Continuous-Time SLAM—Improving Google's Cartographer 3D Mapping

Andreas Nüchter [a,b], Michael Bleier [b], Johannes Schauer [a] and Peter Janotta [c]

[a] Informatics VII – Robotics and Telematics, Julius-Maximilians University Würzburg, Germany; (andreas.nuechter, johannes.schauer)@uni-wuerzburg.de

[b] Zentrum für Telematik e.V., Würzburg, Germany; (andreas.nuechter, michael.bleier)@telematik-zentrum.de

[c] Measurement in Motion GmbH, Theilheim, Germany; peter.janotta@mim3d.de

Abstract: This paper shows how to use the result of Google's simultaneous localization and mapping (SLAM) solution, called Cartographer, to bootstrap a continuous-time SLAM algorithm that was developed by the authors and presented in previous publications. The presented approach optimizes the consistency of the global point cloud, and thus improves on Google's results. Algorithms and data from Google are used as input for the continuous-time SLAM software. In preceding work, the continuous-time SLAM was successfully applied to a similar backpack system which delivers consistent 3D point clouds even in the absence of an IMU. Continuous-time SLAM means that the trajectory of a mobile mapping system is treated in a semi-rigid fashion, i.e., the trajectory is deformed to yield a consistent 3D point cloud of the measured environment.

Keywords: 3D laser scanner; mapping; SLAM; trajectory optimization; personal laser scanner

1. Introduction

On October 5, 2016, Google released the source code of its real-time 2D and 3D simultaneous localization and mapping (SLAM) library Cartographer[1]. The utilized algorithms for solving SLAM in 2D have been described in a recent paper by the authors of the software (Hess et al., 2016). It can deliver impressive results—especially considering that it runs in real-time on commodity hardware. A publication describing the 3D mapping solution is still missing. The released software however, solves the problem. In addition, Google published a very demanding, high-resolution data set to the public for testing their algorithms. Also, custom data sets are easy to process, as Google's software comes with an

[1] https://opensource.googleblog.com/2016/10/introducing-cartographer.html

integration into the robot operating system (ROS) (Quigley et al., 2009). ROS is the de-facto standard middleware in the robotic community. It allows to connect heterogeneous software packages via a standardized inter-process communication (IPC) system and is available on recent GNU/Linux distributions.

Google's sample data set was recorded in the museum "Deutsches Museum" in München, Germany. It is the world's largest museum of science and technology, and has about 28,000 exhibited objects from 50 fields of science and technology. The data set was recorded with a backpack system, which features an inertial measurement unit (IMU) and two Velodyne PUCK (VLP-16) sensors. The processed trajectory was 108 meters long and contained 300,000 single 3D scans from the PUCK sensors.

Due to a high demand on flexible mobile mapping systems, mapping solutions on pushcarts, on trolleys, on mobile robots, and backpacks have recently been developed. Human-carried systems offer the advantage of overcoming doorsteps and the operator can open closed doors etc. To this end, several vendors build human-carried systems which are also often called personal laser scanners.

This article shows how to use the result of Google's SLAM solution to bootstrap our continuous-time SLAM algorithm. Our approach optimizes the consistency of the global point cloud, and thus improves on Google's results. In this article, the algorithms and data from Google are used as input for our continuous-time SLAM solution, which was recently published in (Elseberg et al. 2013). In preceding research, the presented algorithmic solution was successfully applied to a similar backpack system set up by the authors which delivers consistent 3D point clouds even in the absence of an IMU (Nüchter et al., 2015).

In the following, this article will discuss the related work with a focus on unconventional mobile mapping systems and the work on calibration, referencing and SLAM. Then, the focus is shifted towards the registration of 3D scans in more detail and the ICP algorithm described and the globally consistent scan matching is derived. This is finally extended to a continuous-time SLAM solution, which takes Google's Cartographer 3D Mapping as input.

2. Related Works

2.1. Laser Scanner on Robots and Backpacks

Mapping environments has received a lot of attention in the robotics community, especially after the appearance of cost effective 2D laser range finders. Seminal work with 2D profiles in robotics was performed by (Lu and Milios, 1994). After deriving 2D variants of the by now well-known ICP algorithm, they derived a PosegraphSLAM solution (Lu and Milios, 1997) that considers all 2D scans in a global fashion. Afterwards, many other approaches to SLAM were presented, including extended Kalman filters, particle filters, expectation maximization and GraphSLAM. These SLAM algorithms aimed at enabling mobile robots to map the environments where they have to carry out user-specific tasks. Thrun et al. (2000) presented a system, where a horizontally mounted scanner performed FastSLAM—a particle filter approach to SLAM—while an upward looking scanner was used to acquire 3D data, exploiting the robot motion to construct environments in 3D. Lu and Milios' approach was extended to 3D point clouds and possesses six degrees of freedom (DoF) in (Borrmann et al., 2008).

In 2004 an early version of a backpack system was presented. Saarinen et al., 2004, used the term Personal Localization And Mapping (PLAM). They used a horizontally mounted SICK LMS200 scanner in front of the human-carried system and put additional sensors and the computing equipment into a backpack. Chen et al. (2010), presented a backpack system that featured a number of lightweight 2D profilers (Hokuyo scanner) mounted in different viewing directions. In previous work, the authors applied the algorithms to a backpack system without an IMU (Nüchter et al., 2015). The system consists of a horizontally mounted SICK LMS100 scanner and a spinning Riegl VZ400. Similar to the work of Thrun et al. (2000) a horizontal scanner is used to estimate an initial trajectory that is afterwards updated to regard the six DoF motion. The term Personal Laser Scanning System was coined by Liang et al. (2014). They use a single FARO scanner and rely on the global navigation satellite system (GNSS) system. Similarly, the commercially available ROBIN system features a RIEGL VUX scanner and GNSS. In contrast, the Leica Pegasus is a commercially available backpack wearable mobile mapping solution, which is composed of two Velodyne PUCK scanners, cameras and a GNSS. The PUCKs scan 300.000 points per second and have a maximal range of 100 meters. Sixteen profilers are combined to yield a vertical field of view of ±15 degree.

The Google Cartographer backpack was initially presented in September 2014. Back then, the backpack system was based on two Hokuyo profilers and an internal measurement unit (IMU). The current version features two Velodyne PUCK scanners. Figure 1 shows the system from Google and our backpack solution.

2.2. Calibration, Referencing, and SLAM

To acquire high-quality range measurement data with a mobile laser scanner, the position and orientation of every individual sensor have to be known. Traditionally, scanners, GPS and IMU are calibrated against other positioning devices whose pose in relation to the system is already known. The term Boresight calibration is used for the technique of finding the rotational parameters of the range sensor with respect to the already calibrated IMU/GPS unit. In the airborne laser scanning community, automatic calibration approaches are known (Skaloud and Schaer, 2007), and similarly vehicle-based kinematic laser scanning has been considered (Rieger et al., 2010). In the robotics community, there exist approaches for calibrating several range scanners semi-automatically, i.e., with manually labelled data (Underwood et al., 2009) or using automatically computed quality metrics (Sheehan et al., 2011), (Elseberg et al., 2013). Often, vendors do not make their calibration methods public and unfortunately, the authors of this paper have no information on the calibration of the Google Cartographer backpack. In general, calibration inaccuracies can, to some extent, be compensated with a SLAM algorithm.

In addition to from sensor misalignment, timing related issues are second sources of errors. All subsystems on a mobile platform need to be synchronized to a common time frame. This is often accomplished with pure hardware via triggering or with mixes of hardware and software such as pulse per second (PPS) or the network time protocol (NTP). However, good online synchronization is not always available for all sensors. Olson, 2010, has developed a solution for the synchronization of clocks that can be applied after the fact. In ROS, sensor data is time-stamped when it arrives and it is recorded in an open file format (.bag files). Afterwards, one works with the time-stamped data using nearest values or interpolation.

As the term direct referencing or direct Geo-referencing implies, it is the direct measurement of the position and orientation of a mapping sensor, i.e., the laser scanner, such that each range value can be referenced without the need for collecting additional information. This means that the trajectory is then used to "unwind" the laser range measurements to produce the 3D point cloud. This approach has been taken by (Liang et al., 2014).

Some systems employ a horizontally mounted scanner and perform 2D SLAM on the acquired profiles. Thrun et al. (2000), used FastSLAM, whereas Nüchter et al. (2015) used SLAM based on the truncated signed distance function (TSD SLAM), or alternatively, HectorSLAM (Kohlbrecher et al., 2011). These 2D SLAM approaches produce 2D grid maps. Similarly, Google's Cartographer code is for creating 2D grid maps (Hess et al., 2016). Afterwards, the computed trajectory and the IMU measurements are used to "unwind" the laser range measurements to produce the 3D point cloud.

The Google Cartographer library achieves its outstanding performance by grouping scans into probability grids that they call submaps and by using a branch and bound approach for loop closure optimization. While new scans are matched with and subsequently entered into the current submap in the foreground, in the background, the library matches scans to nearby submaps to create loop closure constraints and to continually optimize the constraint graph of submaps and scan poses. The authors differentiate between local scan matching which inserts new scans into the current submap and which will accumulate errors over time and global SLAM which includes loop closing and which removes the errors that have accumulated in each submap that are part of a loop. Both local and global matching are run at the same time.

During local scan matching, the Cartographer library matches each new scan against the current submap using a Ceres-based scan matcher (Agarwal and others). A submap is a regular probability grid where each discrete grid point represents the probability that the given grid point is obstructed or free. These two sets are disjoint. A grid point is obstructed if it contains an observed point. Free points are computed by tracing the laser beam from the estimated scanner location to the measured point through the grid. The optimization function of the scan matcher makes use of the probability grid as part of its minimization function.

During global SLAM, finished submaps (those that no longer change) and the scans they contain are considered for loop closing. As is the case during local scan matching, the problem is passed to Ceres as a nonlinear least squares problem. The algorithm is accurate down to the groups of points defined by the regular probability grid of each submap. By taking the submap grid size as translation step delta and the angle under which a grid point is seen at maximum range as the rotation step delta, a finite set of possible transformations is created. This solution space is searched using a branch and bound approach where nodes are traversed using a greedy depth first search and the upper bound of the inner nodes is defined in terms of computational effort and quality of the bound. To compute the upper bound efficiently, grids are precomputed for tree heights that overlay the involved submaps and store the maximum values of possible scores for each obstructed grid point. This operation is done in $O(n)$ with n being the number of obstructed grid points in each precomputed grid.

Hess et al. (2016), describe the 2D version of the algorithm, which uses the horizontally mounted 2D profiler. The provided data sets also contain data from a setup with Velodyne PUCK scanners (cf. Figure 1). Their algorithm is able to process 3D data and to output poses with six DoF, however, a description of their 3D approach is missing from their paper. Nevertheless, one can understand from their published source code that their 3D implementation is mostly an extension of their 2D approach to three dimensions with a 3D probability grid. Some changes have been made to improve performance. For example, the 3D grid is not fully traversed to find free grid cells but only a configurable distance up to the measured point is checked.

Only a few approaches optimize the whole trajectory in a continuous fashion. Stoyanov and Lilienthal, (2009), presented a non-rigid optimization for a mobile laser scanning system. They optimize point cloud quality by matching the beginning and the end of a single scanner rotation using ICP. The estimate of the 3D pose difference between the two points in time is then used to optimize the robot trajectory in between. In a similar approach, Bosse and Zlot, (2009), use a modified ICP with a custom correspondence search to optimize the pose of six discrete points in time of the trajectory of a robot during a single scanner rotation. The trajectory in between is modified by distributing the errors with a cubic spline. The software of Riegl RiPRECISION MLS automatically performs adjustments of GNSS/INS trajectories to merge overlapping mobile scan data based on planar surface elements. Our own continuous-time SLAM solution improves the entire trajectory of the data set simultaneously based on the raw point cloud. The algorithm is adopted from (Elseberg et al., 2013), where it was used in a different mobile mapping context, i.e., on platforms such as the LYNX mobile mapper or the Riegl VMX-250. As no motion model is required, it can be applied to any continuous trajectory.

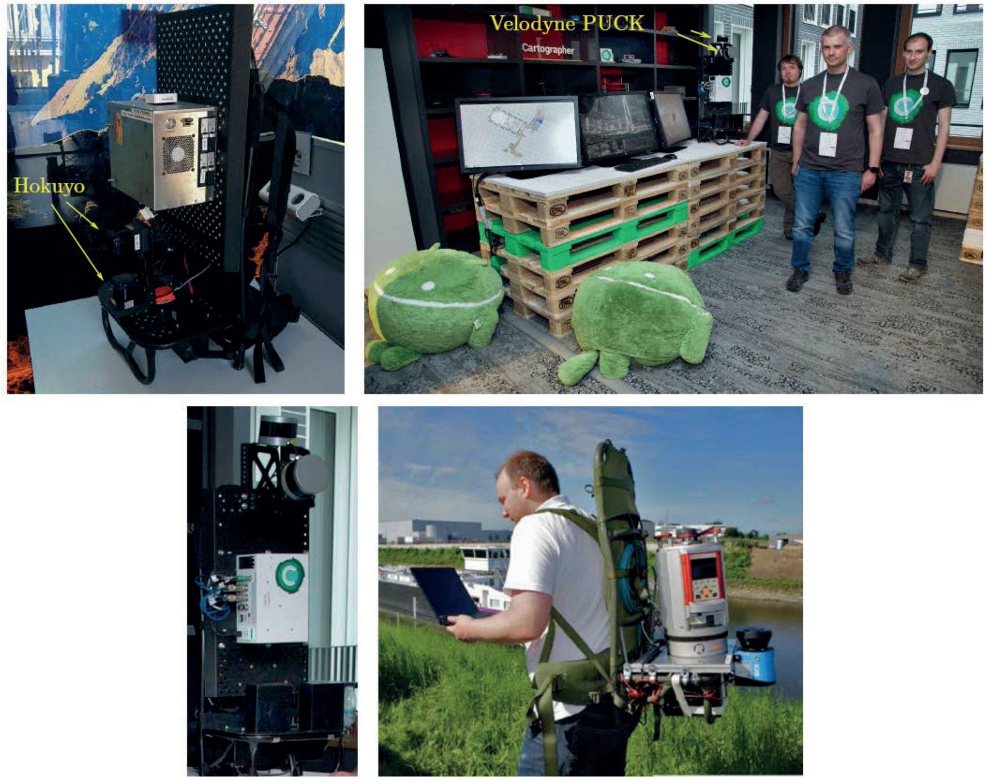

Figure 1. Above left: Google's Cartographer system featuring two Hokuyo laser scanners (image: Google blog). Above right and below left: Google's Cartographer system with two Velodyne PUCKs and the Cartographer team (image courtesy of the Cartographer team). Below right: Second author operating Würzburg's backpack scanner.

3. Registration of 3D Laser Scans

3.1. Feature-Based Registration

Many state-of-the-art registration methods rely on initial pose (position and orientation) estimates acquired by global positioning systems (GPS) or local positioning using artificial landmarks or markers as reference (Wang et al. 2008). Pose information is hard to acquire and in many scenarios is prone to errors or not available at all. Thus, registration without initial pose estimates and place recognition are highly active fields of research.

In addition to range values, laser scanners record the intensity of the reflected light. These intensities provide additional information for the registration process. Böhm and Becker, (2007), suggest using SIFT (scale-invariant feature transform) features for automatic registration and present an example of a successful registration on a 3D scan with a small field of view. Wang and Brenner, (2008), extended this work by using additional geometry features to reduce the number of matching outliers in panoramic outdoor laser scans. Kang et al. (2009), propose a similar technique for indoor and outdoor environments. Weinmann et al. (2011), use a method that is based on both reflectance images and range information. After extraction of characteristic 2D points based on SIFT features, theses points are projected into 3D space by using interpolated range information. For a new scan, combining the 3D points with 2D observations on a virtual plane yields 3D-to-2D correspondences from which the coarse transformation parameters can be estimated via a RANSAC (random sample consensus) based registration scheme including a single step outlier removal for checking consistency (Weinmann, et al., 2011). They extend their method (Weinmann and Jutzi, 2011) to calculate the order of the scans in unorganized terrestrial laser scan data by checking the similarity of the respective reflectance images via the total number of SIFT correspondences between them. Bendels et al., (2004), exploit intensity images often recorded with the range data and propose a fully automatic registration technique using 2D-image features. The fine registration of two range images is performed by first aligning the feature points themselves, followed by a so-called constrained-domain alignment step. In the latter, rather than feature points, they consider feature surface elements. Instead of using a single 3D-point as feature, they use the set of all points corresponding to the image area determined by the position and scale of the feature.

Other approaches rely only on the 3D structure. Brenner et al., (2008), use 3D planar patches and the normal distribution transform (NDT) on several 2D scan slices for a coarse registration. Similarly, Pathak et al., (2010), evaluated the use of planar patches and found that it is mostly usable. A solution using the NDT in 3D is given (Magnusson et al., 2009). While this approach computes global features of the scan, several researchers use features that describe small regions of the scan for place recognition and registration (Huber 2002) (Steder et al., 2010) (Barnea and Filin, 2008). Flint et al., (2007), use a key point detector called ThrIFT, to detect repeated 3D structures in range data of building facades.

In addition to coarse, feature-based registration, many authors use the well-known iterative closest point algorithm (ICP) for fine registration.

3.2. Registration without Using Features— The ICP Algorithm

The following method is the *de facto* standard for registration of two 3D point clouds, given a good initial pose estimate. ICP requires no computation of features. Instead, it matches raw point clouds by selecting point correspondences on the basis of smallest distances and by minimizing the resulting Euclidean error. This iterative algorithm converges to a local minimum. Good starting estimates significantly improve the matching results, i.e., they ensure that ICP converges to a correct minimum. The complete algorithm was invented at the same time in 1991 by Besl and McKay, (1992), by Chen and Medioni, (1991) and by Zhang, (1992). The method is called the Iterative Closest Points (ICP) algorithm.

Given two independently acquired sets of 3D points, M (model set) and D (data set) which correspond to a single shape, one wants to find the transformation $(\boldsymbol{R}, \boldsymbol{t})$ consisting of a rotation matrix $\boldsymbol{R}$ and a translation vector $\boldsymbol{t}$ which minimizes the following cost function

$$E(\boldsymbol{R}, \boldsymbol{t}) = \sum_{i=1} \|\boldsymbol{m}_i - (\boldsymbol{R}\boldsymbol{d}_i - \boldsymbol{t})\|^2 \tag{1}$$

All corresponding points can be represented in a tuple $(\boldsymbol{m}_i, \boldsymbol{d}_i)$ where $\boldsymbol{m}_i \in M \subset \widehat{M}$ and $\boldsymbol{d}_i \in D \subset \widehat{D}$. Two calculations need to be made: First, the corresponding points, and second, the transformation $(\boldsymbol{R}, \boldsymbol{t})$ that minimizes $E(\boldsymbol{R}, \boldsymbol{t})$ on the basis of the corresponding points. The ICP algorithm uses the closest points as corresponding points. A sufficiently accurate starting guess enables the ICP algorithm to converge to the correct minimum.

Current research in the context of ICP algorithms mainly focuses on fast variants of ICP algorithms (Rusinkiewicz and Levoy, 2001). If the inputs are 3D meshes, then a point-to-plane metric can be used instead of Equation (1). Minimizing the use of a point-to-plane metric outperforms the standard point-to-point one, but requires the computation of normal metrics and meshes in a pre-processing step.

4. Globally Consistent n-Scan Matching

Chen and Medioni, (1992), aimed at globally consistent range image alignment when introducing an incremental matching method, i.e., all new scans are registered against the so-called metascan, which is the union of the previously acquired and registered scans. This method does not spread out the error and is order-dependent.

Bergevin et al., (1996), Stoddart and Hilton, (1996), Benjemaa and Schmitt, (1997, 1998), and K. Pulli, (1999) present iterative approaches. Based on networks representing overlapping parts of images, they use the ICP algorithm for computing transformations that are applied after all correspondences between all views have been found. However, the focus of research is mainly 3D modelling of small objects using a stationary 3D scanner and a turn table. Therefore, the used networks consist mainly of one loop (Pulli, 1999). These solutions are locally consistent algorithms that retain the analogy of the spring system (Cunnington and Stoddart, 1999), whereas true globally consistent algorithms minimize the error function in one step.

A probabilistic approach was proposed by Williams and Bennamoun, (1999), where each scan point is assigned a Gaussian distribution in order to model the statistical errors made by laser scanners. This causes high computation time due to the large amount of data in practice. Krishnan, et al., (2000), presented a global registration algorithm that minimizes the global error function by optimization on the manifold of 3D rotation matrices.

To register scans in a globally consistent fashion, a network of poses is formed, i.e., a graph. Every edge represents a link $j \rightarrow k$ of matchable poses. The error function is extended to include all links and to minimize for all rotations and translations at the same time.

$$E = \sum_{j \rightarrow k} \sum_{i=1} \left\| \left(\boldsymbol{R_j m}_i - \boldsymbol{t}_j \right) - \left(\boldsymbol{R_k d}_i - \boldsymbol{t_k} \right) \right\|^2 \tag{2}$$

For some applications, it is necessary to have a notion of the uncertainty of the poses calculated by the registration algorithm. The following is an extension of the probabilistic approach first proposed by Lu and Milios, (1997) to six DoF. This extension is not straightforward, since the matrix decomposition, i.e., Equation (20), cannot be derived from first principles. For a more detailed description of these extensions, refer to Borrmann et al., (2008). In addition to the poses $\boldsymbol{X}_j$, the pose estimates $\overline{\boldsymbol{X}}_j$ and the pose errors $\Delta\boldsymbol{X}_j$ are required.

The positional error of two poses $\boldsymbol{X}_j$ and $\boldsymbol{X}_k$ is described by:

$$E_{k,j} = \sum_{i=1}^{m} \left\| \boldsymbol{X}_j \oplus \boldsymbol{d}_i - \boldsymbol{X}_k \oplus \boldsymbol{m}_i \right\| = \sum_{i=1}^{\mathrm{m}} \left\| \boldsymbol{Z}_i\left(\boldsymbol{X}_j, \boldsymbol{X}_k\right) \right\|^2 \tag{3}$$

here, $\oplus$ is the compounding operation that transforms a point into the global coordinate system. For small pose differences, $E_{k,j}$ can be linearized by use of a Taylor expansion:

$$\boldsymbol{Z}_i(\boldsymbol{X}_j,\boldsymbol{X}_k) \approx \ \overline{\boldsymbol{X}}_j \oplus \ \boldsymbol{d}_i - \overline{\boldsymbol{X}}_k \oplus \ \boldsymbol{m}_i - \ (\nabla_j \ \boldsymbol{Z}_i(\overline{\boldsymbol{X}}_j,\overline{\boldsymbol{X}}_k)\Delta\boldsymbol{X}_j - \nabla_k\boldsymbol{Z}_i \ (\boldsymbol{X}_j,\overline{\boldsymbol{X}}_k)\Delta\boldsymbol{X}_k) \tag{4}$$

where ∇k denotes the derivative with respect to $\boldsymbol{X}_j$ and $\boldsymbol{X}_k$. Utilizing the matrix decompositions $\boldsymbol{M}_i\boldsymbol{H}_j$ and $\boldsymbol{D}_j\boldsymbol{H}_k$ of the respective derivatives that separate the poses from the associated points gives:

$$\begin{aligned} \boldsymbol{Z}_i \ (\boldsymbol{X}_j,\boldsymbol{X}_k) &= \boldsymbol{Z}_i(\ \overline{\boldsymbol{X}},\overline{\boldsymbol{X}}_k) - \ (\boldsymbol{M}_i \ \boldsymbol{H}_j \ \Delta\boldsymbol{X}_j - \boldsymbol{D}_i \ \boldsymbol{H}_k \ \Delta \ \boldsymbol{X}_k \ ) \\ &= \ \boldsymbol{Z}_i(\overline{\boldsymbol{X}},\boldsymbol{X}_k) \ - \ (\boldsymbol{M}_i \ \boldsymbol{X}'_j \ - \ \boldsymbol{D}_i \ \boldsymbol{X}'_k \ ) \end{aligned} \tag{5}$$

Appropriate decompositions are given for both the Euler angles and quaternion representation in the following paragraphs. Because $\boldsymbol{M}_i$ as well as $\boldsymbol{D}_i$ are independent of the pose, the positional error $\boldsymbol{E}_{j,k}$ is minimized with respect to the new pose difference $\boldsymbol{E}'$, i.e.,

$$\begin{aligned} \boldsymbol{E}_{j,k} &= \ (\boldsymbol{H}_j \ \Delta \ \boldsymbol{X}_j \ - \ \boldsymbol{H}_k \ \Delta \ \boldsymbol{X}_k \\ &= \ (\boldsymbol{X}'_j \ - \ \boldsymbol{X}'_k) \end{aligned} \tag{6}$$

is linear in the quantities $\boldsymbol{X}_j$ that will be estimated so that the minimum of $\boldsymbol{E}_{j.k}$ and the corresponding covariance are given by

$$\overline{\boldsymbol{E}}_{j,k} = \ (\boldsymbol{M}^T\boldsymbol{M}^{-1})\boldsymbol{M}^T\boldsymbol{Z} \tag{7}$$

$$\boldsymbol{C}_{j,k} = \ s\text{^}2 \ (\boldsymbol{M}^T\boldsymbol{M}) \tag{8}$$

where s^2 is the unbiased estimate of the covariance of the identically, independently distributed errors of $\boldsymbol{Z}$:

$$s^2 = \frac{(\boldsymbol{Z} - \boldsymbol{M}\overline{\boldsymbol{E}})^T(\boldsymbol{Z} - \boldsymbol{M}\overline{\boldsymbol{E}})}{(2m-3)}. \tag{9}$$

Here, $\boldsymbol{Z}$ is the concatenated vector consisting of all $\boldsymbol{Z}_i(\overline{\boldsymbol{X}}_j,\overline{\boldsymbol{X}}_k)$ and $\boldsymbol{M}$ the concatenation of all $\boldsymbol{M}_i$s.

Up to now, all considerations have been on a local scale. With the linearized error metric $\boldsymbol{E}'_{j,k}$ and the Gaussian distribution $\overline{\boldsymbol{E}}_{j,k}$, $\boldsymbol{C}_{j,k}$ a Mahalanobis distance that describes the global error of all the poses is constructed:

$$\begin{aligned} \boldsymbol{W} &= \sum_{j\to k}(\overline{\boldsymbol{E}}_{j,k} - \boldsymbol{E}'_{j,k})^{-1} \ \boldsymbol{C}_{j,k}^{-1}(\overline{\boldsymbol{E}}_{j,k} - \boldsymbol{E}'_{j,k}) \\ &= \sum_{j\to k}(\overline{\boldsymbol{E}}_{j,k} - (\boldsymbol{X}'_j - \boldsymbol{X}'_k))^{-1} \end{aligned} \tag{10}$$

In matrix notation, $\boldsymbol{W}$ becomes:

$$\boldsymbol{W} = (\overline{\boldsymbol{E}} - \boldsymbol{HX})^T \boldsymbol{C}^{-1}(\overline{\boldsymbol{E}} - \boldsymbol{HX}). \tag{11}$$

Here, $\boldsymbol{H}$ is the signed incidence matrix of the pose graph, $\overline{\boldsymbol{E}}$ is the concatenated vector consisting of all $\overline{\boldsymbol{E}}'_{j,k}$ and $\boldsymbol{C}$ is a block-diagonal matrix comprised of $\boldsymbol{C}^{-1}_{j,k}$ as submatrices. Minimizing this function yields new optimal pose estimates. The minimization of $\boldsymbol{W}$ is accomplished via the following linear equation system:

$$(\boldsymbol{H}^T \boldsymbol{C}^{-1} \boldsymbol{H})\boldsymbol{X} = \boldsymbol{H}^T \boldsymbol{C}^{-1} \overline{\boldsymbol{E}} \tag{12}$$

$$\boldsymbol{BX} = \boldsymbol{A} \tag{13}$$

The matrix $\boldsymbol{B}$ consists of the submatrices

$$\boldsymbol{B}_{j,k} = \begin{cases} \sum\limits_{k=0}^{n} \boldsymbol{C}^{-1}_{j,k} & (j = k) \\ \boldsymbol{C}^{-1}_{j,k} & (j = k) \end{cases} \tag{14}$$

The entries of $\boldsymbol{A}$ are given by:

$$\boldsymbol{A}_j = \sum_{\substack{k=0 \\ k \neq j}}^{n} \boldsymbol{C}^{-1}_{j,k} \;\; \overline{\boldsymbol{E}}_{j,k}. \tag{15}$$

In addition to $\boldsymbol{X}$, the associated covariance of $\boldsymbol{C}_X$ is computed as follows:

$$\boldsymbol{C}_X = \boldsymbol{B}^{-1} \tag{16}$$

Note that the results have to be transformed in order to obtain the optimal pose estimates.

$$\boldsymbol{X}_j = \overline{\boldsymbol{X}}_j - \boldsymbol{H}_j^{-1} \boldsymbol{X}'_j \tag{17}$$

$$\boldsymbol{C}_j = (\boldsymbol{H}_j^{-1} \boldsymbol{C}_j^X \left(\boldsymbol{H}_j^{-1}\right)^T. \tag{18}$$

The representation of pose $\boldsymbol{X}$ in Euler angles, as well as its estimate and error is as follows:

$$= \begin{pmatrix} t_x \\ t_y \\ t_z \\ \theta_x \\ \theta_x \\ \theta_z \end{pmatrix}, \overline{\boldsymbol{X}} = \begin{pmatrix} \bar{t}_x \\ \bar{t}_y \\ \bar{t}_z \\ \bar{\theta}_x \\ \bar{\theta}_x \\ \bar{\theta}_z \end{pmatrix}, \Delta \boldsymbol{X} = \begin{pmatrix} \Delta t_x \\ \Delta t_y \\ \Delta t_z \\ \Delta \theta_x \\ \Delta \theta_x \\ \Delta \theta_z \end{pmatrix} \tag{19}$$

The matrix decomposition $\boldsymbol{M}_i\boldsymbol{H} = \nabla\boldsymbol{Z}_i\overline{\boldsymbol{X}}$ is given by

$$\boldsymbol{H} = \begin{pmatrix} 1 & 0 & 0 & 0 & \bar{t}_z\cos(\bar{\theta}_x) + \bar{t}_y\sin(\bar{\theta}_x) & \bar{t}_y\cos(\bar{\theta}_y)\cos(\bar{\theta}_x) - \bar{t}_z\cos(\bar{\theta}_x)\sin(\bar{\theta}_x) \\ 0 & 1 & 0 & -\bar{t}_z & -\bar{t}_x\sin(\bar{\theta}_x) & \bar{t}_x\cos(\bar{\theta}_x)\cos(\bar{\theta}_x) - \bar{t}_z\sin(\bar{\theta}_x) \\ 0 & 0 & 1 & \bar{t}_y & -\bar{t}_x\cos(\bar{\theta}_x) & \bar{t}_x\cos(\bar{\theta}_y)\sin(\bar{\theta}_x) + \bar{t}_y\sin(\bar{\theta}_y) \\ 0 & 0 & 0 & 1 & 0 & \sin(\bar{\theta}_y) \\ 0 & 0 & 0 & 0 & \sin(\bar{\theta}_x) & \cos(\bar{\theta}_x)\cos(\bar{\theta}_y) \\ 0 & 0 & 0 & 0 & \cos(\bar{\theta}_y) & -\cos(\bar{\theta}_y)\sin(\bar{\theta}_x) \end{pmatrix} \tag{20}$$

$$\boldsymbol{M}_i = \begin{pmatrix} 1 & 0 & 0 & 0 & -d_{y,i} & -d_{z,i} \\ 0 & 1 & 0 & d_{z,i} & d_{x,i} & 0 \\ 0 & 0 & 1 & -d_{y,i} & 0 & d_{x,i} \end{pmatrix} \tag{21}$$

As required, $\boldsymbol{M}_i$ contains all point information while $\boldsymbol{H}$ expresses the pose information. Thus, this matrix decomposition constitutes a pose linearization similar to those proposed in the preceding sections. Note that, while the matrix decomposition is arbitrary with respect to the column and row ordering of $\boldsymbol{H}$, this particular description was chosen due to its similarity to the 3D pose solution given in (Lu and Milios, 1997).

5. Continuous-Time SLAM

Unlike other state-of-the-art algorithms, (Stoyanov and Lilienthal, 2009 and Bosse et al., 2012), our continuous-time SLAM algorithm is not restricted to purely local improvements. Our method makes no rigidity assumptions, except for the computation of the point correspondences. For instance, the method requires no explicit motion model of a vehicle, thus it works well on backpack systems. The continuous-time SLAM for trajectory optimization works in full six DoF. The algorithm requires no high-level feature computation, i.e., it requires only the points themselves.

In the case of mobile mapping, one does not have separate terrestrial 3D scans. In the current state-of-the-art in the robotics community developed by Bosse et al. (2012), for improving overall map quality of mobile mappers, the time is coarsely discretized and the scene is described by features, i.e., local planar patches. This results in a partition of the trajectory into sub-scans that are treated rigidly. Then, rigid registration algorithms such as the ICP and other solutions to the SLAM problem are employed. Obviously, trajectory errors within a sub-scan cannot be improved in this fashion. Applying rigid pose estimation to this non-rigid problem directly is also problematic since rigid transformations can only approximate the underlying ground truth. When a finer discretization is used, single 2D scan slices or single points result that do not constrain a six DoF pose sufficiently for rigid algorithms.

More mathematical details of our algorithm are in the available open-source code and are given in Elseberg et al. (2013). Essentially, the algorithm first splits the trajectory into sections, and matches these sections using the automatic high-precision registration of terrestrial 3D scans, i.e., globally consistent scan matching that is the 6D SLAM core. Here, the graph is estimated using a heuristic that measures the overlap of sections using the number of closest point pairs. After applying globally consistent scan matching on the sections, the actual continuous-time or semi-rigid matching as described in (Elseberg et al., 2013) is applied, using the results of the rigid optimization as starting values to compute the numerical minimum of the underlying least square problem. To speed up the calculations, the algorithm exploits the sparse Cholesky decomposition by Davis (2006).

Given a trajectory estimate, the point cloud is "unwound" into the global coordinate system and uses a nearest neighbour search to establish correspondences at the level of single scans (those that can be single 2D scan profiles). Then, after computing the estimates of pose differences and their respective covariance, the algorithm optimizes the trajectory. In a pre-dependent step, trajectory elements in every *k* step are considered and *l* trajectory elements around these steps are fused temporarily into a meta-scan.

A key issue in continuous-time SLAM is the search for closest point pairs. An octree and a multi-core implementation using OpenMP is used to solve this task efficiently. A time-threshold for the point pairs is used, i.e., the algorithm matches only to points if they were recorded at least t_d time steps away.

Finally, all scan slices are joined in a single point cloud to enable efficient viewing of the scene. The first frame, i.e., the first 3D scan slice from the PUCKs scanner, defines the arbitrary reference coordinate system.

6. Bootstrapping Continuous-Time SLAM with Google's SLAM Solution

To improve the Cartographer 3D mapping, the graph is estimated using a heuristics that measures the overlap of sections using the number of closest point pairs. After applying globally consistent scan matching on the sections for several iterations, the actual continuous-time SLAM is started.

The data set provided by Google is challenging in several ways: Due to the enormous amount of data, clever data structures are needed to store and access the point cloud. The trajectory is slit every 300 PUCK-scans and ±150 PUCK-scans are joined into a meta-scan. These meta-scans are processed with an octree where a voxel size of 10 cm is used to reduce the point cloud by selecting one point per voxel. We prefer a data structure that stores the raw point cloud over a highly approximate voxel representation. While the latter one is perfectly justifiable for some use cases, it is incompatible with tasks that require exact point measurements such as scan matching. Our implementation of an octree prioritizes memory efficiency. The implementation uses pointers in contrast to serialized pointer-free encodings in order to efficiently access the large amounts of data. The octree is free of redundancies and is nevertheless capable of fast access operations. Our implementation allows for access operations in $O(\log n)$. The usage of 6 bytes for pointers is already sufficient to address a total of 256 terabyte. Two-bit fields signal if there is a child or leaf node, thus our implementation needs a few bit operations and 8 bytes are sufficient for an octree node.

The pre-processing step of the continuous-time SLAM runs for 20 iterations, where the edges in the graph are added, when more than 400 point pairs between these meta-scans are present. The maximal allowed point-to-point distance is set to 50 cm. Figure 2 and 6 present results, where the consistency of the point cloud has been improved. Figure 3 details the changes in the trajectory's position and orientation. It is an open traverse, thus the changes are mainly at the end of the trajectory. Processing was done on a server featuring four Intel Xeon CPUs E7-4870 with 2.4 GHz (40 cores, 80 threads). The overall computing time for the optimization of the Google data set was 10–12 days (few interruptions).

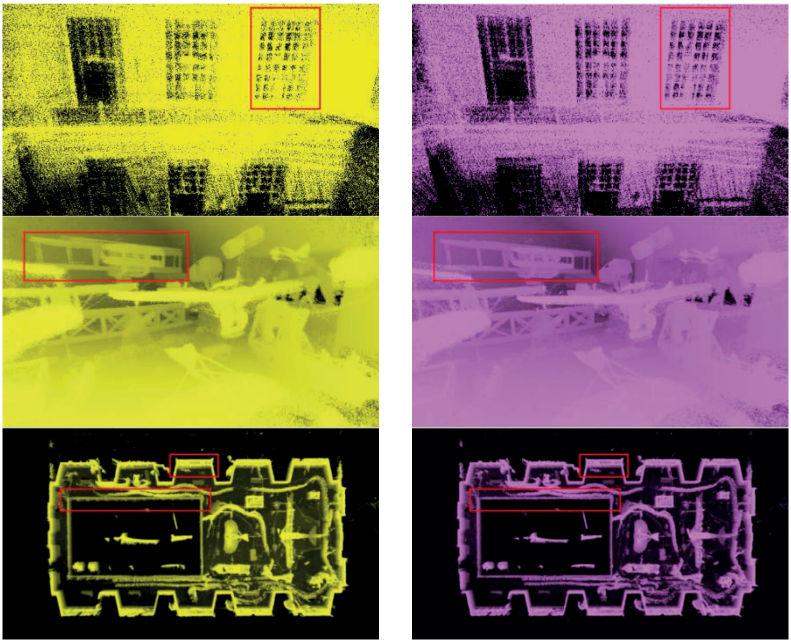

Figure 2. Results of continuous-time simultaneous localization and mapping (SLAM) on Google's Cartographer sample data set from Deutsches Museum in München. Left: input. Right: output of our solution. Shown are three 3D views (perspective) of the scene. Major changes in the point cloud are highlighted in red.

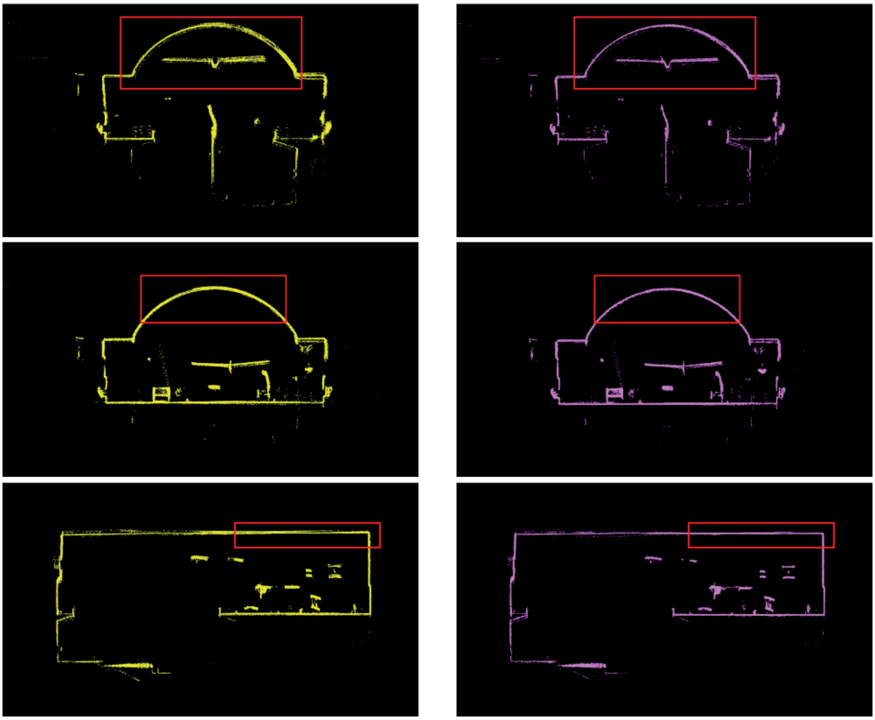

Figure 3. Results of continuous-time SLAM on Google's Cartographer sample data set from Deutsches Museum in München. Left: input. Right: output of our solution. Shown are sections of the point cloud. Major changes in the point cloud are highlighted in red.

In a second experiment, the trajectory from Google has been discretised on a much coarser level in order to process the complete trajectory. The smallest element is now two complete PUCK sweeps. The changes in the trajectory are given in Figure 4. At this level, the trajectory is adjusted at a coarser scale, i.e., the changes of the inner accuracy are smaller, cf. Figure 5.

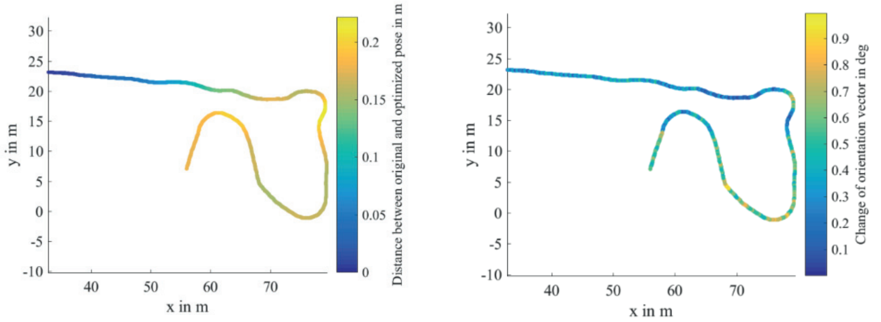

Figure 4. Visualization of the changes in the trajectory computed by our method to bootstrapped trajectory. Left: distance. Right: orientation.

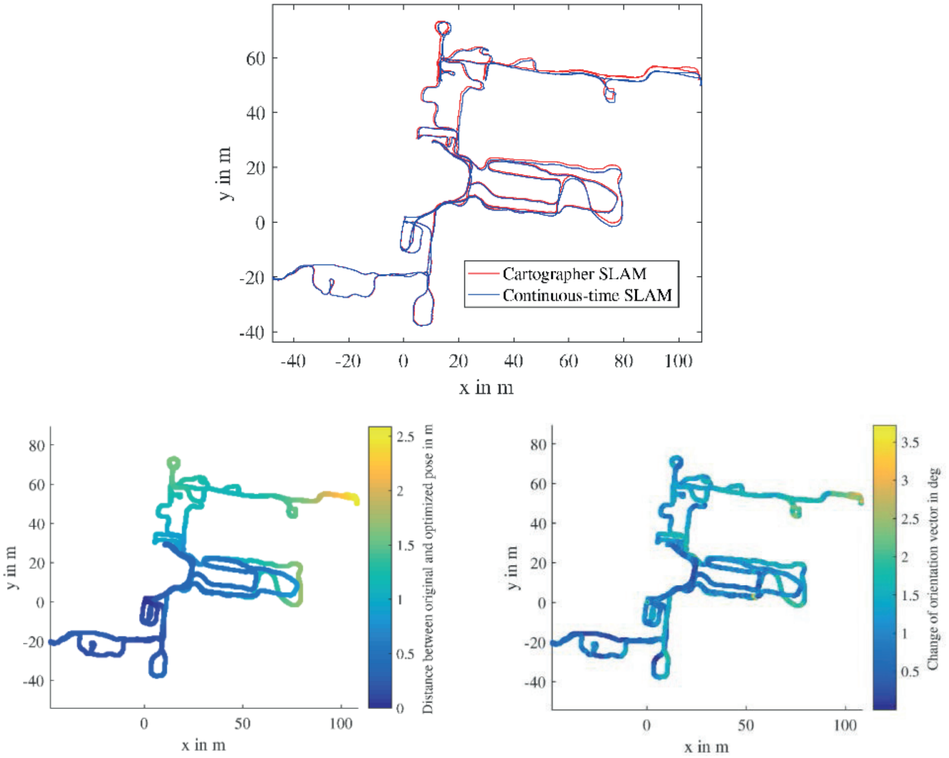

Figure 5. Visualization of the changes in the trajectory over the complete trajectory of the data set. The part presented in Figure 3 is located in the center of the plots.

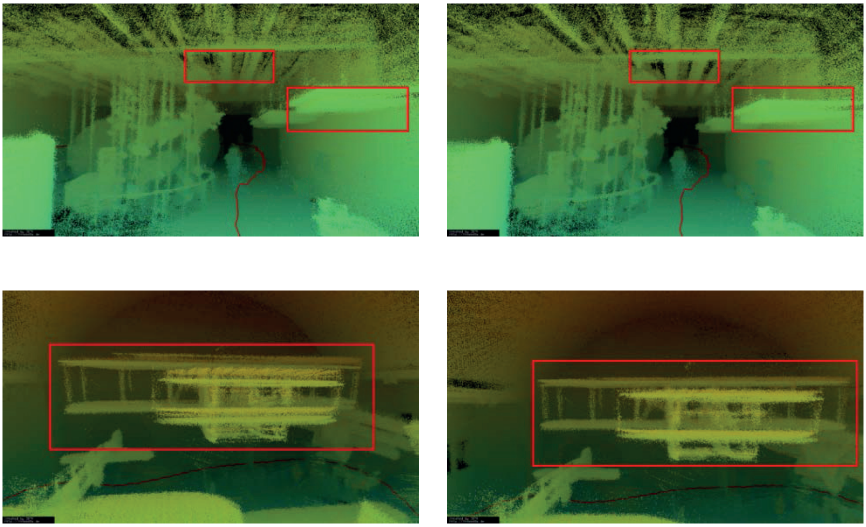

Figure 6. Results of continuous-time SLAM applied to the complete data set. Throughout the data set, the changes are minor, however, in open spaces, the visual impression is still improved.

7. Summary, Conclusions and Future Work

This work revisits a continuous-time SLAM algorithm and its application on Google's Cartographer sample data. The algorithm starts by splitting the trajectory into sections, and matches these sections using the automatic high-precise globally consistent registration of terrestrial 3D scans.

This article has shown how to use the result of Google's novel SLAM solution to bootstrap an existing continuous-time SLAM algorithm. Our approach optimizes the consistency of the global point cloud, i.e. the inner accuracy, and thus improves on Google's results in a local and global fashion. A visual inspection shows the improvements. Personal localization and mapping or personal laser scanning systems will be an emerging research topic in the near future, since the hardware is becoming affordable. The research is also applicable to SLAM systems that work with structured-light scanners such as Microsoft's Kinect, time-of-flight cameras such as the kinectv2, or flash LiDARs (light detection and ranging). An intrinsic challenge remains: How to handle the enormous amount of 3D point cloud data. The time complexity and the memory needs exceed current computing capabilities.

A further conclusion of the article is that the ICP-algorithm with its many variants and extensions is the basic method that is common in the robotics SLAM community and photogrammetry and computer vision community. Bundle adjustment, also called the GraphSLAM method, dominates the mapping methods.

Needless to say, a lot of work remains to be done. First of all, we plan to evaluate the 2D mapping method as we have indicated above. Secondly, as calibration is as crucial as SLAM, we will apply our calibration framework (Elseberg et al., 2013) to the data files provided by Google. Furthermore, we will transfer our continuous-time SLAM to different application areas, e.g., underwater and aerospace mapping applications.

References

1. Agarwal, S.; Mierle, K.; Others. Ceres Solver. Available online: http://ceres-solver.org.
2. Barnea, S.; Filin, S. Keypoint based Autonomous Registration of Terrestrial Laser Point-Clouds. *ISPRS J. Photogramm. Remote Sens.* **2008**, *63*, 19–35.
3. Bendels, G.H.; Degener, P.; Wahl R., Körtgen M.; Klein R. Image-based registration of 3d-range data using feature surface elements. In *The 5th International Symposium on Virtual Reality, Archaeology and Cultural Heritage (VAST 2004)*; Chrysanthou, Y., Cain, K., Silberman, N., Niccolucci, F., Eds.; Eurographics: Lower Saxony, Germany, 2004; pp. 115–124.
4. Benjemaa, R.; Schmitt, F. Fast Global Registration of 3D Sampled Surfaces Using a Multi-Z-Buffer Technique. In Proceedings IEEE International Conference on Recent Advances in 3D Digital Imaging and Modeling (3DIM ’97), Ottawa, Canada, 12–15 May 1997.
5. Benjemaa, R.; Schmitt, F. A Solution for the Registration Of Multiple 3D Point Sets Using Unit Quaternions. In Proceedings of the 5th European Conference on Computer Vision Computer Vision—ECCV ’98, Freiburg, Germany, 2–6 June 1998; Volume 2, pp. 34–50.
6. Bergevin, R.; Soucy, M.; Gagnon, H.; Laurendeau, D. Towards a general multi-view registration technique. *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)* **1996**, *18*, 540–547.
7. Besl, P.; McKay, D. A Method for Registration of 3–D Shapes. *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)* **1992**, *14*, 239–256.
8. Böhm, J.; Becker, S. Automatic marker-free registration of terrestrial laser scans using reflectance features. In Proceedings of 8th Conference on Optical 3D Measurement Techniques, Zurich, Switzerland, 9–13 July 2007; pp. 338–344.
9. Borrmann, D.; Elseberg, J.; Lingemann, K.; Nüchter, A.; Hertzberg, J. Globally consistent 3D mapping with scan matching. *J. Robot. Auton. Syst. (JRAS)* **2008**, *56*, 130–142.
10. Bosse, M.; Zlot, R.; Flick, P. Zebedee: Design of a spring-mounted 3-D range sensor with application to mobile mapping. *IEEE Trans. Robot. (TRO)* **2012**, *28*, 1104–1119.
11. Bosse, M.; Zlot, R. Continuous 3D Scan-Matching with a Spinning 2D Laser. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA ’09), Kobe, Japan, 12–17 May 2009, pp. 4312–4319.
12. Brenner C.; Dold, C.; Ripperda N. Coarse orientation of terrestrial laser scans in urban environments. *ISPRS J. Photogramm. Remote Sens.* **2008**, *63*, 4–18.

13. Chen, G.; Kua, J.; Shum, S.; Naikal, N.; Carlberg M.; Zakhor, A. Indoor Localization Algorithms for a Human-Operated Backpack System. In Proceedings of the International Conference on 3D Data Processing, Visualization, and Transmission (3DPVT '10), Paris, France, 17–20 May 2010.
14. Chen, Y.; Medioni, G. Object Modelling by Registration of Multiple Range Images. In Proceedings of the IEEE Conference on Robotics and Automation (ICRA '91), Sacramento, CA, USA, 9–11 April 1991; pp. 2724–2729.
15. Chen, Y.; Medioni, G. Object Modelling by Registration of Multiple Range Images. *Image Vision Comput.* **1992**, *10*, 145–155.
16. Cunnington, S.; Stoddart, G. N-View Point Set Registration: A Comparison. In Proceedings of the 10th British Machine Vision Conference (BMVC '99), Nottingham, UK, 13–16 September 1999.
17. Davis, T.A. *Direct Methods for Sparse Linear Systems*; SIAM: Philadelphia, PA, USA, 2006.
18. Elseberg, J.; Borrmann, D.; Nüchter, A. Algorithmic solutions for computing accurate maximum likelihood 3D point clouds from mobile laser scanning platforms. *Remote Sens.* **2013**, *5*, 5871–5906.
19. Flint, A.; Dick, A.; van den Hengel, A.J. Thrift: Local 3d structure recognition. In Proceedings of the 9th Biennial Conference of the Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications (DICTA '07), Glenelg, Australia, 3–5 December 2007; pp. 182–188.
20. Hess, W.; Kohler, D.; Rapp H.; Andor D. Real-time loop closure in 2d lidar slam. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '16), Stockholm, Sweden, 16–21 May 2016.
21. Huber, D. Automatic Three-dimensional Modeling from Reality. PhD Thesis, Carnegie Mellon University, Pittsburgh, PA, USA, 2002.
22. Kang, Z.; Li J.; Zhang, L.; Zhao, Q.; Zlatanova, S. Automatic Registration of Terrestrial Laser Scanning Point Clouds using Panoramic Reflectance Images. *Sensors* **2009**, *9*, 2621–2646.
23. Kohlbrecher, S.; Meyer, J.; von Stryk, J.; Klingauf, U. A flexible and scalable slam system with full 3d motion estimation. In Proceedings of the IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR' 11), Kyoto, Japan, 1–5 November 2011.
24. Krishnan, S.; Lee, P.Y.; Moore, J.B.; Venkatasubramanian S. Global Registration of Multiple 3D Point Sets via Optimization on a Manifold. In Proceedings of the Eurographics Symposium on Geometry Processing 2005, Vienna, Austria, 4–6 July 2005.
25. Liang, X.; Kukko, A.; Kaartinen, H.; Hyyppä, J.; Xiaowei, Y.; Jaakkola, A.; Wang, Y. Possibilities of a personal laser scanning system for forest mapping and ecosystem services. *Sensors* **2014**, *14*, 1228–1248.
26. Lu, F.; Milios, E. Robot Pose Estimation in Unknown Environments by Matching 2D Range Scans. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '94), Seattle, WA, USA, 21–23 June 1994; pp. 935–938.

27. Lu F.; Milios, E. Globally Consistent Range Scan Alignment for Environment Mapping. *Auton. Robot.* **1997**, *4*, 333–349.
28. Magnusson, M.; Andreasson, H.; Nüchter, A.; Lilienthal, A.J. Automatic appearance-based loop detection from 3d laser data using the normal distributions transform. *J. Field Robot. (JFR)* **2009**, *26*, 892–914.
29. Nüchter, A.; Borrmann, D.; Koch, P.; Kühn, M.; May, S. A man-portable, imu-free mobile mapping system. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.* **2015**, *II-3/W5*, 17–23.
30. Olson, E. A Passive Solution to the Sensor Synchronization Problem. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '10), St. Louis, MO, USA, 18–22 October 2010; pp. 1059–1064.
31. Pathak, K.; Borrmann, D.; Elseberg, J.; Vaskevicius, N.; Birk, A.; Nüchter, A. Evaluation of the robustness of planar-patches based 3d-registration using marker-based ground-truth in an outdoor urban scenario. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '10), Taipei, Taiwan, 11–15 October, 2010; pp. 5725–5730.
32. Pulli, K. Multiview Registration for Large Data Sets. In Proceedings of the 2nd International Conference on 3D Digital Imaging and Modeling (3DIM '99), Ottawa, Canada, 4–8 October 1999; pp. 160–168.
33. Quigley, M.; Gerkey, B.; Conley, K.; Faust J.; Foote, T.; Leibs, J.; Berger, E.; Wheeler, R.; Ng, A. Ros: an open-source robot operating system. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '09), Kobe, Japan, 12–17 May 2009.
34. Rieger, P.; Studnicka, N.; Pfennigbauer, M. Boresight Alignment Method for Mobile Laser Scanning Systems. *J. Appl. Geodesy (JAG)* **2010**, *4*, 13–21.
35. Rusinkiewicz, S.; Levoy, M. Efficient variants of the ICP algorithm. In Proceedings of the Third International Conference on 3D Digital Imaging and Modellling (3DIM '01), Quebec City, Canada, 19–21 May 2001; pp. 145–152.
36. Saarinen, J.; Mazl, R.; Kulich, M.; Suomela, J.; Preucil, L.; Halme, A. Methods For Personal Localisation And Mapping. In Proceedings of the 5th IFAC symposium on Intelligent Autonomous Vehicles (IAV '04), Lisbon, Portugal, 5–7 July 2004.
37. Sheehan, M.; Harrison, A.; Newman, P. Self-calibration for a 3D Laser. *Int. J. Robot. Res. (IJRR)* **2011**, *31*, 675–687.
38. Skaloud, J.; Schaer, P. Towards Automated LiDAR Boresight Self-calibration. In Proceedings of the 5th International Symposium on Mobile Mapping Technology (MMT '07), Padua, Italy, 25–29 May 2007.
39. Steder, B., Grisetti, G., and Burgard, W., 2010. Robust place recognition for 3D range data based on point features. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '10), Anchorage, AK, USA, 3–7 May 2010; pp. 1400–1405.
40. Stoddart, A.; Hilton, A. Registration of multiple point sets. In Proceedings of the 13th IAPR International Conference on Pattern Recognition, Vienna, Austria, 25–29 August 1996; pp. 40–44.

41. Stoyanov, T.; Lilienthal, A.J. Maximum Likelihood Point Cloud Acquisition from a Mobile Platform. In Proceedings of the IEEE International Conference on Advanced Robotics (ICAR '09), Munich, Germany, 22–26 June 2009; pp. 1–6.
42. Thrun, S.; Fox, D.; Burgard, W. A Real-time Algorithm for Mobile Robot Mapping with Application to Multi Robot and 3D Mapping. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '00), San Francisco, CA, USA, 24–28 April 2000.
43. Underwood, J.P.; Hill, A.; Peynot, T.; Scheding, S.J. Error Modeling and Calibration of Exteroceptive Sensors for Accurate Mapping Applications. *J. Field Robot. (JFR)* **2009**, *27*, 2–20.
44. Wang, X.; Toth, C.; Grejner-Brzezinska, D.; Sun, H. Integration of Terrestrial Laser Scanner for Ground Navigation in GPS-challenged Environments. In Proceedings of the XXIst ISPRS Congress: Commission V, WG 3, Berlin, Germany, 1–2 December 2008; pp. 513–518.
45. Wang, Z.; Brenner, C. Point Based Registration of Terrestrial Laser Data using Intensity and Ge- ometry Features. In Proceedings of the ISPRS Congress ('08), Beijing, China, 3–11 July 2008; pp. 583–590.
46. Weinmann, M.; Jutzi, B. Fully automatic image-based registration of unorganized TLS data. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2011**, *XXXVIII-5/W12*, 55-60.
47. Weinmann, M.; Hinz, S.; Jutzi, B. Fast and automatic image-based registration of TLS data. *ISPRS J. Photogramm. Remote Sens.* **2011**, *66*, 62–70.
48. Williams, J.A.; Bennamoun, M. Multiple View 3D Registration using Statistical Error Models. In Proceedings of the Vision Modeling and Visualization, Erlangen, Germany, November 1999.
49. Zhang, Z. *Iterative Point Matching for Registration of Free–Form Curves*; Technical Report RR-1658; INRI—Sophia Antipolis: Valbonne, France, 1992.

Nüchter, A.; Bleier, M.; Schauer, J.; Janotta, P. Continuous-Time SLAM—Improving Google's Cartographer 3D Mapping. In *Latest Developments in Reality-Based 3D Surveying and Modelling*; Remondino, F., Georgopoulos, A., González-Aguilera, D., Agrafiotis, P., Eds.; MDPI: Basel, Switzerland, 2018; pp. 53–73.