}_i - \boldsymbol{v}\|$ and is computed for every point. For neighboring points within a scan line, we use a different threshold that corresponds to the angular resolution $\Delta\phi$ of the range scanner (and taking into account subsampling if applied).

3.4. 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 the neighborhood of a point 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 $\boldsymbol{n}_i$ for point $\boldsymbol{p}_i$

directly on the mesh as the weighted average of the plane normals of the N_T faces surrounding $\boldsymbol{p}_i$:

$$\boldsymbol{n}_i = \frac{\sum_{j=0}^{N_T}(\boldsymbol{p}_{j,a} - \boldsymbol{p}_{j,b}) \times (\boldsymbol{p}_{j,a} - \boldsymbol{p}_{j,c})}{\|\sum_{j=0}^{N_T}(\boldsymbol{p}_{j,a} - \boldsymbol{p}_{j,b}) \times (\boldsymbol{p}_{j,a} - \boldsymbol{p}_{j,c})\|}, \tag{9}$$

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

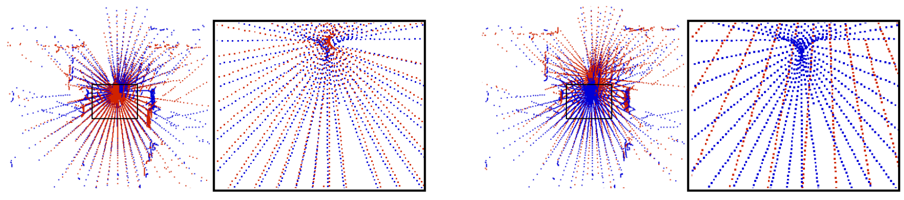
$$\boldsymbol{\Sigma}_i^A = \boldsymbol{R}_{\boldsymbol{n}_i}^A \begin{pmatrix} \epsilon & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \boldsymbol{R}_{\boldsymbol{n}_i}^{A\ T},\ \boldsymbol{\Sigma}_j^B = \boldsymbol{R}_{\boldsymbol{n}_j}^B \begin{pmatrix} \epsilon & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \boldsymbol{R}_{\boldsymbol{n}_j}^{B\ T} \tag{10}$$

with rotation matrices $\boldsymbol{R}_{\boldsymbol{n}_i}^A$ and $\boldsymbol{R}_{\boldsymbol{n}_i}^B$ so that ϵ reflects the uncertainty along the approximated normals $\boldsymbol{n}_i^A$ and $\boldsymbol{n}_i^B$. The intuition behind this is that we assume the point to lie on the approximated surface while not knowing where the point is lying on the surface. The lower the ϵ the more local planarity is assumed around the point. Consequently, with a low value ($\epsilon \leq 10^{-3}$), the registration error (4) to be minimized converges to a plane-to-plane error metric.

3.5. Registration with Approximate Covariance Estimates

The actual registration of, respectively, two point clouds and the two approximated surface meshes 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 the current and the last aggregated 3D point cloud), we first compute approximate surface reconstructions for both clouds and remove all points not contained in any polygon of the mesh. Using the surface approximations, we compute for all residual points (subsets A' and B') approximate covariance estimates using (10). In each iteration, we then search for closest points in B' for all points $\boldsymbol{a'} \in A'$. Each found correspondence pair (ij) contributes a measurement error to the non-linear optimization problem using the generalized error metric (4). For finding the optimal transformation minimizing (4), we use the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm which approximates Newton's method. The required second-order derivatives can be efficiently computed analytically due to the simple form of the simplified likelihood $L(\boldsymbol{T})$ in (4). Optimization using Newton's method with the found correspondence pairs is stopped when the algorithm converges (usually in 3 to 5 iterations in our experiments) or when the maximum number of iteration steps is reached (in our implementation 10). We inspect the computed pose change and stop the iterative alignment if the pose no longer changes. In case of changes, we apply the computed transformation and start the next iteration with new correspondence pairs.

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 Figure 3. The low angular resolution in these point clouds affects the convergence of the original Generalized-ICP. In effect, it aligns the individual scan lines and not the sensed environmental structures. Hence, it diverges even from a good initial pose estimate. Our approach accurately aligns the two 3D point clouds.



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

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

For a thorough experimental evaluation of the convergence and divergence behavior of our approach, we refer to the pairwise registration experiments in Section 5. Overall, the approximate mesh registration can robustly align sparse point clouds, but shows minimal inaccuracies in the final alignment (e.g., deviations in the range of centimeters when compared to ground truth pose estimates).

4. Mapping with Sparse 3D Laser Scans

Registration of sparse 3D point clouds (Section 3) can be used to compute the relative transformation between the view poses where two point clouds have been acquired (or aggregated in our case). Likewise, sequential pairwise scan-to-scan registration can be used to obtain an initial trajectory estimate. However, by only using the last point cloud to align a newly acquired one, even small registration errors accumulate and lead to a drift in the estimated trajectory. The resulting trajectories are usually locally accurate and smooth but globally not consistent. The drift can lead to inconsistencies in the map when returning to a previously visited place (loop closures).

4.1. Graph-Based Simultaneous Localization and Mapping

Graph-based simultaneous localization and mapping (SLAM) aims at computing globally consistent trajectories and maps by building and optimizing a pose graph in which the edges encode the spatial relation between connected poses. The graph optimization forms the back-end of the system while a front-end detects loop closures and computes relative poses to feed the back-end.

In its simplest form, pairwise registration as in Section 3 can be used as a front-end to determine an initial trajectory estimate and the vicinity of estimated poses determines the connectivity in the graph, e.g., by connecting all poses within a certain radius and having a similar orientation. Since the laser scanner of our MAV perceives the environment almost omnidirectionally, we can neglect the orientation of view poses and instead connect purely based on Euclidean distance (see Figure 4a). In the following, we will present two versions of vertex connectivity:

1. a *single-edge baseline system*: a classic version with a single edge encoding the relative transformation between two view poses and the associated covariance matrix representing the uncertainty in the relative transformation, and

2. the *proposed multi-edge system*: a graph where the connectivity between vertices is not represented using a single edge encoding a relative transformation but instead using multiple edges each encoding a point-to-point correspondence.

In both cases, the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ represents view poses $\boldsymbol{v}_i$ as a set of vertices $\mathcal{V}$ and spatial relations between two view poses $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ as edges e_{ij} in the set of edges $\mathcal{E}$. Each edge in the graph encodes two entities: a local contribution to the measurement error $\boldsymbol{e}$ and an information matrix $\boldsymbol{H}$ 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. The difference between the two systems is the type and number of edge constrains e_{ij}, i.e., the choice of $\boldsymbol{e}$ and $\boldsymbol{H}$. In both cases, however, we model and optimize the graph using the graph optimization framework g2o [7].

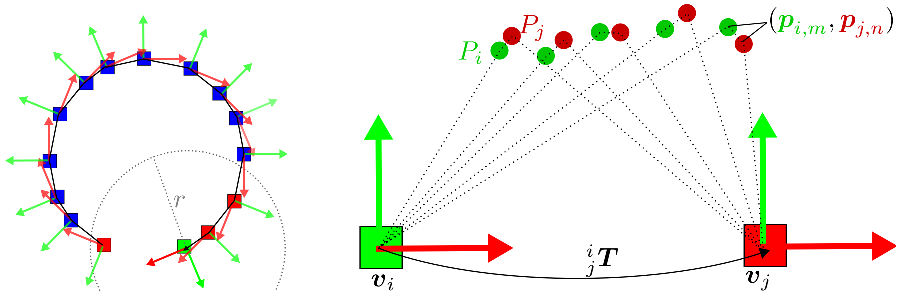
4.2. Baseline System — Single Edge Connections Encoding Pose & Uncertainty

A straightforward extension of our approximate surface registration approach to a graph-based mapping system can be achieved by first applying registration sequentially to pairs of consecutive point clouds $(P_i, P_{j=i+1})$ in order to determine both the relative transforms ${}^i_j\boldsymbol{T}$ and the initial graph connectivity. In this stage, all poses within a radius r get connected in the graph (see Figure 4a). In the second phase, we register all connected pairs $(P_i, P_{j\neq i+1})$ that have not yet been registered in the initial registration of consecutive point clouds. In both stages we collect, for every registered pair of point clouds (P_i, P_j), the estimated transformation ${}^i_j\boldsymbol{T}$ as well as the associated covariance matrix $\boldsymbol{\Sigma}_{{}^i_j\boldsymbol{T}}$ and its inverse, the information matrix $\boldsymbol{\Sigma}^{-1}_{{}^i_j\boldsymbol{T}}$.

In order to get an estimate of the relative pose uncertainty in the form of $\boldsymbol{\Sigma}_{{}^i_j\boldsymbol{T}}$, we do not only use the estimated transform ${}^i_j\boldsymbol{T}$ but also the set C of found point correspondences. We compute $\boldsymbol{\Sigma}_{{}^i_j\boldsymbol{T}}$ using the approximation by Censi [39]:

$$\boldsymbol{\Sigma}_{{}^i_j\boldsymbol{T}} \approx \left(\frac{\partial^2 L}{\partial \boldsymbol{x}^2}\right)^{-1} \frac{\partial^2 L}{\partial z \partial \boldsymbol{x}} \, \boldsymbol{\Sigma}(z) \, \frac{\partial^2 L}{\partial z \partial \boldsymbol{x}}^T \left(\frac{\partial^2 L}{\partial \boldsymbol{x}^2}\right)^{-1} \tag{11}$$

where L is the simplified likelihood function from Equation (4), z denotes the individual found correspondences C between the two point clouds P_i and P_j, and $\boldsymbol{\Sigma}(z)$ the covariance of the correspondence pairs. Here, the relative transformation between two view poses is not represented as a homogeneous transformation matrix ${}^i_j\boldsymbol{T}$, but in a parameterized form $\boldsymbol{x} = (\boldsymbol{t}, \boldsymbol{q})^T$ with translation $\boldsymbol{t}$ and rotation by the unit quaternion $\boldsymbol{q}$.



(a) Graph structure

(b) Connecting edges (one per correspondence)

Figure 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 the transformation ${}^i_j\boldsymbol{T}$ between two vertices $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ (as in the baseline approach in Section 4.2), we add an edge (dotted lines) for every point correspondence between the two point clouds P_i and P_j in the proposed multi-edge approach (Section 4.3).

After registration and covariance estimation, we add a single edge e_{ij} with

$$\text{measurement error} \quad \boldsymbol{e}_{ij}(\boldsymbol{T}) = {}^i_j\boldsymbol{T} \tag{12}$$

$$\text{and information matrix} \quad \boldsymbol{H}_{ij} = \boldsymbol{\Sigma}^{-1}_{{}^i_j\boldsymbol{T}} \tag{13}$$

to $\mathcal{G}$ as a spatial constraint between view poses $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$.

For the actual optimization, we use sparse Cholesky decomposition and Levenberg Marquardt within the g2o framework [7]. In order to compensate for loop closures not present in the initial trajectory estimate but introduced by the optimization, we re-compute the connectivity graph. In case of changes (added connections, removed connections or changed connections), we optimize the newly constructed graph again. If no such changes are detected or if a maximum number of iteration steps is exceeded (10 in our experiments), we stop optimizing the graph and compute the final map by aggregating all point clouds using the updated view poses.

4.3. Our approach — Multi-Edge Connections Encoding Point Correspondences

The acquired point clouds are quite sparse and, consequently, our estimated transformations are accurate but not as accurate as each laser measurement itself (see Section 5.1). When connecting view poses using a single edge encoding transformation and relative pose uncertainty, all the individual correspondence covariances are merged into a single estimate. The merged covariances provide adequate information for refining individual transformations when optimizing over multiple point cloud connections, but can lose the accuracy in individual point correspondences.

Instead, we do not use a single edge between vertices to encode their relative pose but connect directly using estimated point correspondences in between the point clouds (see Figure 4b). In particular, for every pair of neighboring vertices $(\boldsymbol{v}_i, \boldsymbol{v}_j)$ we search for point correspondences between the respective 3D point clouds P_i and P_j. The central idea behind this decision is three-fold: 1) we maintain local surface-to-surface alignment accuracy (over multiple point clouds), 2) we gain a second mean for compensating for the non-uniform densities in our scans and 3) using point correspondences as edges allows iteratively optimizing the graph and re-estimating the updated correspondences.

For each point correspondence, we add an edge to the graph again using the information extracted from approximate surface reconstruction. Assuming that we already computed local surface normals and approximate covariance estimates as in Equations (9) and (10), the idea is to use the same error metric in Equation (4) as in the pairwise registration. As a straightforward error measurement between, respectively, two vertices $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ and the correspondence pair $(\boldsymbol{p}_{i,m}, \boldsymbol{p}_{j,n})$, we use the point-to-point difference vector and approximate its information matrix using the error metric of our registration algorithm:

$$\begin{aligned} \text{measurement error} \quad \boldsymbol{e}_{ij,mn}({}^i_j\boldsymbol{T}) &= \boldsymbol{p}_{j,n} - {}^i_j\boldsymbol{T}\boldsymbol{p}_{i,m}, &(14)\\ \text{and information matrix} \quad \boldsymbol{H}_{ij,mn}({}^i_j\boldsymbol{T}) &= \left(\boldsymbol{\Sigma}_n^{P_j} + \boldsymbol{R}\boldsymbol{\Sigma}_m^{P_i}\boldsymbol{R}^T\right)^{-1}. &(15)\end{aligned}$$

The effect is that every edge contributes its approximate surface-to-surface error term to the system information matrix—thus automatically giving lower influence to 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 $\boldsymbol{p}_{i,m} \in P_i$ in P_j for every two vertices $(\boldsymbol{v}_i, \boldsymbol{v}_j)$ that are to be connected and 2) optimizing the resulting linearized system for a maximum of ten inner iterations. We repeat these two steps for a maximum of ten outer iterations. For a fast initial coarse alignment in early and an accurate refinement in later outer iterations, we use a linearly decreasing distance threshold between correspondence pairs, starting with double the distance between the vertices. In every outer iteration step, the graph is optimized using dense Cholesky decomposition and Levenberg Marquardt within the g2o framework [7]. For both inner and outer iterations, we stop when the system has converged. Convergence in graph optimization (inner iterations) can be detected based on the changes in both view poses and system error as well as the damping factor applied by Levenberg Marquardt. For detecting convergence in the overall graph refinement in the outer iterations, we check whether the view pose connectivity and the correspondences between connected view poses have changed. When no more changes are found and the inner optimization has converged, we stop optimizing the trajectory estimate and build the final map of the environment.

5. Experiments and Results

In order to assess the performance of our approach and the involved components, we have run a series of experiments. For making the presented results

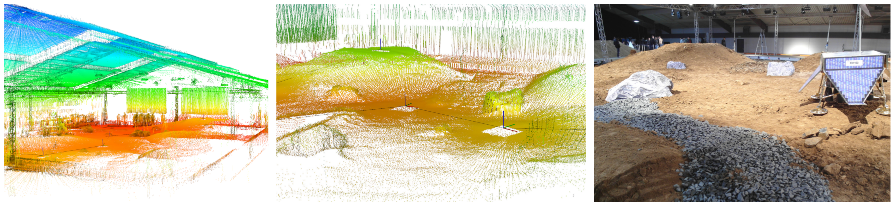


Figure 5: Environment where the dataset was recorded. Shown are snapshots of all aligned point clouds (dense, $\Delta\theta = 1^\circ$) and a photo of the arena.

both reproducible and comparable, we have recorded different datasets which we make publicly available[1].

5.1. Experiments on Pairwise Registration of Point Clouds

The first series of experiments concerns the robustness of our registration approach in terms of both the convergence and the divergence behavior under different resolutions (angles between individual scan lines) and under different initial conditions (e.g., noise in the initial pose estimates).

Registration problems considerably vary depending on the availability and quality of initial pose estimates. Assuming an optimal (ground truth) pose estimate, the point clouds are already aligned and a correct registration result is equal to the initial estimate. That is, any transformation applied by the registration causes the alignment to diverge from the optimal solution. Consequently, a deviation from the ground truth transformation is considered an error in translation and rotation. The divergence behavior is usually not examined in related works but is of utmost importance here since the sparse point clouds acquired by our MAV quickly cause standard registration approaches to diverge when the angles between individual scan lines increase. We also analyze the convergence behavior of the registration approach as is done in related works. Here, the central question is if and how well a registration algorithm converges to the optimal solution for initial pose estimates that are noisy or considerably deviate from the optimal alignment.

In order to evaluate convergence and divergence behavior for different angular resolutions, we have created a dataset of organized point clouds containing ground truth pose information. It was recorded using the same rotating laser scanner but on a mobile ground robot standing still while acquiring 3D point clouds—thus avoiding inaccuracies in laser scan 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. The total trajectory length between the eight poses is roughly 50 m. It was recorded by Schadler et al. [40] in the arena of the DLR SpaceBot Cup[2] competition for semi-autonomous ex-

[1]We have made all datasets in this article publicly available. They can be obtained from our MAV Laser Mapping Dataset Repository: `http://www.ais.uni-bonn.de/mav_mapping`

[2]NimbRo Centauro, DLR SpaceBot Cup: `http://www.ais.uni-bonn.de/nimbro/Centauro`

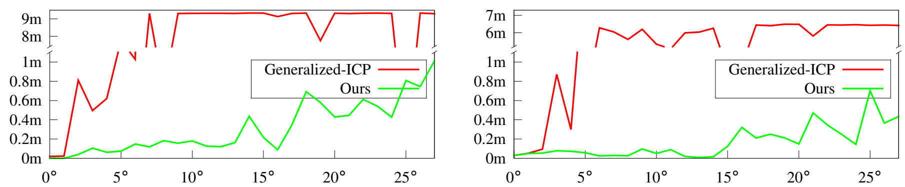


Figure 6: Divergence behavior for decreasing density (increasing angles $\Delta\theta$ between scan lines) of our approximate surface registration approach compared to Generalized-ICP for the alignment of two pairs of point clouds (left and right). Translation errors $e_{t,i}$ increase with an increasing angle $\Delta\theta$ between scan lines. Our approach can compensate for non-uniform densities and achieves fairly acceptable results even for larger angles between scans.

ploration and mobile manipulation in rough terrain (see Figure 5). 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 organized point clouds ($\Delta\theta \approx 0.3°$). We annotated each point cloud with the ground truth pose estimate obtained from an accurate multi-resolution surfel mapping approach for dense point clouds [40]. For the experimental evaluation, we generated thinned out versions of these eight original point clouds with different angular resolutions and angles $\Delta\theta \in [1°, 90°]$.

For both convergence and divergence behavior, we measure registration success in terms of the registration error. In particular, for consecutive point clouds acquired at times i and $i+1$, we inspect the relative deviations $\mathbf{E}_i$ with

$$\mathbf{E}_i := \left(\mathbf{Q}_i^{-1}\mathbf{Q}_{i+1}\right)^{-1}\left(\mathbf{P}_i^{-1}\mathbf{P}_{i+1}\right) \tag{16}$$

between ground truth poses $\mathbf{Q}$ and estimated poses $\mathbf{P}$. As suggested in [41], we focus on the translation error

$$e_{t,i} = \|\mathrm{trans}\left(\mathbf{E}_i\right)\|_2, \tag{17}$$

i.e., the Euclidean distance between the estimated (relative) pose estimates (trans$(\cdot)$ extracts the translation component). In case of $\boldsymbol{P}_i = \boldsymbol{Q}_i$ and $\boldsymbol{P}_{i+1} = \boldsymbol{Q}_{i+1}$, $\boldsymbol{E}_i$ is the identity matrix and $e_{t,i} = 0$.

Divergence Behavior. In order to evaluate the divergence behavior, we have chosen pairs of consecutive point clouds from the data set and registered the respective thinned out copies. In a comparative evaluation, we registered the point clouds of each pair using both the original Generalized-ICP algorithm and our variant with approximate surface registration. Figure 6 shows the results of this comparison with decreasing angular resolution (increasing angle $\Delta\theta$).

The presented results include only two of the seven pairs of consecutive point clouds; all results are available online together with the data set. Both algorithms achieve optimal registration results for the dense point clouds with deviations from ground truth of only few centimeters. In fact, it is hard to tell whether the pose estimate used as ground truth is better or worse than the

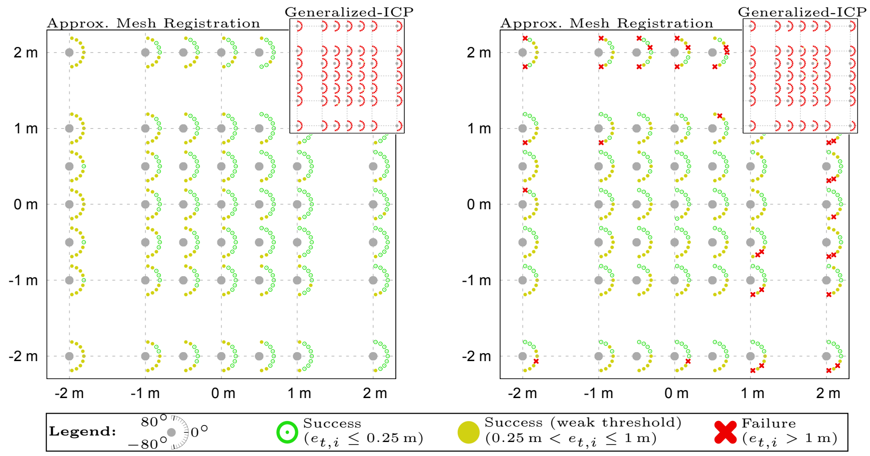


Figure 7: Convergence behavior of our approach for poor initial pose estimates at $\Delta\theta = 10°$ (using the same pairs of points clouds as in Figure 6). Registration success is measured w.r.t. the translation error $e_{t,i}$ using a strict threshold (green) and a weaker threshold (yellow). Exceeding the weaker threshold is considered a failure (red). Each subplot encodes translation errors along the x and y axes with seven initial orientations from $-80°$ to $80°$ rotation error with $0°$ pointing along the horizontal axis. For comparison, despite few exceptions Generalized-ICP fails in all cases.

achieved alignment. For increasing angles between scan lines, the Generalized-ICP algorithm quickly starts to fail showing the aforementioned behavior of dragging individual scan lines (and the scan origins) onto another instead of aligning sensed environmental structures. In its extreme, both scan origins coincide and the maximum error in the registration results reflects the Euclidean distance between the ground truth poses. Our approach achieves fairly acceptable results even for very low angular resolutions (angles between scans of $\Delta\theta > 15°$). For smaller angles ($\Delta\theta \leq 10°$), the resulting alignments are very accurate.

Convergence Behavior. In order to evaluate the convergence behavior, we have used the same pairs of point clouds as in the evaluation of the divergence behavior. Instead of using all available angular resolutions, we focus on the expected angular resolution of our scanner when flying (i.e., $\Delta\theta \approx 9°$). For each scan pair, we registered the respective point clouds under different initial conditions and quality of initial pose estimates. In particular, we simulate inaccuracies using translation errors of up to 2 m along the x and y axes (i.e., the plane the robot is moving on) and rotation errors of up to $80°$ about the z axis (i.e., affecting the robot's heading estimate).

In order to measure registration success for the different initial conditions, we have chosen two thresholds for the final translation error $e_{t,i}$ (17): a stricter one (0.25 m) and a weaker one (1 m) similar to the evaluation of registration algorithms by Magnusson et al. [42]. The intuition behind the two thresholds is that poses within the stricter translation threshold are difficult to tell apart for a human observer; poses within the weaker threshold are inaccurate but still fairly well aligned. We consider a registration as failed if the translation error exceeds the weaker threshold. Figure 7 presents the results of the evaluation with different initial conditions for the same two scan pairs as used in the evaluation of the divergence behavior. As can be seen, our approach fails in only very few cases (with high initial rotational error and/or high translation error). In the majority of registrations (even with high initial rotational and translation errors), our approach achieves an acceptable alignment even with the strict threshold. With very few exceptions, where the translation error stays within the weaker threshold, the Generalized-ICP algorithm fails in almost all cases.

5.2. Experiments on Simultaneous Localization and Mapping

In order to evaluate the performance of our complete mapping pipeline, we recorded a dataset with the flying MAV in a smaller indoor scenario equipped with a motion capture system. It contains a total of 1772 laser range scans in 82 aggregated point clouds. The overall trajectory length is 18.10 m. In a comparative evaluation, we process the complete dataset with different approaches: the original Generalized-ICP algorithm vs. our approximate mesh registration and the baseline SLAM approach with single edge connections vs. the proposed multi-edge approach. We also compare the two SLAM variants without prior registration, i.e., optimizing the initial vehicle trajectory. In order to evaluate the performance of the approaches under different initial conditions (quality of initial pose estimates), we run three series of experiments: with visual odometry estimates as initial pose estimates, without initial pose estimates (all transformations being identity), and with pose estimates considerably affected by noise (translation errors of up to 2 m and rotation errors of up to 45°).

For evaluating the accuracy of the pose estimates, we use an error metric proposed in [41]: the absolute trajectory error (ATE). The ATE focuses on global consistency by aligning and directly comparing absolute pose estimates (and trajectories):

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

with $\mathbf{F}_i(\Delta) := \mathbf{Q}_i^{-1}\mathbf{S}\mathbf{P}_i$, where $\mathbf{S}$ is the rigid-body transformation mapping the estimated trajectory $\mathbf{P}_{i:n}$ to the ground truth trajectory $\mathbf{Q}_{i:n}$. The operator $\mathrm{trans}\,(\mathbf{F})$ extracts the 3×1 translational component of $\mathbf{F}$.

For evaluating the quality of the resulting map (aggregated point map of all aligned point clouds), we use a measure of mean map entropy. For every point $\mathbf{p}_i$ in the resulting point map P, we compute the local entropy h by:

$$h(\boldsymbol{p}_i) = \frac{1}{2}\ln\left|2\pi e\boldsymbol{\Sigma}(\boldsymbol{p}_i)\right|, \tag{19}$$

where $\boldsymbol{\Sigma}(\boldsymbol{p}_i)$ is the covariance of the points in a radius r around $\boldsymbol{p}_i$. The mean map entropy $H(P)$ is then averaged over all points $\boldsymbol{p}_i$ in the map:

$$H(P) = \frac{1}{|P|} \sum_{i=1}^{|P|} h(\boldsymbol{p}_i), \tag{20}$$

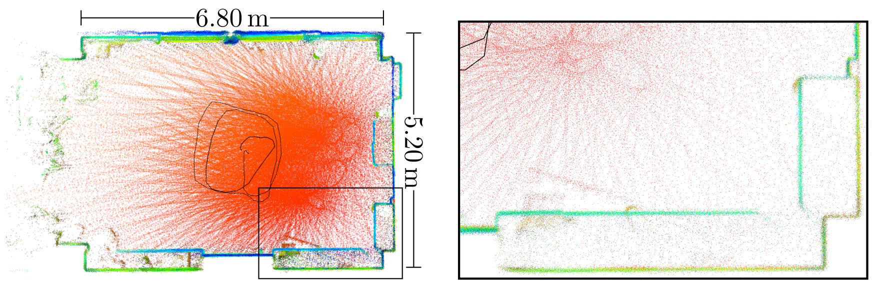
where $|P|$ is the number of points in P. The intuition behind this metric is the following: the sharper a map region is the lower is the value of the local point entropies in this region. That is, it encodes how planar the region appears in the final map. Consequently, a high-quality map with flat walls, floor and ceiling as well as sharp corners and edges will have a lower map entropy compared to a map resulting from a globally consistent but slightly inaccurate trajectory estimate. However, the metric assumes that the maps to be compared are roughly globally consistent for a fair comparison. Because of that, we also visually inspect the resulting maps and mark those that show inconsistencies.

In Figure 8, we report both the mean map entropy of the resulting point maps and the absolute trajectory errors (with root mean square error (RMSE), mean, standard deviation (stdev), min. and max. translation error). The most important finding here is that the proposed multi-edge graph-based approach with the approximate surface registration error metric outperforms the baseline approach with standard single-edge connections. Moreover, our approach yields exactly the same optimal results for all initial conditions.

Approximate Mesh Registration vs. Generalized-ICP algorithm. As can be expected from the direct comparison on pairwise registration in Section 5.1, our variant with approximate surface information clearly outperforms the original Generalized-ICP algorithm. The more accurate covariance estimates computed directly on the approximated surfaces allow correcting local alignments. Naturally, the estimated trajectory still drifts away from the ground truth estimate since smaller inaccuracies are accumulated in pairwise registration. Consequently, the resulting point map is globally not consistent.

SLAM with Multi-Edge vs. Single-Edge Connectivity. Both approaches can adequately compensate for the drift and produce globally consistent trajectories and maps. Still, by using locally very accurate correspondence covariances in the edges of the graph, the multi-edge variant achieves more accurate alignments and scores better in both absolute trajectory error and mean map entropy. Even in case of large simulated errors in the pose estimates, the initial registration with approximate surface information allows both variants converging to a globally consistent estimate.

SLAM without initial registration. In addition to the proposed pipelines of initial registration and subsequent pose graph optimization, we also evaluated the performance of pose graph optimization without initial registration, i.e., directly on the initial trajectory estimates. Here, the multi-edge variant quickly converges to its solution (within at most 5 outer iterations), regardless of the quality of the initial pose estimates. In fact, the resulting trajectories and maps



(a) Top view(s) of the scans aligned using our approach (MR+OPT)

		ATE (m)				Map entropy
Init.	Sequence	RMSE	mean ± stdev	min	max	mean
Visual Odometry	Initial	0.0297	0.0279±0.0101	0.0095	0.0544	-2.6575
	GICP	0.5723	0.5448±0.1753	0.1348	1.0191	-2.1956*
	MR	0.0988	0.0874±0.0459	0.0088	0.1899	-3.4229*
	MR+BL	0.0296	0.0267±0.0135	0.0050	0.0644	-3.4344
	MR+OPT	**0.0249**	**0.0223±0.0110**	**0.0030**	**0.0542**	**-3.8095**
	BL only	0.0286	0.0256±0.0128	0.0043	0.0622	-3.4920
	OPT only	**0.0249**	**0.0223±0.0110**	**0.0031**	**0.0563**	**-3.8098**
None (all Identity)	Initial	0.9371	0.8971±0.2707	0.3329	1.5810	-2.0035*
	GICP	0.7958	0.7632±0.2252	0.2600	1.2173	-2.2896*
	MR	0.0988	0.0874±0.0459	0.0088	0.1899	-3.4229*
	MR+BL	0.0299	0.0267±0.0134	0.0055	0.0640	-3.4380
	MR+OPT	**0.0249**	**0.0223±0.0110**	**0.0031**	**0.0563**	**-3.8096**
	BL only	0.2723	0.1492±0.2277	0.0152	1.3935	-2.5073*
	OPT only	**0.0249**	**0.0223±0.0111**	**0.0031**	**0.0563**	**-3.8097**
Simulated Errors	Initial	6.2885	5.9298±2.0934	1.1895	11.790	-3.3392*
	GICP	4.3437	3.8863±1.9400	0.8205	8.5243	-2.5488*
	MR	0.1216	0.1075±0.0568	0.0129	0.2265	-3.4135*
	MR+BL	0.0297	0.0265±0.0134	0.0054	0.0649	-3.4435
	MR+OPT	**0.0249**	**0.0223±0.0111**	**0.0031**	**0.0563**	**-3.8096**
	BL only	0.0824	0.0747±0.0346	0.0148	0.17593	-3.0243*
	OPT only	**0.0250**	**0.0226±0.0113**	**0.0035**	**0.0567**	**-3.8080**

* The resulting maps are globally not consistent, e.g., due to drifts.

(b) Absolute trajectory error and map entropy under different initial conditions

Figure 8: Results for the Motion Capture Dataset: (a) Top views of the aligned scans using our approach and (b) absolute trajectory error and map quality for the different approaches under different initial conditions: visual odometry as initial pose estimates, no initial pose estimates, and noise-affected pose estimates simulating errors (±200 cm, ±45°, uniformly distributed).
Legend: GICP (Generalized-ICP), MR (Mesh Reg., Generalized-ICP with approximate covariances), BL (single edge connections, Section 4.2), OPT (multi-edge connections, Section 4.3).

are almost identical. The baseline approach converges considerably slower (especially for large errors in the initial pose estimates) and does not find a globally consistent trajectory estimate in 10 iterations.

5.3. Results of Complete Mapping Missions with the Micro Aerial Vehicle

The small indoor environment in the previous experiment series does not pose major challenges. In fact, since it is only a single room where all environmental structures can be sensed from every view pose, the data set constitutes the best case for registration algorithms.

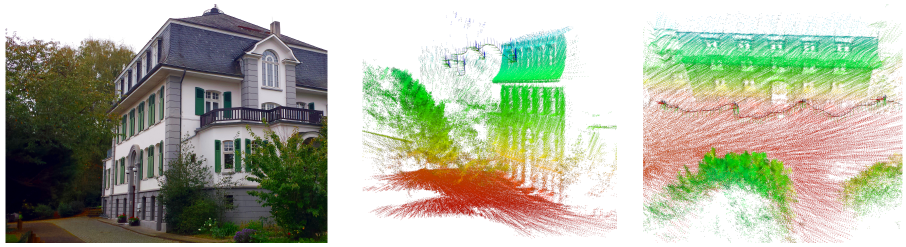
As a proof-of-concept for the complete system, we conducted two outdoor mapping missions. In these missions, the micro aerial vehicle autonomously navigated to a set of predefined waypoints in order to map a building of *Gut Frankenforst*—a research station operated by the Institute for Veterinary Research at the University of Bonn (see Figure 9a). Not only at the predefined waypoints but over the whole trajectory, the MAV collected laser range scans which where then processed offline by our approximate mesh registration and multi-edge pose graph optimization approach. This environment poses far more challenges than the indoor scenario: most notably a larger part of the measurements do not lie on the distinctive structures of the building but on vegetation around the building, thus forming rather random measurements when seen in individual sparse point clouds. Moreover, the point clouds only capture parts of the environment making it necessary to correctly detect and align loop closures.

The first mission aims at mapping the front facade of the building. The MAV captured a total of 2409 laser range scans over a trajectory of 30.43 m along the facade of the building. On a single core of an Intel Core i7-3740QM CPU, our approach took 92 s to construct a globally consistent point map out of the 118 aggregated point clouds. Most of this time was spent on the pose graph optimization while aggregating and pre-processing point clouds as well as registering consecutive point clouds was a matter of only few milliseconds per cloud. The pose graph optimization converged after four iterations. The resulting point map and trajectory estimate are shown in Figure 9b.

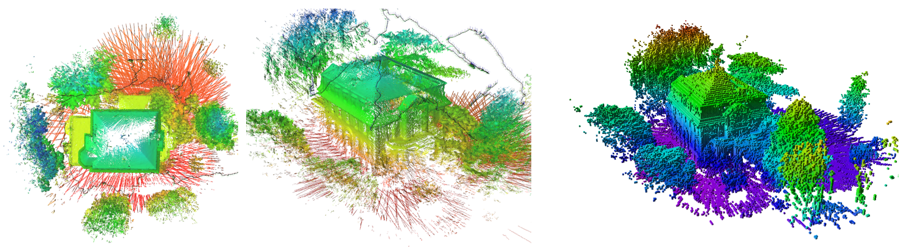
In the second mission, waypoints were distributed around the complete building. The MAV traveled a total of 307.13 m to reach all waypoints and collected 21 475 laser range scans. The scans were aggregated to 859 point clouds. Our approach took roughly 200 s to align all point clouds and construct a globally consistent map of the building and the surrounding vegetation. The resulting point map and trajectory estimate are shown in Figure 9c. Finally, we construct an OctoMap [17] as a memory-efficient representation of the environment, e.g., for being able to plan paths for future missions. It is shown in Figure 9d.

6. Conclusions and Future Work

We have presented a complete pipeline for registration and mapping with particularly sparse laser scans acquired by an autonomous MAV. The non-uniform point densities within and between individual laser range scans in the



(a) Photo of the building (b) Mapping the front facade: point map and trajectory



(c) Mapping the building: point map and trajectory (d) Final OctoMap of the building

Figure 9: Results of two complete missions for mapping a building (a): a shorter mission for mapping the front facade (b) and a longer mission for mapping the complete building (c). Shown are both the resulting map as an aggregation of all aligned point clouds and the estimated vehicle trajectory. The constructed OctoMap [17] is shown in (d).

aggregated point clouds negatively affect standard approaches to registration as well as neighborhood searches and local feature estimation. In order to compensate for the non-uniform point densities in the point clouds, we exploited the organized data structure and computed an approximate surface reconstruction. Point features such as local surface normals and covariances were then deduced from the topology in the resulting mesh in order to obtain proper estimates of the underlying surface even in regions where the measurement density is particularly sparse.

We presented a registration approach—a variant of the Generalized-ICP algorithm—that makes use of the approximated surface information. It is able to adequately align aggregated point clouds regardless of the quality of initial pose estimates. Moreover, experiments have shown that the proposed approach clearly outperforms the original Generalized-ICP algorithm for the sparse point clouds acquired by our MAV. It is very likely that the combination of approximate surface reconstruction and deducing surface statistics from the resulting mesh can also improve the performance of other surface-based registration algorithms, e.g., the one of Magnusson et al. [23].

For being able to construct globally consistent 3D environment maps, we have presented an approach to pose graph optimization again making use of the approximated surface information. Instead of representing relative pose estimates in single edges between connected vertices in the graph, it uses one edge per point correspondence between the acquired point clouds. Each of these edges encodes the same error metric as in the registration approach. Our experiments indicate that this multi-edge variant shows superior performance compared to a baseline system following the single-edge approach with robust covariance estimates.

In a final experiment, we could demonstrate that our approach is able to adequately align point clouds aggregated in real mapping missions and to provide both globally consistent environment maps and reliable trajectory estimates.

In this paper, the presented approach was only used offline to process the data after it had been acquired in a mapping mission. Furthermore, our approach adequately aligns the aggregated point clouds but does not change the pose of individual laser range scans within an aggregated point cloud. That is, our approach can compensate for pose estimation errors between view poses where point clouds have been aggregated but not for errors in the motion estimation during point cloud aggregation. In the worst case, a single point cloud may become ill-formed and not correctly aligned to the other point clouds (i.e., an outlier point cloud in the resulting map). It is a matter of future work to apply the registration and pose graph optimization pipeline online and also to correct the pose of individual laser range scans once the neighboring aggregated point clouds are aligned.

Acknowledgments

The authors gratefully acknowledge the financial support from the German Research Foundation (DFG) under grant BE 2556/7-1 and from the European Union's Seventh Framework Programme under grant agreement no 610917.

[1] T. Tomić, K. Schmid, P. Lutz, A. Domel, M. Kassecker, E. Mair, I. Grixa, F. Ruess, M. Suppa, D. Burschka, 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] S. Grzonka, G. Grisetti, W. Burgard, Towards a navigation system for autonomous indoor flying, in: Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA), 2009, pp. 2878–2883.

[3] A. Bachrach, R. He, N. Roy, Autonomous flight in unstructured and unknown indoor environments, in: Proc. of the European Micro Aerial Vehicle Conference (EMAV), 2009, pp. 1–8.

[4] S. Shen, N. Michael, V. Kumar, Autonomous multi-floor indoor navigation with a computationally constrained micro aerial vehicle, in: Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA), 2011, pp. 2968–2969.

[5] D. Droeschel, D. Holz, S. Behnke, Omnidirectional perception for lightweight MAVs using a continously rotationg 3D laser scanner, Photogrammetrie Fernerkundung Geoinformation (PFG) 5 (2014) 451–464.

[6] A. Segal, D. Hähnel, S. Thrun, Generalized-ICP, in: Proc. of Robotics: Science and Systems, 2009.

[7] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, W. Burgard, g2o: A general framework for graph optimization, in: Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA), 2011, pp. 3607–3613.

[8] D. Holz, S. Behnke, Registration of non-uniform density 3D point clouds using approximate surface reconstruction, in: Proc. of the Int. Symposium on Robotics (ISR) and the German Conference on Robotics (ROBOTIK), 2014.

[9] D. Holz, S. Behnke, Mapping with micro aerial vehicles by registration of sparse 3D laser scans, in: Proc. of the Int. Conf. on Intelligent Autonomous Systems (IAS), Padova, Italy, 2014.

[10] D. Scaramuzza, M. Achtelik, L. Doitsidis, F. Fraundorfer, E. Kosmatopoulos, A. Martinelli, M. Achtelik, M. Chli, S. Chatzichristofis, L. Kneip, et al., Vision-controlled micro flying robots: from system design to autonomous navigation and mapping in GPS-denied environments., IEEE Robotics and Automation Magazine 21 (3) (2014) 26–40.

[11] C. Urmson, J. Anhalt, H. Bae, J. A. D. Bagnell, C. R. Baker , R. E. Bittner, T. Brown, M. N. Clark, M. Darms, D. Demitrish, J. M. Dolan, et al., Autonomous driving in urban environments: Boss and the Urban Challenge, Journal of Field Robotics 25 (8) (2008) 425–466.

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

[13] S. Grzonka, G. Grisetti, W. Burgard, A fully autonomous indoor quadrotor, IEEE Transactions on Robotics 28 (1) (2012) 90–100.

[14] S. Huh, D. Shim, J. Kim, Integrated navigation system using camera and gimbaled laser scanner for indoor and outdoor autonomous flight of UAVs, in: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), 2013, pp. 3158–3163.

[15] J. Fossel, D. Hennes, D. Claes, S. Alers, K. Tuyls, OctoSLAM: a 3D mapping approach to situational awareness of unmanned aerial vehicles, in: Proc. of the Int. Conf. on Unmanned Aircraft Systems (ICUAS), 2013, pp. 179–188.

[16] S. Kohlbrecher, J. Meyer, O. von Stryk, U. Klingauf, 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.

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

[18] W. Morris, I. Dryanovski, J. Xiao, S. Member, 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.

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

[20] H. Cover, S. Choudhury, S. Scherer, S. Singh, Sparse tangential network (SPARTAN): Motion planning for micro aerial vehicles, in: Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA), 2013, pp. 2820–2825.

[21] J. Zhang, S. Singh, Loam: Lidar odometry and mapping in real-time, in: Proceedings of Robotics: Science and Systems, Berkeley, USA, 2014.

[22] A. Nüchter, K. Lingemann, J. Hertzberg, H. Surmann, 6D SLAM with approximate data association, in: Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA), 2005, pp. 242–249.

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

[24] P. J. Besl, N. D. McKay, A Method for Registration of 3-D Shapes, IEEE Transactions on Pattern Analysis and Machine Intelligence 14 (2) (1992) 239–256.

[25] D. Droeschel, J. Stückler, S. Behnke, Local multi-resolution surfel grids for MAV motion estimation and 3D mapping, in: Proc. of the Int. Conf. on Intelligent Autonomous Systems (IAS), 2014.

[26] M. Bosse, R. Zlot, P. Flick, Zebedee: Design of a spring-mounted 3-d range sensor with application to mobile mapping, IEEE Transactions on Robotics 28 (5) (2012) 1104–1119.

[27] U. Frese, P. Larsson, T. Duckett, A multilevel relaxation algorithm for simultaneous localization and mapping, IEEE Transactions on Robotics 21 (2) (2005) 196–207.

[28] E. Olson, J. Leonard, S. Teller, Fast iterative optimization of pose graphs with poor initial estimates, in: Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA), 2006.

[29] G. Grisetti, R. Kuemmerle, C. Stachniss, U. Frese, C. Hertzberg, Hierarchical optimization on manifolds for online 2D and 3D mapping, in: Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA), 2010.

[30] G. Grisetti, C. Stachniss, W. Burgard, Nonlinear constraint network optimization for efficient map learning, IEEE Transactions on Intelligent Transportation Systems 10 (3) (2009) 428–439.

[31] P. Agarwal, W. Burgard, C. Stachniss, Survey of geodetic mapping methods: Geodetic approaches to mapping and the relationship to graph-based slam, IEEE Robotics & Automation Magazine 21 (3) (2014) 63–80.

[32] F. Lu, E. Milios, Globally consistent range scan alignment for environment mapping, Autonomous Robots 4 (4) (1997) 333–349.

[33] D. Borrmann, J. Elseberg, K. Lingemann, A. Nüchter, J. Hertzberg, Globally consistent 3D mapping with scan matching, Robotics and Autonomous Systems 56 (2) (2008) 130–142.

[34] R. B. Rusu, N. Blodow, M. Beetz, Fast point feature histograms (FPFH) for 3D registration, in: Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA), 2009, pp. 3212–3217.

[35] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, W. Burgard, An evaluation of the RGB-D SLAM system, in: Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA), 2012, pp. 1691–1696.

[36] M. Ruhnke, R. Kümmerle, G. Grisetti, W. Burgard, Highly accurate 3D surface models by sparse surface adjustment, in: Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA), 2012, pp. 751–757.

[37] R. Zlot, M. Bosse, Efficient large-scale three-dimensional mobile mapping for underground mines, Journal of Field Robotics 31 (5) (2014) 758–779.

[38] D. Holz, S. Behnke, Approximate triangulation and region growing for efficient segmentation and smoothing of range images, Robotics and Autonomous Systems 62 (9) (2014) 1282–1293.

[39] A. Censi, An accurate closed-form estimate of ICP's covariance, in: Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA), Rome, Italy, 2007, pp. 3167–3172.

[40] M. Schadler, J. Stückler, S. Behnke, Rough terrain 3D mapping and navigation using a continuously rotating 2D laser scanner, KI - Künstliche Intelligenz 28 (2) (2014) 93–99.

[41] J. Sturm, N. Engelhard, F. Endres, W. Burgard, D. Cremers, A benchmark for the evaluation of RGB-D SLAM systems, in: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), 2012, pp. 573–580.

[42] M. Magnusson, A. Nüchter, C. Lörken, A. J. Lilienthal, J. Hertzberg, Evaluation of 3D registration reliability and speed – a comparison of ICP and NDT, in: Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA), 2009, pp. 3907–3912.