


Article

# Automatic Extrinsic Self-Calibration of Mobile Mapping Systems Based on Geometric 3D Features

Markus Hillemann<sup>1,2,\*</sup>, Martin Weinmann<sup>1</sup>, Markus S. Mueller<sup>1</sup> and Boris Jutzi<sup>1</sup> 

<sup>1</sup> Institute of Photogrammetry and Remote Sensing, Karlsruhe Institute of Technology, 76131 Karlsruhe, Germany

<sup>2</sup> Fraunhofer Institute of Optonics, System Technologies and Image Exploitation IOSB, 76275 Ettlingen, Germany

\* Correspondence: markus.hillemann@kit.edu

Received: 24 July 2019; Accepted: 17 August 2019; Published: 20 August 2019



**Abstract:** Mobile Mapping is an efficient technology to acquire spatial data of the environment. The spatial data is fundamental for applications in crisis management, civil engineering or autonomous driving. The extrinsic calibration of the Mobile Mapping System is a decisive factor that affects the quality of the spatial data. Many existing extrinsic calibration approaches require the use of artificial targets in a time-consuming calibration procedure. Moreover, they are usually designed for a specific combination of sensors and are, thus, not universally applicable. We introduce a novel extrinsic self-calibration algorithm, which is fully automatic and completely data-driven. The fundamental assumption of the self-calibration is that the calibration parameters are estimated the best when the derived point cloud represents the real physical circumstances the best. The cost function we use to evaluate this is based on geometric features which rely on the 3D structure tensor derived from the local neighborhood of each point. We compare different cost functions based on geometric features and a cost function based on the Rényi quadratic entropy to evaluate the suitability for the self-calibration. Furthermore, we perform tests of the self-calibration on synthetic and two different real datasets. The real datasets differ in terms of the environment, the scale and the utilized sensors. We show that the self-calibration is able to extrinsically calibrate Mobile Mapping Systems with different combinations of mapping and pose estimation sensors such as a 2D laser scanner to a Motion Capture System and a 3D laser scanner to a stereo camera and ORB-SLAM2. For the first dataset, the parameters estimated by our self-calibration lead to a more accurate point cloud than two comparative approaches. For the second dataset, which has been acquired via a vehicle-based mobile mapping, our self-calibration achieves comparable results to a manually refined reference calibration, while it is universally applicable and fully automated.

**Keywords:** mobile mapping; laser scanning; self-calibration; 3D point clouds; geometric features

## 1. Introduction

Mobile Mapping is an efficient technology to acquire spatial data of the environment. The spatial data is further processed to 3D city models, building models or models of indoor environments which are nowadays an essential data source for applications in crisis management, civil engineering or autonomous driving. Depending on the scale of the environment, mobile platforms like airplanes, cars, Unmanned Aerial Vehicles (UAVs) or mapping backpacks are considered. The mobile platform is typically equipped with one or more mapping sensors. A widely utilized mapping sensor is a laser scanner, also known as Light Detection And Ranging (LiDAR) sensor, since it acquires accurate and dense spatial data of the environment in form of a 3D point cloud.

The quality of the 3D point cloud captured with a Mobile Mapping System (MMS) is limited by the accuracy of the mapping sensor itself and mainly three more components: The estimation of the *pose* of the MMS, the *intrinsic calibration* of the individual sensors and the *extrinsic calibration* of the MMS.

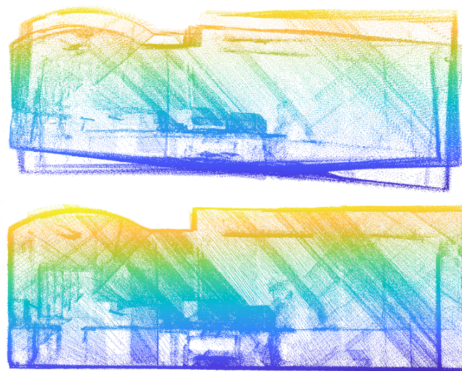
The *pose* describes the position and orientation of the MMS with respect to a superordinate coordinate frame. It is used to register a single scan to a common point cloud. The pose is estimated in a specific frame which we call the navigation frame. The pose estimation task can be solved by using a technology that observes the sensor system, like for example the Global Navigation Satellite System (GNSS), or by using on-board sensors and an odometry- or Simultaneous Localization and Mapping (SLAM)-algorithm.

The *intrinsic calibration* is the process of estimating the interior calibration parameters of the individual sensors. In case of a laser scanner, these are for example a range finder offset and beam direction influences. As the impact of the interior calibration parameters on the accuracy of the point cloud is typically lower than the impact of the other mentioned components, the interior calibration parameters of mapping sensors are often neglected. However, for highest accuracy requirements, the interior calibration parameters must be taken into account.

The objective of the *extrinsic calibration* of a MMS is to find a rigid transformation from the navigation frame to the frame of the mapping sensor. The extrinsic calibration is also known as determining the *relative pose* between the two sensors.

For the sake of clarity, in this work we focus on estimating the extrinsic calibration parameters. For simplicity, we will call this process the calibration from now on.

Figure 1 visualizes the effect of an inaccurate calibration on the basis of a point cloud resulting from Mobile Mapping within an indoor scenario. Due to the use of a line scanning device as mapping sensor and heterogeneous movements of the sensors, the points are not uniformly sampled. The point cloud on top is generated using an insufficient calibration with an error of five degrees in the pitch angle. This leads to a systematically distorted point cloud. In contrast, the point cloud at the bottom is calibrated using a target calibration approach for cameras and laser scanners [1–3].



**Figure 1.** Point cloud as a result of a mobile mapping in an indoor environment. Due to the use of a line scanning device and non-uniform movements of the mobile mapping platform, the points are inhomogeneously sampled. The color of each point illustrates its height. Top: Inaccurate guess of the extrinsic calibration parameters. Bottom: Calibrated using the Robust Automatic Detection in Laser Of Calibration Chessboards (RADLOCC) [1–3], which is a target calibration approach for cameras and laser scanners.

In general, the calibration approaches can be arranged into three categories: (i) Calibration approaches which are performed in the laboratory, (ii) calibration approaches which use specific, artificial targets and (iii) self-calibrations. Self-calibrations are the most practical and time-saving approaches, as it is possible to calibrate the system based on the data that shall be collected anyways. There is no need to prepare the environment before data collection, to design artificial targets or to

use additional sensors in self-calibration processes. Moreover, self-calibrations can be more accurate because changes of the calibration parameters during data collection can be taken into account.

In this paper, we propose a self-calibration approach which is based on geometric features. Therefore, we also provide a brief introduction to these features.

### 1.1. Geometric Features

Geometric features are often used to automatically analyze 3D point clouds. The most popular task in this context is classification [4,5], but geometric features have also been used for coarsely registering 3D point clouds [6] and retrieving objects in 3D point clouds [7]. Particularly, the geometric features derived from eigenvalues of the 3D structure tensor have proven to be descriptive [4]. As these features represent a specific property of the local neighborhood by a single value, they are rather intuitive. The three features of linearity, planarity and sphericity enable for example to quickly identify the primary dimensionality within a local neighborhood.

In this paper, we use such geometric features to evaluate the quality of a point cloud and thus to evaluate the calibration of a MMS with a laser scanner and a pose estimation sensor.

### 1.2. Contributions and Structure

With this paper, we introduce a novel extrinsic self-calibration approach for Mobile Mapping Systems. In this context, we also analyze the suitability of different cost functions for the extrinsic self-calibration task. The MATLAB code of our self-calibration framework is publicly available (<https://github.com/markushillemann/FeatCalibr>).

Our self-calibration approach combines all of the following advantages simultaneously: (i) the estimation of all extrinsic calibration parameters between an arbitrary pose estimation sensor and a mapping sensor, (ii) a universal solution suitable for different kinds of mapping sensors (e.g., 2D and 3D laser scanners), (iii) no assumptions about the environment and no requirements for artificial targets or additional sensors, (iv) a solution robust to initial calibration errors due to a multi-scale approach.

The paper is organized as follows: Section 2 provides an overview of related work in the field of extrinsic self-calibration of MMSs and in the field of geometric point cloud features. The methodology of our calibration approach, which is based on a nonlinear least squares optimization of the calibration parameters, is presented in Section 3. Section 4 describes the synthetic and real datasets and the experiments that have been performed on these datasets. We examine if some of the potentially usable geometric features are better suited for the use as cost function for our self-calibration than others and compare the results to a self-calibration which uses a cost function from the literature [8]. Furthermore, we focus on questions about the applicability to different sensors and environments and prove the suitability to real sensor data. In Section 5, we elaborate on the results of the experiments and derive under which conditions the self-calibration achieves good results. Finally, Section 6 provides concluding remarks as well as an outlook on future work.

## 2. Related Work

As the use of geometric features in terms of a self-calibration is rare, we split the related work section. Section 2.1 presents the related work in terms of relevant calibration approaches. These approaches usually distinguish between two types of mapping sensors: 2D and 3D laser scanners. 2D laser scanners measure the range to the illuminated surface while scanning the surface linearly. Therefore, 2D laser scanners are also known as line laser scanners. The scanning is usually performed sequentially, or more rarely synchronously, by utilizing linearly arranged fiber optics. An example of a 2D laser scanner is the *Hokuyo UTM-30LX-EW* (Hokuyo Automatic Co., Ltd., Osaka, Japan). 3D laser scanners are in principle 2D laser scanners combined with a rotating platform for scanning perpendicular to the previous mentioned scanning direction. An example of a 3D laser scanner is the *Velodyne HDL-64E* (Velodyne Lidar, Inc., San Jose, CA 95138 USA).

The literature offers many calibration approaches which require a 3D laser scanner. As our approach is also suited to estimate the extrinsic calibration parameters of 2D laser scanners and as this is the more restrictive sensor, we focus this section on approaches that can handle both types of sensors. As geometric features are a fundamental component of our calibration approach, we also present the related work of these features in Section 2.2. Most of the previous works on geometric features are linked to classification tasks.

### 2.1. Extrinsic Calibration of Mobile Mapping Systems

Many previous works handle the extrinsic calibration of MMSs with specific combination of sensors. Especially for the extrinsic calibration of MMSs with a laser scanner and a camera, numerous approaches have been presented in literature. There are target calibrations that require special calibration objects like triangles, folding patterns, cubes or more complex calibration objects consisting of multiple connected planes [9–15] and widely used approaches with the need of planar checkerboards [1–3,16–18]. The latter approaches are practical because the utilized checkerboards can also be used to perform the intrinsic calibration of the camera. All of these methods are designed on the idea of minimizing the distance of points to the surface of the calibration target. The calibration parameters can be carried out based on corresponding observations of the two sensors. However, these approaches work exclusively with the specific combination of sensors from a laser scanner and a camera. Moreover, they require an at least partially overlapping field of view of the laser scanner and the camera. Calibration methods which work with arbitrary pose estimation sensors and which are independent of the position and orientation of the sensors with respect to each other are rarely addressed in the literature.

Using GNSS/INS for pose estimation and a laser scanner as mapping sensor, it is not possible to exploit corresponding observations. This sensor combination is typical for vehicle-based mobile mapping. The GNSS/INS system is utilized for the registration and georeferencing of the acquired point clouds. Using this sensor combination for a calibration is equivalent to using an arbitrary six degrees of freedom (6-DOF) pose estimation sensor. Thus, calibration methods that are similar to the presented method are often connected to these fields of use.

Several calibration approaches exist that consider artificial targets. These artificial targets can be ground control points signalized with retro-reflecting material [19], reference geometries like planar surfaces [20,21] or more special geometries like a half-circle surface [22]. The artificial targets must either be arranged in a specific way [20], or the positions or even the 6-DOF poses have to be provided [21]. Moreover, the artificial targets have to be identified in the point cloud which can be time-consuming and error-prone.

Closer to our work is the *rigorous approach* [23]. It is, like our approach, not restricted to GNSS/INS-derived pose but adaptable to any kind of 6-DOF pose estimation sensor. The basic assumption of the rigorous approach is that the point cloud contains groups of points which can be conditioned to lie on surfaces of a known form, particularly on a plane. The position and orientation of these planes are estimated together with the rotation parameters of the extrinsic calibration and one parameter for a constant range finder offset in a Gauss-Helmert adjustment model. The residual function they use is, straightforwardly, formulated as the sum of the distances of each selected point to its corresponding plane. In contrast to our approach, the translation parameters of the extrinsic calibration are assumed to be known and must therefore be determined separately. Moreover, the form of the surfaces has to be determined and adapted manually in the algorithm if the calibration environment does not fit the model assumption. Finally, the identification of the points lying on the specific surfaces is an additional, time-consuming processing step.

The *quasi-rigorous approach* [24] is another choice for extrinsic calibration that does not rely on artificial targets. This approach requires laser scans and associated poses of the sensor system, as well as initial calibration parameters. The quasi-rigorous approach uses at least two overlapping laser scans to optimize the initial calibration parameters. Similar to the approaches described so far,

the quasi-rigorous approach minimizes the sum of distances of a point from the first laser scan to a corresponding plane from the second laser scan, i.e., a point-to-plane metric is used. This corresponding plane is a triangular patch which is derived from a Triangulated Irregular Network (TIN) generation of the second laser scan. Consequently, there is no need to use artificial targets. In contrast to the quasi-rigorous approach, we do not use a point-to-plane metric for minimization, but geometric 3D features. Registration approaches based on Iterative Closest Point (ICP) [25,26] are known for not being robust to initial misalignments. Applied to the calibration, this means that the initial calibration parameters already have to be known very precisely so that the optimization converges to the global minimum.

Recently, the quasi-rigorous approach has successfully been used for UAV-based mobile mapping [27]. As it is important for the interpretation of the results from our real data experiments in Section 4.2.2, we briefly summarize some details of this work: For a typical UAV-based mobile mapping, it is possible to robustly estimate five of the six parameters of the calibration solely based on the flight data. Therefore, the scene has to have a horizontal and vertical expansion and the scene has to be flown over several times in different directions and altitudes. However, the vertical component of the translation, can only be estimated using vertical control, i.e., in form of horizontal planar patches with known altitude. For a more detailed discussion about the minimal requirements on the poses or the environment in context of calibration, the reader is referred to the quasi-rigorous approach [27].

A self-calibration approach that has no assumptions about the environment has been presented by [8]. Instead, a cost function which is based on the Rényi quadratic entropy (*RQE*) is minimized to optimize initial calibration parameters. The cost function is defined as

$$f_{RQE} = - \sum_{i=1}^N \sum_{j=1}^N G(X_i - X_j, \Sigma), \quad (1)$$

where  $X_i, X_j \in \mathbb{R}^3$  are points of the point cloud,  $N$  is the number of points and  $G(\mu, \Sigma)$  is a Gaussian function with mean  $\mu$  and covariance  $\Sigma$ . A derivation of the cost function is not repeated here and the reader is referred to [8]. For the covariance  $\Sigma = \sigma^2 I$  the authors use an isotropic kernel  $I$  and the same variance  $\sigma^2$  for each dimension. This reduces the tuning parameters of the cost function to a single parameter  $\sigma$ . It is suggested to choose this tuning parameter close to the expected accuracy of a captured 3D point. The *RQE* can be interpreted as a measure of the “crispness” of the point cloud and depends on pairwise point distances and a tuning parameter  $\sigma$ . The authors calibrate a system consisting of three 2D laser scanners mounted on a rotating platform.

The self-calibration based on the *RQE* has also been used to calibrate vehicle-based MMSs [28]. In a first step, the authors extrinsically calibrate a MMS with a 3D laser scanner and a GNSS/INS unit. In a second step, the resulting point cloud is exploited to calibrate additional 2D laser scanners. This is done by maximizing the similarity between the point cloud resulting from the first step and the point cloud from the 2D laser scanners. To evaluate the similarity between these two point clouds, they use the Jensen-Rényi Divergence [29]. In essence, this is equivalent to finding the entropy contribution of the cross-terms between the two point clouds. This approach has also been used to calibrate the *Oxford RobotCar* which also includes 3D laser scanners, 2D laser scanners and a GNSS/INS unit [30].

The main advantage of the self-calibration based on the *RQE* [8] is that it has no assumptions about the environment. In contrast, other approaches assume for example planes at specific positions, the existence of surfaces of a known form or a 3D model of the environment. Consequently, it is also more practical than the other approaches. Our self-calibration approach shares these advantages. In contrast to the self-calibration based on the *RQE*, we use different cost functions for optimization and evaluate their suitability for self-calibration. We use a neighborhood with a fixed size for calculation of the cost function instead of taking all point pairs into account which reduces the computational complexity. Moreover, we present results of a multi-scale approach that increases the radius of convergence and improves the accuracy of the self-calibration.

## 2.2. Geometric Features

To describe the local characteristics around a considered point  $\mathbf{X}_i \in \mathbb{R}^3$ , a diversity of geometric features may be utilized. Among these geometric features, particularly the ones derived from the 3D structure tensor  $\mathbf{S}_i \in \mathbb{R}^{3 \times 3}$  are widely used. Thereby, the 3D structure tensor represents a 3D covariance matrix calculated on the basis of the 3D coordinates of those points within the local neighborhood  $\mathcal{N}_i$  of  $\mathbf{X}_i$ . The three eigenvalues of this 3D covariance matrix indicate the extent of a 3D ellipsoid along its principal axes. Denoting the eigenvalues by  $\lambda_{1,i}, \lambda_{2,i}, \lambda_{3,i} \in \mathbb{R}$  with  $\lambda_{1,i} \geq \lambda_{2,i} \geq \lambda_{3,i} \geq 0$ , the following structures may easily be recognized:

- A linear (1D) structure is given for  $\lambda_{1,i} \gg \lambda_{2,i}, \lambda_{3,i}$ , since the respective points in the local neighborhood are mainly spread along one principal axis.
- A planar (2D) structure is given for  $\lambda_{1,i}, \lambda_{2,i} \gg \lambda_{3,i}$ , since the considered points spread within a plane spanned by two principal axes.
- A volumetric (3D) structure is given for  $\lambda_{1,i} \approx \lambda_{2,i} \approx \lambda_{3,i}$ , since the considered points are similarly spread in all directions.

An extension of this has been presented with the analytical consideration of the eigenvalues for describing the local point cloud characteristics with respect to a set of specific shape primitives [31]. Instead of such an analysis with respect to specified parametric models, the local structure around  $\mathbf{X}_i$  may also be described via geometric features derived from the eigenvalues of the 3D structure tensor without assuming a parametric model. In this regard, geometric features that are also known as local 3D shape features [32,33] are commonly used which are rather intuitive and represent a specific property of the local neighborhood by a single value. Thus, these features are interpretable, but recent investigations on the accuracy and robustness of such geometric features reveal that some of them are more susceptible to discretization and noise whereas others are more robust [34].

Other commonly used geometric features are represented by angular characteristics [35], point distances and height differences [36], a variety of low-level geometric 3D and 2D features [4], and moments and height features [5]. Besides the diversity of interpretable geometric features, there are also more sophisticated geometric features like spin image descriptors [37], 3D shape context descriptors [7], Signature of Histograms of Orientations (SHOT) descriptors [38,39], shape distributions [40] or point feature histograms [41]. These are typically derived by sampling specific properties within the local neighborhood of  $\mathbf{X}_i$ , e.g., in form of histograms, so that the resulting feature vectors describing the local structure are of higher dimension and single entries are hardly interpretable.

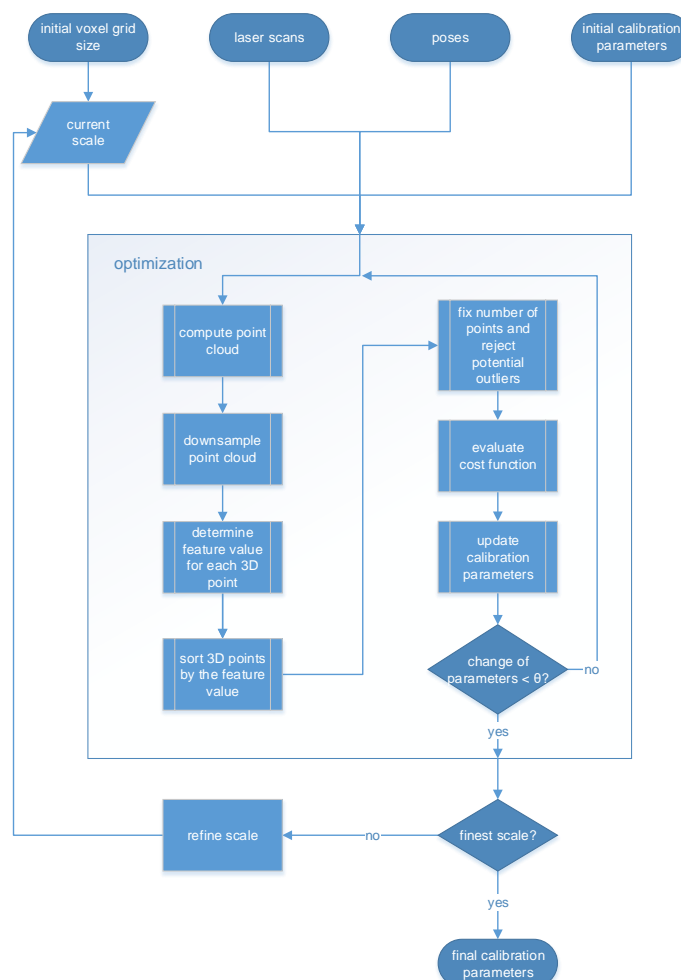
In our work, we assume that self-calibration is performed best when the derived point cloud represents real physical circumstances the best. Such real physical circumstances in turn can be described by interpretable geometric features with a single value. Consequently, we focus on the use of a metric based on interpretable geometric features, also known as local 3D shape features [32,33], to evaluate when real physical circumstances are represented the best.

## 3. Methodology

First, we introduce the principle of the self-calibration approach and give a general overview of our methodology. Then, we present the relevant processing steps in more detail and explain important design choices.

The fundamental assumption of the self-calibration is that the calibration parameters are estimated the best when the derived point cloud represents the real physical circumstances the best and, therefore, when the point cloud has the best quality. To evaluate the quality, we use a cost function which is based on a geometric feature. This geometric feature relies on the 3D structure tensor derived from the local neighborhood of each point. This means that local characteristics of the point cloud are considered which are described by the extent of the neighborhood around each point in three perpendicular directions. By iteratively optimizing the calibration parameters in a nonlinear least squares optimization the cost function is minimized and thereby the point cloud quality is maximized.

Figure 2 shows the workflow of the self-calibration and serves as a general overview of the method. The inputs to the self-calibration are laser scans and corresponding poses of the pose estimation sensor as well as initial calibration parameters and an initial grid size for a voxel grid filter. The basis of the calibration is a nonlinear least squares optimization, that is represented by the box in the middle of Figure 2. Each step inside this box is performed with each iteration of the optimization. First, the initial point cloud is calculated with the given input data. Next, this point cloud is downsampled using a voxel grid filter. For each point and its  $k$  nearest neighbors in the downsampled point cloud, the value of the selected geometric feature is determined. To make the algorithm more robust to outliers, we then filter the points based on the value of the geometric feature. Further, a cost function  $\mathcal{R}$  which is based on the feature values of the remaining points is evaluated. This cost function is minimized in a nonlinear least squares optimization. Consequently, the initial calibration parameters are optimized such that the cost function is minimized. The optimization stops if the change of the calibration parameters is less than a given threshold  $\theta$ . To improve the calibration result and increase the radius of convergence of the approach, we optimize recursively in multiple scales. Thus, we refine the scale by reducing the size of the voxel grid after the optimization has stopped and perform consecutive optimizations. We record the final calibration parameters after the optimization has run through a specified number of scales.



**Figure 2.** Workflow of the self-calibration. Besides the scan points and associated poses, initial calibration parameters and an initial voxel grid size are the inputs to a nonlinear least squares optimization. This optimization is performed recursively with multiple scales to be robust as well as accurate. It estimates the calibration parameters by minimizing an cost function which is based on a selected geometric feature.

### 3.1. Input and Output Parameters

As depicted in Figure 2 there are four categories of input parameters for the self-calibration approach:

1. The first category consists of a single parameter that represents the initial size  $a$  of a voxel grid filter. The size is defined by the edge length of each voxel. Section 3.2.2 gives more details on the downsampling.
2. The second category contains the measurements of the mapping sensor. For each timestep  $t$  the mapping sensor measures a point or a point set  ${}^m\mathbf{X}_t$  in the Cartesian mapping frame  $m$ . The set of point sets for each utilized time step is the input to the self-calibration.
3. The third category contains the measurements of the pose estimation sensor, analogous to the second category. For each timestep  $t$  the pose estimation sensor provides a 6-DOF pose  ${}^w_n\mathbf{M}_t$  that is associated to a single point set. The 6-DOF pose  ${}^w_n\mathbf{M}_t$  represents the rigid transformation from the navigation frame  $n$  to the world frame  $w$ . For this simplified formal description, synchronized sensors and the interpolation of movements during the scanning are assumed.
4. The fourth category consists of the six initial calibration parameters. The initial calibration matrix  ${}^n_m\mathbf{C}$  is a reparametrization of these parameters. In Section 4.1.2 we investigate how accurate the initial calibration parameters must be for the self-calibration to be applicable.

The third category, namely the poses, can best be controlled by the operator of the MMS. Therefore some considerations follow how the MMS should be moved so that the result of the calibration is accurate. As mentioned, local characteristics of the neighborhood around each point are considered to quantify the quality of the point cloud. To ensure that the effects of inaccurate calibration parameters can be quantified with this metric, the same objects must be captured from different angles. However, this does not mean that identical physical points must be captured from different angles. This is one characteristic that distinguishes the presented self-calibration approach fundamentally from procedures that minimize the difference between multiple measurements of ground control points (like for example [19]) or that are based on the direct comparison of two point clouds (like for example [24,27]). In principle, the 6-DOF poses should show large variations for all parameters in a confined space to capture the same objects from different angles.

The above mentioned parameters are the input to a nonlinear least squares optimization. This nonlinear least squared optimization iteratively optimizes the initial calibration parameters. Thereby, the output parameters of the self-calibration approach are optimized calibration parameters of the final calibration matrix  ${}^n_m\hat{\mathbf{C}}$ .

### 3.2. Optimization

As the nonlinear least squares optimization is the basis of the calibration, we describe each step in more detail in this section.

#### 3.2.1. Computation of the Point Cloud $\mathcal{P}$

Given the point set  ${}^m\mathbf{X}_t$  in the Cartesian mapping frame  $m$ . The points  ${}^w\mathbf{X}_t$  in the world frame  $w$  are then computed by following the transformation sequence:

$${}^w\mathbf{X}_t = {}^w_n\mathbf{M}_t {}^n_m\mathbf{C} {}^m\mathbf{X}_t, \quad (2)$$

where  ${}^w_n\mathbf{M}_t$  is the pose at time  $t$  and  ${}^n_m\mathbf{C}$  is the calibration matrix, which we assume to be constant in the selected period of the data acquisition.

Then the point cloud  $\mathcal{P}$  is the set of laser scans in the common world frame:

$$\mathcal{P} = \{{}^w\mathbf{X}_t | t \in \{1, \dots, T\}\}, \quad (3)$$

where  $T$  is the number of time steps or poses and associated laser scans.



### 3.2.2. Downsampling

In principle, when working with scanning devices, the acquired point cloud  $\mathcal{P}$  is inhomogeneously sampled. Close regions show a higher density than far regions. Furthermore, a non-uniform movement of the mobile mapping platform effects the homogeneity of the sampling. To overcome these effects, it is common practice to apply a voxel grid filter  $f_d(x, a)$  [5], where  $x$  is the input point cloud and  $a$  is the edge length of each voxel. After applying the voxel grid filter, the downsampled point cloud  $\mathcal{P}' = f_d(\mathcal{P}, a)$  has a homogeneous sampling. The applied voxel grid filter  $f_d$  approximates all the points in a voxel by the centroid of these points. Thereby, the downsampled point cloud  $\mathcal{P}'$  approximates the geometry of the original point cloud  $\mathcal{P}$  well and even small changes of the geometry are observable. This is important for the iterative calibration approach. In contrast, the approximation of each voxel by a random point within the voxel would be more efficient, but less accurate as small changes would not be observable. The downsampling has two additional benefits: Firstly, it can remove small objects in the point cloud that might disturb the calibration process. This mainly depends on the size of the used voxels and the size of the objects. Secondly, it accelerates the calibration process, because the selected geometric feature has to be computed for less points in coarse scales. However, the downsampling also raises a new challenge: The number of points in the point cloud may vary at each iteration of the least squares optimization. To be able to use a standard optimization algorithm anyway, we need a workaround which is described in Section 3.2.4.

### 3.2.3. Determination of the Geometric Feature

Given the spatial 3D coordinates corresponding to all points within the local neighborhood  $\mathcal{N}_i$  of a considered point  $\mathbf{X}_i$ , we calculate the 3D structure tensor and its eigenvalues (cf. Section 2.2). Thereby, we define the neighborhood  $\mathcal{N}_i$  as the 50 neighbor points with the smallest Euclidean distance to the considered point  $\mathbf{X}_i$ . The eigenvalues are used to derive a set of geometric features which comprises the features of linearity  $f_{L,i}$ , planarity  $f_{P,i}$ , sphericity  $f_{S,i}$ , omnivariance  $f_{O,i}$ , anisotropy  $f_{A,i}$ , eigenentropy  $f_{E,i}$  and change of curvature  $f_{C,i}$  [32,33]. These geometric features are defined as follows:

$$f_{L,i}(\mathbf{X}_i, \mathcal{N}_i) = \frac{\lambda_{1,i} - \lambda_{2,i}}{\lambda_{1,i}} \quad (4)$$

$$f_{P,i}(\mathbf{X}_i, \mathcal{N}_i) = \frac{\lambda_{2,i} - \lambda_{3,i}}{\lambda_{1,i}} \quad (5)$$

$$f_{S,i}(\mathbf{X}_i, \mathcal{N}_i) = \frac{\lambda_{3,i}}{\lambda_{1,i}} \quad (6)$$

$$f_{O,i}(\mathbf{X}_i, \mathcal{N}_i) = \sqrt[3]{\lambda_{1,i}\lambda_{2,i}\lambda_{3,i}} \quad (7)$$

$$f_{A,i}(\mathbf{X}_i, \mathcal{N}_i) = \frac{\lambda_{1,i} - \lambda_{3,i}}{\lambda_{1,i}} \quad (8)$$

$$f_{E,i}(\mathbf{X}_i, \mathcal{N}_i) = -\sum_{j=1}^3 \lambda_{j,i} \ln(\lambda_{j,i}) \quad (9)$$

$$f_{C,i}(\mathbf{X}_i, \mathcal{N}_i) = \frac{\lambda_{3,i}}{\lambda_{1,i} + \lambda_{2,i} + \lambda_{3,i}} \quad (10)$$

For the special case  $\lambda_1 = \lambda_2 = \lambda_3 = 0$ , the features linearity (4), planarity (5), sphericity (6), anisotropy (8) and change of curvature (10) are not defined. This special case occurs when all points in the considered point set have identical coordinates. In practice there is sensor noise that prevents this.

To determine the calibration parameters, we minimize a cost function which is based on a selected feature. Selecting the feature of linearity, for example, would lead to the assumption that the calibration is estimated best, if linear structures in the point cloud are the sharpest. This would be a restricted assumption because usually linear structures are not predominant in point clouds.

In man-made environments, planar structures like the ground, the ceiling, walls, tables, etc. arise more prevalent. Thus, the feature of planarity would be a more obvious choice. Maximizing the planarity in a point cloud is a similar basic assumption as minimizing the distance of points to their corresponding, detected or estimated planes (cf. Section 2.1).

If continuous surfaces in the point cloud, like lines, planes, cylinders, etc. shall be exploited, the change of curvature, namely the variation of a point along the surface normal [42], may be chosen. So in this case, the assumption would be that the calibration is estimated best when the change of curvature over the whole scene is the smallest. For example, if lines in the point cloud are sharp and points corresponding to a plane have small perpendicular distances to this plane.

However, choosing features like sphericity, omnivariance or eigenentropy is the most generic. By minimizing these features, the sharpness of the cloud is maximized. In contrast to all existing calibration approaches mentioned in Section 2.1 except for the self-calibration based on the RQE [8,28] and the quasi-rigorous approach [24], this approach does not assume the existence of geometries with a specific, predefined form. In other words, selecting one of these features does not imply model assumptions about the environment.

In Section 5.1, we discuss the question of which feature is best suited for the self-calibration in more detail based on simulations (cf. Section 4.1.1) and experiments on real data (cf. Section 4.2.1).

### 3.2.4. Computation of the Cost Function and Parameter Estimation

As mentioned above, we aim to maximize the point cloud quality, by minimizing a cost function  $\mathcal{R}$  which is based on the selected geometric feature. Because some geometric features should have large values for the point cloud to be accurate, and others small values, the cost function  $\mathcal{R}$  depends on the feature selection. In case of the features of sphericity, omnivariance, eigenentropy and change of curvature small values lead to a more accurate point cloud, while in case of the features of linearity, planarity and anisotropy large values lead to a better quality of the point cloud. For unification, we redefine the geometric features as follows:

$$g_{L,i}(\mathbf{X}_i, \mathcal{N}_i) = 1 - f_{L,i}(\mathbf{X}_i, \mathcal{N}_i) \quad (11)$$

$$g_{P,i}(\mathbf{X}_i, \mathcal{N}_i) = 1 - f_{P,i}(\mathbf{X}_i, \mathcal{N}_i) \quad (12)$$

$$g_{S,i}(\mathbf{X}_i, \mathcal{N}_i) = f_{S,i}(\mathbf{X}_i, \mathcal{N}_i) \quad (13)$$

$$g_{O,i}(\mathbf{X}_i, \mathcal{N}_i) = f_{O,i}(\mathbf{X}_i, \mathcal{N}_i) \quad (14)$$

$$g_{A,i}(\mathbf{X}_i, \mathcal{N}_i) = 1 - f_{A,i}(\mathbf{X}_i, \mathcal{N}_i) \quad (15)$$

$$g_{E,i}(\mathbf{X}_i, \mathcal{N}_i) = f_{E,i}(\mathbf{X}_i, \mathcal{N}_i) \quad (16)$$

$$g_{C,i}(\mathbf{X}_i, \mathcal{N}_i) = f_{C,i}(\mathbf{X}_i, \mathcal{N}_i), \quad (17)$$

where  $g_{F,i}(\mathbf{X}_i, \mathcal{N}_i) \in [0, 1]$  is the redefined geometric feature with  $F \in \{L, P, S, O, A, E, C\}$ . As  $1 - f_{A,i}(\mathbf{X}_i, \mathcal{N}_i)$  is the same as  $f_S(\mathbf{X}_i, \mathcal{N}_i)$ , we ignore the feature of anisotropy in the following.

To summarize the processing steps explained so far, the calculation of the cost function to be minimized is described in the following. Given the measured points  ${}^m\mathbf{X}_t$  in the Cartesian mapping frame  $m$ , the poses  ${}^w_n\mathbf{M}_t$  and the initial calibration matrix  ${}^n_m\mathbf{C}$ , the point cloud  $\mathcal{P}$  is determined by inserting Equation (2) in (3):

$$\mathcal{P} = \{ {}^w_n\mathbf{M}_t {}^n_m\mathbf{C} {}^m\mathbf{X}_t \mid t \in \{1, \dots, T\} \}. \quad (18)$$

The calibration matrix  ${}^n_m\mathbf{C}$  is a reparametrization of the six calibration parameters which are optimized. Then, the downsampled point cloud

$$\mathcal{P}' = f_d(\mathcal{P}, a) \quad (19)$$

is calculated. Consequently, the cost function to be minimized is

$$\mathcal{R} = \sum_{i=1}^{N'} (g_{F,i}(\mathbf{X}_i, \mathcal{N}_i))^2, \text{ with } \mathbf{X}_i \in \mathcal{P}', \quad (20)$$

where  $g_{F,i}(\mathbf{X}_i, \mathcal{N}_i)$  is the value of the selected, redefined geometric feature and  $N'$  is the number of points in the downsampled point cloud  $\mathcal{P}'$ .

As mentioned in Section 3.2.2, we need a workaround to ensure that the number of considered points is the same for each iteration of the optimization algorithm. Therefore, we sort the points by their feature value and use a subset with the lower  $\zeta$  percent of the values. We choose a smaller subset for the first scale ( $\zeta_0$ ), because the number of points in the point cloud may vary strongly at the beginning of the optimization, depending on the initial calibration. Usually, the point cloud which results from a low-quality calibration is less compact than from the optimal calibration. This results in a considerably higher number of points in the downsampled point cloud, due to the equidistant nature of the voxel grid filter.

Sorting the points and rejecting potential outliers has another advantage over random rejections for some of the features: It makes the algorithm more robust to environments that do not perfectly fit the assumptions about spatial surface characteristics that some of the features imply. A real environment does not exclusively consist of e.g., planar objects. There is vegetation outdoors and there are houseplants indoors which can lead to large values for the selected redefined feature. So, the calibration methodology has to be robust to scattering objects in the point cloud. In unfavorable situations, the remaining points still contain scattering objects with large feature values. To estimate the parameters robustly anyway, we use an M-estimator [43] in an iteratively re-weighted least squares optimization to down-weight the points with large feature values. We choose the Huber-estimator

$$\rho_k(x) = \begin{cases} \frac{1}{2}x^2 & \text{for } x < k \\ k(|x| - \frac{1}{2}k) & \text{otherwise} \end{cases} \quad (21)$$

where  $k$  is the tuning constant [43].

Thus, the final cost function for the optimization becomes

$$\mathcal{R} = \sum_{i=1}^L \rho_k(g_{F,i}(\mathbf{X}_i, \mathcal{N}_i)), \text{ with } \mathbf{X}_i \in \mathcal{P}', \quad (22)$$

where  $L = \lceil \zeta M \rceil$  is the number of points in the subset with  $\zeta = \zeta_0$  for the first scale and  $\mathbf{X}_i \in \mathcal{P}'$  is a point in the transformed, downsampled point cloud  $\mathcal{P}'$  (c.f. Equations (18)–(20)). However, using the cost function in Equation (20) for the subset of points should be sufficiently robust in most cases.

This cost function is minimized in order to estimate the six parameters of the calibration, which represent the translation and the rotation between the sensors. As we want to express the rotation with the minimal number of parameters, we choose the axis-angle representation as parametrization of the rotational part of the calibration. In contrast to Euler and quaternion representations, this parametrization is well-suited for unconstrained optimization algorithms. The derivatives needed by the optimization are determined numerically by using the difference quotient. For this purpose, the cost function has to be evaluated at least once for each parameter to be estimated. This is the most time-consuming processing step depending on the number of points  $N$  in the point cloud. The iteratively re-weighted least squares optimization should stop as soon as the parameters no longer change significantly. Thus, a threshold on the change of parameters is applied.

### 3.3. Multi-Scale Approach

The sampling of the point cloud has a large impact on the accuracy and robustness of the calibration (cf. Section 4.1.3). If the sampling is very fine and the initial calibration parameters

are inaccurate, the optimization likely converges to a local minimum. If the sampling is coarse, the calibration is more robust, but less accurate. Thus, we decided to choose a multi-scale approach.

We define the scale of the point cloud by the size of the voxel grid, which is used for downsampling. Every time the optimization has converged to a solution, we refine the scale of the point cloud by reducing the size of the voxel grid filter. Through the successive refinement of the point cloud, the optimization takes smaller and smaller areas into account because the geometric features are determined based on a fixed number of neighbors for each scale. For coarse samplings, the optimization roughly minimizes the feature value determined at large-scale structures like the ground and walls. For finer scales, it minimizes the feature value at even smaller structures. Hence, we can achieve the robustness of a coarse scale and the accuracy of a fine scale. The calibration finally ends after a fixed number of scales.

#### 4. Experiments

In Section 4.1 the synthetic datasets and the experiments performed on these datasets are introduced. These experiments focus on analyzing how suitable the different cost functions are for the self-calibration task, how accurate the initial parameters have to be known and how the multi-scale approach performs in comparison with the approach that uses a single scale. Subsequently, Section 4.2 describes the real datasets and the experiments with the calibration performed on these data. These experiments show the applicability of the calibration to real data and different sensor setups.

Additionally, for each dataset, we compare the results of our self-calibration approach to the self-calibration based on the *RQE*. Therefore, we use our workflow and exchange the cost function by the cost function based on Equation (1). The main difference to the cost function presented by [8] is that we use a neighborhood  $\mathcal{N}_i$  for each point in the cloud instead of evaluating the distances from every point to every other point in the point cloud. This is conform to a suggestion of [8] and can be justified by the fact that large distances have a very small impact on the cost function. Due to the use of neighborhoods, the impact of the tuning constant  $\sigma$  decreases. However, to take the tuning parameter  $\sigma$  into account, we always test different  $\sigma$  values from optimistic guesses of the point accuracy up to pessimistic guesses and show the best calibration results.

##### 4.1. Synthetic Data

To generate the synthetic data, we use ROS (<http://www.ros.org/>) (Robot Operating System) and Gazebo (<http://gazebo.org/>). With this software it is possible to simulate arbitrary environments, sensors, sensor configurations and trajectories. Our synthetic application area is a closed room with an extent of 10 m  $\times$  10 m  $\times$  5 m. We simulate a 2D laser scanner just like the *Hokuyo UTM-30LX-EW*, as this is one of the most frequently used mapping sensors for reconstruction tasks in small-scale environments. This laser scanner has a field-of-view of 270°, scans 1080 points per line and captures data in ranges from 10 cm up to 30 m. The range measurements are simulated without noise for this analysis. The laser scanner is rigidly mounted on a carrier platform. This platform moves to random poses inside of the simulated environment in order to scan the surroundings from different positions and view angles. With this setup, we simulate ten different mobile mappings of the synthetic environment. Each of these mobile mappings is represented by 100 system poses and associated scan lines. Before extrinsically calibrating the MMS, we falsify the true calibration parameters by 5° for each angle (initial rotation parameters) and 5 cm in each direction (initial translation parameters). As we know the exact parameters of the calibration on simulated data, we show directly the deviations from the ground truth. For convenience, we present the results of the calibrations using the following metrics based on the L2-norm  $\|\cdot\|_2$ :

$$\text{Translation error: } \Delta d = \|\hat{\mathbf{t}} - \tilde{\mathbf{t}}\|_2, \quad (23)$$

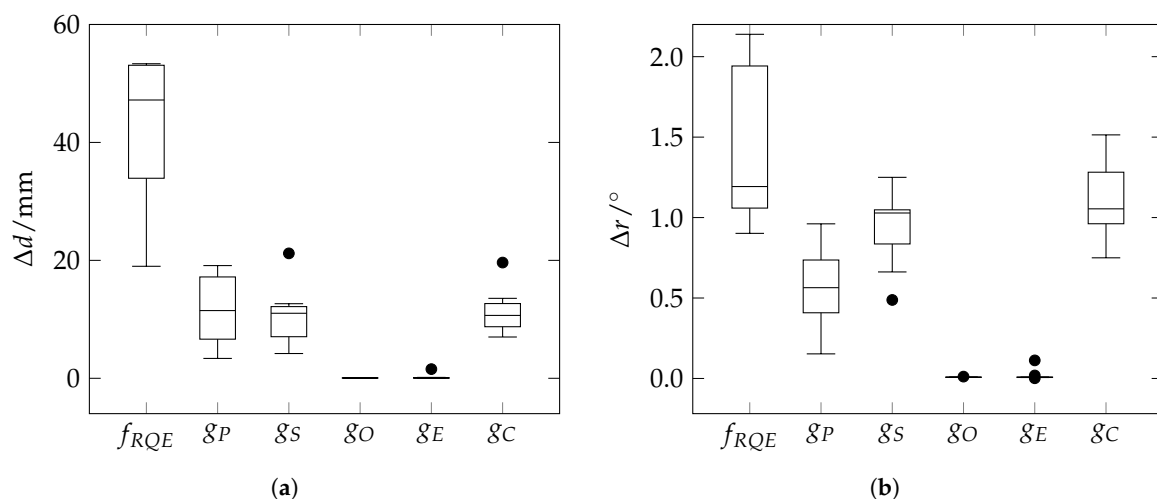
$$\text{Rotation error: } \Delta r = \|\hat{\mathbf{r}} - \tilde{\mathbf{r}}\|_2, \quad (24)$$

where  $\hat{\mathbf{t}} = (\hat{t}_x \ \hat{t}_y \ \hat{t}_z)^T$  is the estimated and  $\tilde{\mathbf{t}} = (\tilde{t}_x \ \tilde{t}_y \ \tilde{t}_z)$  the true translation vector and  $\hat{\mathbf{r}} = (\hat{\phi} \ \hat{\theta} \ \hat{\psi})^T$  is the estimated and  $\tilde{\mathbf{r}} = (\tilde{\phi} \ \tilde{\theta} \ \tilde{\psi})$  the true rotation vector. So,  $\Delta d$  is the deviation of estimated and true distance between the laser scanner and the navigation frame and  $\Delta r$  is the deviation of estimated and true rotation.

#### 4.1.1. Suitability of the Cost Functions

To find out which cost function is best suited for self-calibration, we perform calibrations with each cost function for all ten mobile mappings. Additionally to the presented geometric features, we compare the results to the self-calibration based on the *RQE*. Figure 3 shows the results of the calibrations in terms of translation and rotation errors.

The calibration based on the feature of linearity results in parameters, which are very close to the initial calibration parameters. The corresponding errors are significantly larger than the errors of the calibrations and are not shown for display reasons. The calibrations based on the remaining features result in smaller translation and rotation errors compared to the calibration based on *RQE*. For the features of omnivariance and eigenentropy, the median error is the smallest. To test the suitability of the cost functions under realistic conditions with sensor noise, we compare the results of the calibrations based on the different geometric features again with one of the real datasets in Section 4.2.1.



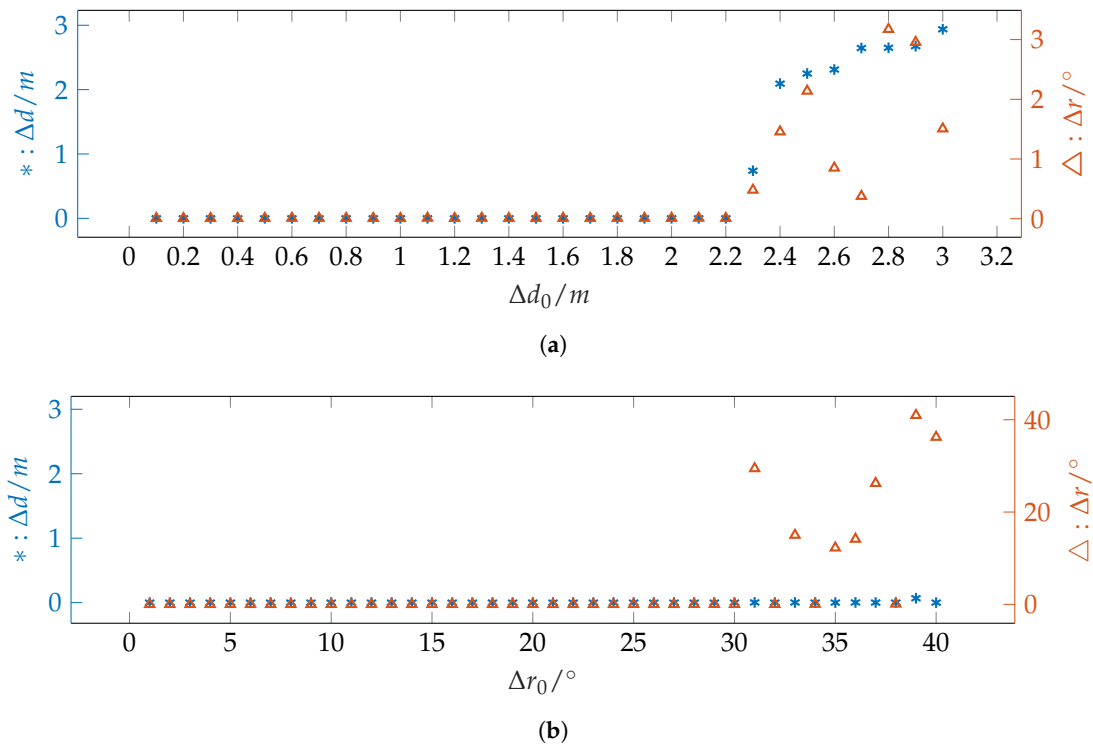
**Figure 3.** Calibration results depending on the  $f_{RQE}$  and the features of planarity ( $g_P$ ), sphericity ( $g_S$ ), omnivariance ( $g_O$ ), eigenentropy ( $g_E$ ) and change of curvature ( $g_C$ ). (a) shows the translation error, (b) the rotation error. The feature of linearity has turned out to be unsuitable for the calibration approach and is not shown for display reasons. The features of omnivariance and eigenentropy clearly outperform  $f_{RQE}$  and all other features.

#### 4.1.2. Radius of Convergence

Each optimization needs reliable initial calibration parameters to converge to the global minimum. An important question for an operator is if it is necessary to measure the initial calibration parameters beforehand with additional sensors or if a rough guess might be accurate enough. In order to estimate the radius of convergence, simulations are carried out with varying initial calibration parameters.

Figure 4 shows the calibration error plotted against the error of the initial calibration parameters. Figure 4a shows the results for a simulated initial translation error  $\Delta d_0$ . For initial translation errors up to 2.2 m the final calibration error is clearly below 1 mm and  $0.01^\circ$ . This means that even with a deviation of two meters from the true distance of the sensors, the calibration is still able to converge close to the global optimum. For larger initial translation errors, the final calibration error considerably increases.

For the results in Figure 4b, an initial rotation error from one to 40° is simulated. The rotation error of the calibrations is below 0.01° for initial rotation errors up to 30°. The translation error up to this mark is below 1 mm. For worse initial rotation parameters, the rotation error clearly grows, since the optimization is then no longer able to find the global minimum. The translation error is small for this simulation, as the initial translation has not been falsified.



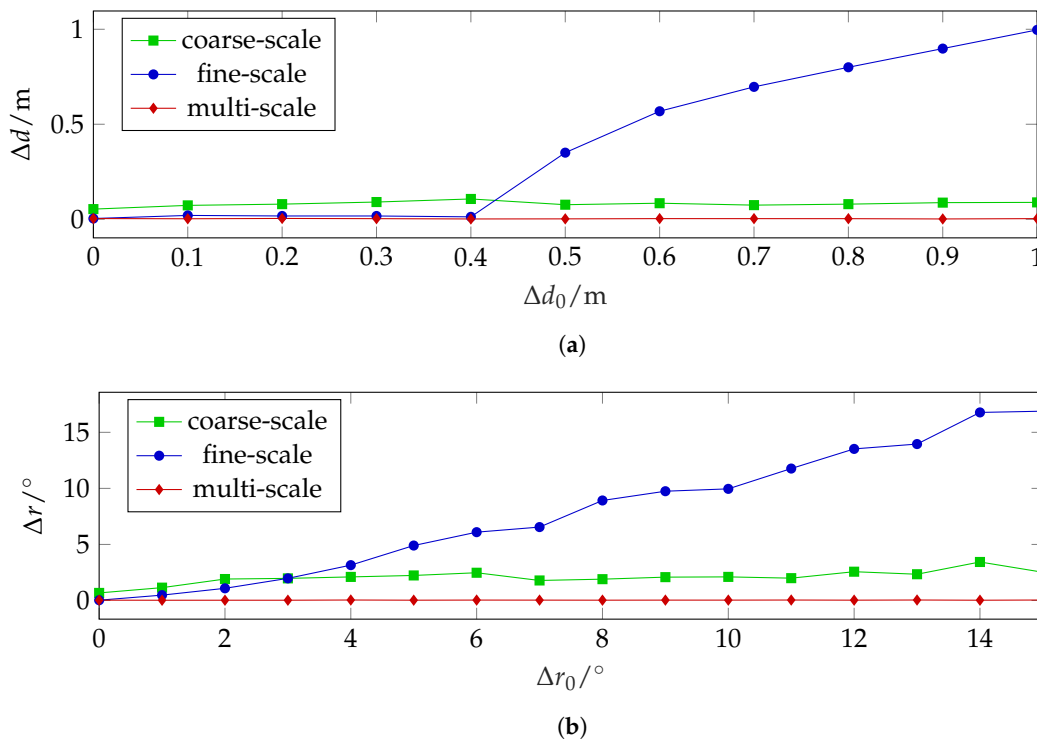
**Figure 4.** Required accuracy of the initial calibration parameters ((a): Initial translation, (b): Initial rotation). Up to an initial translation error of 2.2m or an initial rotation error of 31° the results are in the range of sub-millimeters and sub-degrees. More inaccurate values for the initial calibration parameters mean that the optimization no longer converges to the global minimum and the errors clearly grow.

#### 4.1.3. Single-Scale vs. Multi-Scale

In order to support the statements from Section 3.3, we compare the calibration results based on different scales (see Figure 5) in terms of robustness against initial calibration errors. As mentioned before, the optimization needs good initial calibration parameters to converge to the global minimum. An important question to an operator might be if it is necessary to measure the calibration parameters beforehand with additional sensors or if a rough guess might be accurate enough. In order to estimate the radius of convergence, simulations were carried out with varying initial calibration translation errors  $\Delta d_0$  and varying initial rotation errors  $\Delta r_0$ . Since the feature of omnivariance achieved the best calibration results, this investigation was carried out with this feature.

The coarse-scale and the fine-scale approach do not refine the scale iteratively, but use one fixed scale. The fine scale provides a close to the original but still homogeneous sampling of the point cloud. The multi-scale approach starts with the coarse scale and refines it to the fine scale in two iterations.

For this analysis, we add Gaussian noise to the ranges and to the poses, as we want to test the robustness of the single-scale approaches and the multi-scale approach under more realistic conditions. With a Visual-Inertial SLAM and the laser scanner used in the experiments in mind, we add a noise of 2 cm and 0.1° to the poses and a noise of 3 cm to the ranges. However, adding Gaussian noise to all sensor measurements is still a simplification of the reality. For convenience, we only show the translation error for varying initial translation errors and the rotation error for varying initial rotation errors.



**Figure 5.** Single-scale vs. multi-scale calibration. (a) shows the translation error after performing the calibration depending on the initial translation error. (b) shows the rotation error after performing the calibration depending on the initial rotation error. The coarse-scale approach is more robust to large initial calibration errors than the fine-scale approach. However, the fine-scale approach is more accurate for small initial calibration errors. The multi-scale approach combines the advantages of both single-scale approaches.

The experiments show that the coarse-scale approach is less accurate than the fine-scale approach for small initial calibration errors. However, for large initial calibration errors, it achieves better results compared to the fine-scale approach. The multi-scale approach always achieves the best results.

#### 4.2. Real Data

To test our calibration approach on real data, we use two datasets with different characteristics. The first one is a self-collected dataset, where a hand-held 2D laser scanner is moved in a small-scale indoor laboratory environment. A Motion Capture System (MCS) records the poses. We refer to this dataset as the small-scale indoor dataset. The second dataset is a part of the well-known, publicly available ([http://www.cvlibs.net/datasets/kitti/eval\\_odometry.php](http://www.cvlibs.net/datasets/kitti/eval_odometry.php)) KITTI dataset [44]. This dataset is a large-scale outdoor vehicle-based mobile mapping dataset. It consists of the data of a GNSS/INS unit, a 3D laser scanner and a stereo camera system captured in different environments.

Please note, that we use the same parameters for each dataset. Only the amount of data (poses and scan points) was adjusted for each dataset, in order to limit the runtime of the calibration.

##### 4.2.1. Small-Scale Indoor Dataset

The mapping sensor used for this real-data experiment is the 2D laser scanner *Hokuyo UTM-30LX-EW* with the same specifications as described in Section 4.1. The range measurement accuracy of this laser scanner is specified with  $\pm 30$  mm for ranges up to 10 m and  $\pm 50$  mm for ranges from 10 m to 30 m. The poses are determined with the MCS *Optitrack Prime 17 W*. The MCS tracks the positions of retro-reflective markers by triangulation with sub-millimeter accuracy. If at least three markers are connected rigidly, it is possible to determine the 6-DOF pose of this set of markers which is also called a rigid body. So, the objective of the calibration in this case is to find the

transformation between the mapping frame and the frame of the rigid body which is rigidly attached to the laser scanner.

The environment used for the self-calibration is a laboratory with dimensions of  $8\text{ m} \times 5\text{ m} \times 2.5\text{ m}$ . The room contains objects like cupboards, desks, chairs and monitors. The captured data consists of 2618 different poses and their corresponding scan lines, which results in a point cloud with 2.8 million points.

As it is not possible to provide exact ground truth for the calibration task, we compare our self-calibration qualitatively to a target calibration (TC). This TC is based on the unified intrinsic and extrinsic calibration of a MMS with a multi-camera-system and a laser scanner (UCalMiCel) [18]. The TC is a two-step approach. The first step is based on RADLOCC [1–3] and estimates the transformation between the laser scanner and the camera. The second step is based on accurate measurements of the MCS and determines the transformation between the camera and the rigid body which is rigidly mounted to the camera. For more details about the TC, the reader is referred to the work of [18]. However, since the target calibration is two-step, it has more potential sources of error compared to our self-calibration which directly determines the transformation between the laser scanner and the rigid body. As mentioned, the results are additionally compared to the self-calibration based on *RQE*.

Concerning the initial calibration parameters, we use the parameters which are estimated by the TC and falsify these parameters about  $5^\circ$  for each rotation parameter and 5 cm for each translation parameter.

As mentioned in Section 2.2, some of the features are more robust to discretization and noise than others [34]. Therefore, we additionally verify the simulation results by comparing the various features also on real data. Figures 6 and 7 depict all resulting point clouds. The point clouds in the left column are computed with the parameters resulting from the target calibration. For better comparability these point clouds are colorized based on the feature values used for the associated self-calibrations. This means that the point clouds in the left column do not differ apart from their color display. The point clouds in the right column show the results of the self-calibrations colored by means of the respective feature. All feature values have been scaled to values between zero and one.

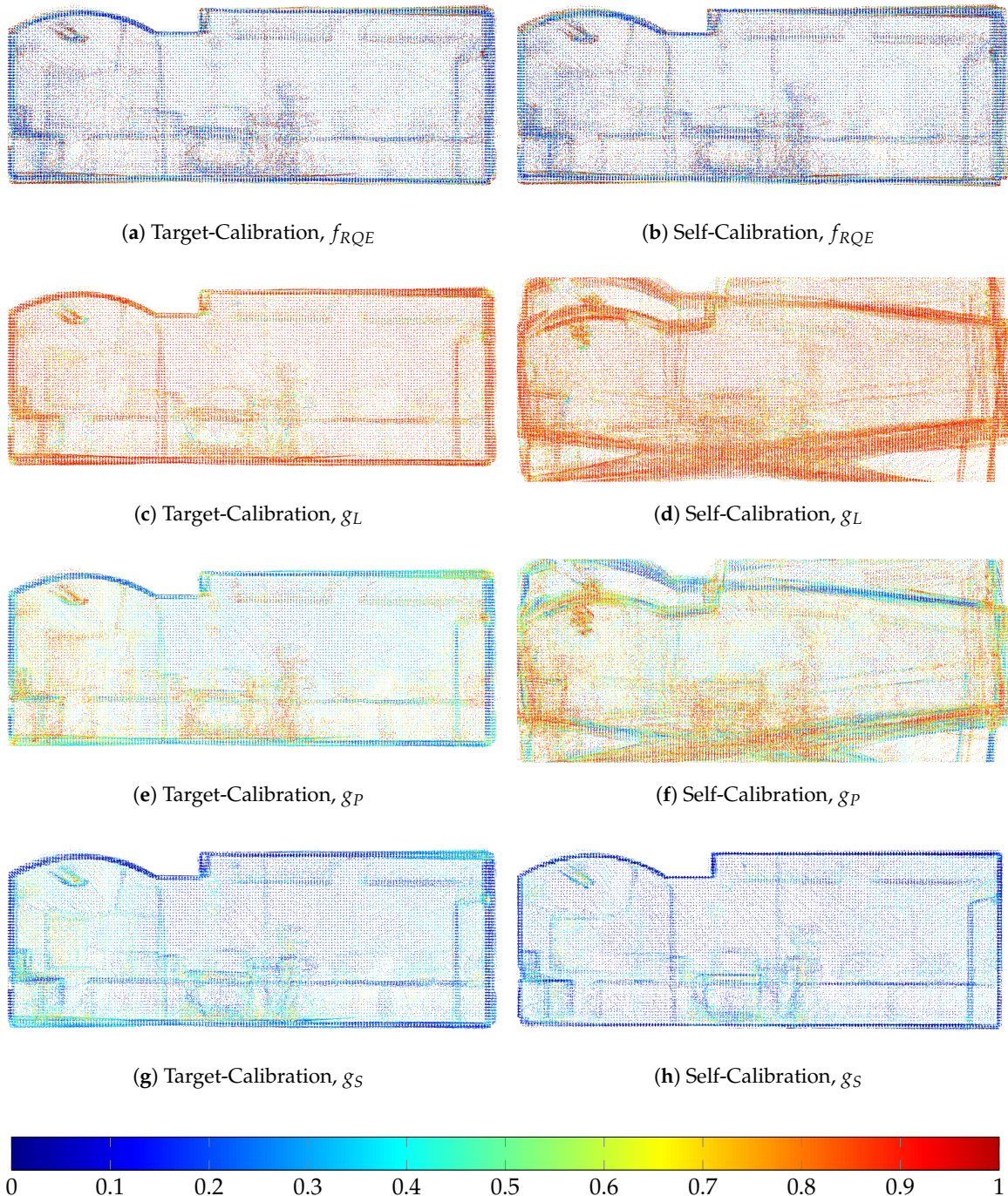
Figure 6d,f show that the parameters estimated by the self-calibration based on the features of linearity and planarity are inaccurate. For these cases, the optimization converges to a local minimum. In all other cases, the point clouds look basically similar to the target calibration, which means that self-calibration is possible with these features. Furthermore, all point clouds resulting from the self-calibrations except for the one in Figure 6b look more compact and sharper than the point cloud based on the TC. This can be seen especially at the top right and bottom left corners of the room. The geometry of the point clouds which result from the calibrations based on the features of omnivariance and eigenentropy look even sharper. However, it is hard to rank the results of the different features based on visual inspection. Thus, we use two additional quantitative metrics to judge the performance of the features.

Firstly, we compare the feature values computed on the point clouds. Table 1 shows the medians of the feature values computed based on the point clouds resulting from different calibrations. Except for the median of feature of linearity, the medians of the feature values are the smallest for the self-calibration based on omnivariance. Furthermore, all self-calibrations except for the calibrations based on linearity and planarity lead to lower values of all features compared to the target calibration and the calibration based on *RQE*. This supports the visual impression of Figures 6 and 7.

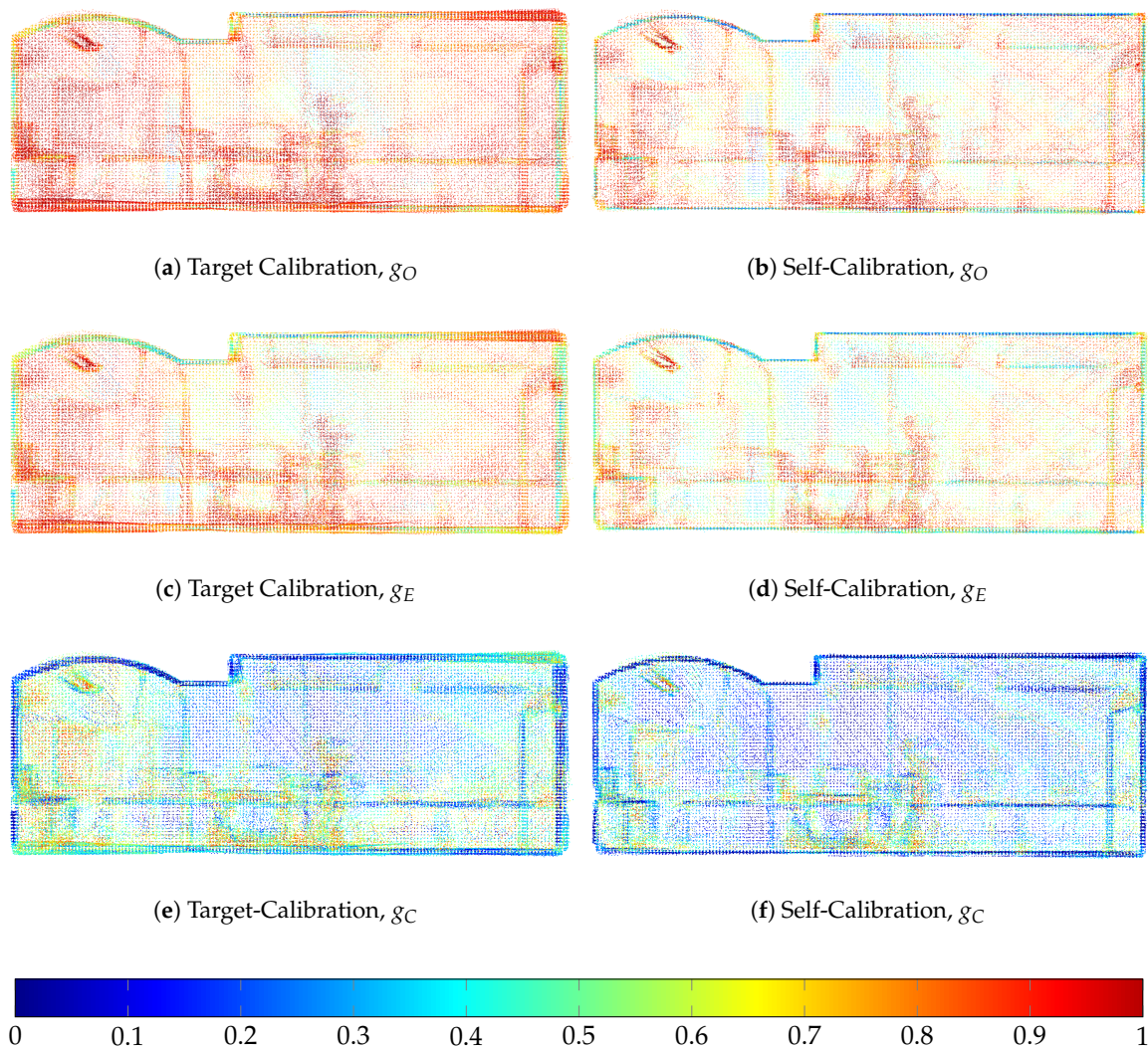
Secondly, we evaluate the calibration results with a further metric which is independent of the geometric features. For this, we consider the mean distances of points to corresponding planes which are fitted to different planar regions of the point clouds. In other words, we evaluate the average thickness of planar structures in the point cloud. We use six planar structures, one on the floor (Plane 1), one on the ceiling (Plane 2) and one at each wall of the room (Planes 3–6). For this purpose, we define regions of interest for the planar structures, fit planes to the points in these regions of interest and



compute the mean point-to-plane distances. Table 2 shows these mean point-to-plane distances. Again, all self-calibrations except for  $g_L$  and  $g_P$  achieve better results as the TC and the calibration based on RQE. For five of the six planes,  $g_O$  achieves the lowest mean point-to-plane distances.



**Figure 6.** Real data results on the small-scale indoor dataset. The self-calibration based on the RQE achieves comparable results to the TC. The self-calibrations based on the features of linearity  $g_L$  and planarity  $g_P$  failed. The self-calibration based on the feature of sphericity  $g_S$ , however, results in a sharper point cloud than the target calibration.



**Figure 7.** Real data results on the small-scale indoor dataset. The self-calibrations based on the features of omnivariance  $g_O$ , eigenentropy  $g_E$  and change of curvature  $g_C$  lead to sharper point clouds compared to the target calibration.

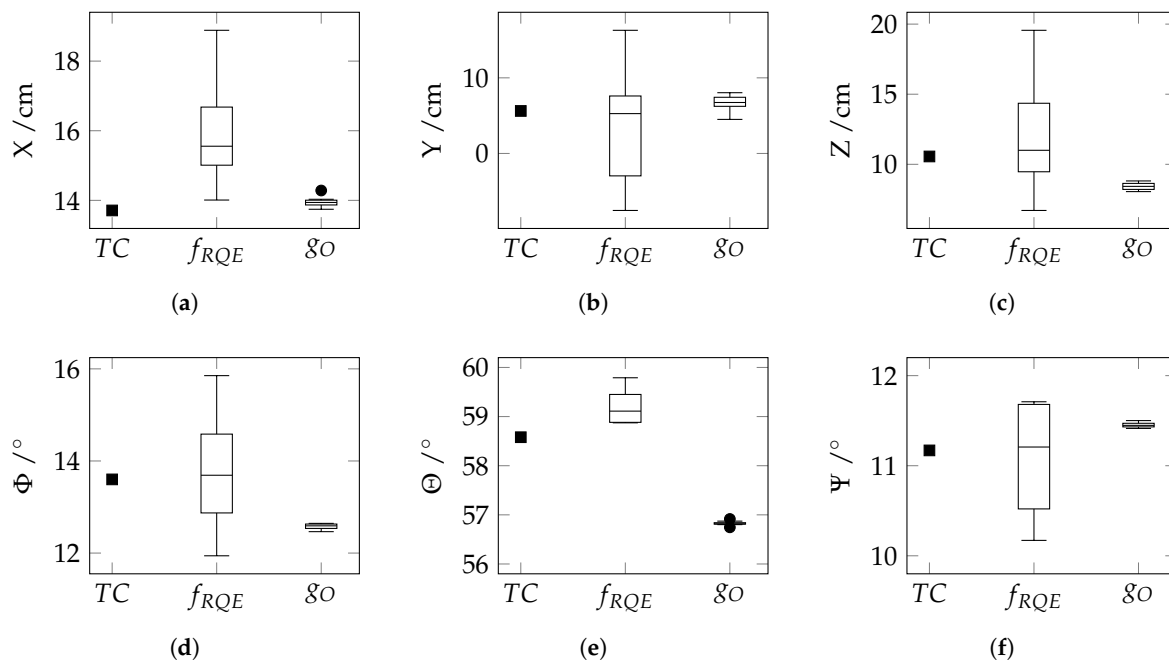
**Table 1.** Comparison of the median of the RQE ( $\text{med}(RQE)$ ) and the redefined features ( $\text{med}(g_L), \dots, \text{med}(g_C)$ ) for different calibration approaches. TC is the target calibration based on RADLOCC.  $RQE$  is the self-calibration based on the RQE.  $g_L, \dots, g_C$  are the self-calibrations based on the corresponding feature. Lower feature values are better. The lowest feature value in each column is marked in bold. X: Calibration failed. The calibration based on the RQE leads to slightly higher values than the TC. The self-calibrations based on  $g_S, g_O, g_E$  and  $g_C$  outperform the TC and the calibration based on  $RQE$ . Omnivariance  $g_O$  achieves the best results.

Approach	$\text{med}(f_{RQE})$	$\text{med}(g_L)$	$\text{med}(g_P)$	$\text{med}(g_S)$	$\text{med}(g_O)$	$\text{med}(g_E)$	$\text{med}(g_C)$
TC	0.161	0.842	0.389	0.215	0.275	0.734	0.106
$f_{RQE}$	0.205	0.845	0.450	0.268	0.288	0.787	0.129
$g_L$	X	X	X	X	X	X	X
$g_P$	X	X	X	X	X	X	X
$g_S$	0.092	<b>0.826</b>	0.281	0.087	0.217	0.546	0.047
$g_O$	<b>0.082</b>	0.830	<b>0.264</b>	<b>0.072</b>	<b>0.205</b>	<b>0.514</b>	<b>0.039</b>
$g_E$	0.088	0.828	0.271	0.079	0.211	0.529	0.042
$g_C$	0.095	<b>0.826</b>	0.283	0.090	0.220	0.552	0.048

**Table 2.** Mean point-to-plane distances for different parts of the point cloud (in mm). X: Calibration failed. The smallest point-to-plane distance in each column is marked in bold. Again, all self-calibrations except for  $g_L$  and  $g_P$  achieve better results than the TC and the calibration based on  $RQE$ . For five of the six planes, the omnivariance  $g_O$  achieves the lowest mean distances.

Approach	Plane 1	Plane 2	Plane 3	Plane 4	Plane 5	Plane 6
TC	14.1	21.7	13.6	15.1	14.5	22.8
$f_{RQE}$	14.3	33.0	17.0	25.0	28.4	26.3
$g_L$	X	X	X	X	X	X
$g_P$	X	X	X	X	X	X
$g_S$	7.1	11.4	9.5	7.5	<b>9.9</b>	6.7
$g_O$	<b>6.3</b>	<b>8.3</b>	<b>7.7</b>	<b>7.4</b>	10.7	<b>6.6</b>
$g_E$	10.2	<b>8.3</b>	8.1	9.3	10.1	8.0
$g_C$	8.6	11.2	7.8	7.9	11.6	9.7

Finally, Figure 8 shows the calibration results of the TC, the results of the self-calibration based on the  $RQE$  and the self-calibration based on the feature of omnivariance for the small-scale indoor dataset. The parameters are separated into their single components instead of using the L2-norm, to visualize whether the precision varies significantly. Figure 8a–c show the translation parameters, Figure 8d–f show the rotation errors. For the TC, only a single set of parameters is available without information about the precision. The medians of multiple calibrations based on the  $RQE$  and also based on the feature of omnivariance differ significantly from the results of the target calibration. The precision of the results based on the feature omnivariance, is higher than the precision of the calibration based on  $RQE$  for each parameter. The self-calibration based on the feature of omnivariance estimates each parameter with a precision of a few millimeters and millidegrees, respectively.



**Figure 8.** Calibration results of the different approaches for the small-scale indoor dataset. (a–c) show the translation parameters, (d–f) show the rotation parameters. For the TC, only a single set of parameters is available without information about the precision. The medians of multiple calibrations differ significantly from the results of the target-calibration. The self-calibration based on omnivariance  $g_O$  estimates all parameters with higher precision compared to the calibration based on  $RQE$ .

#### 4.2.2. Large-Scale Outdoor Dataset

This experiment shows the calibration results on a part of the publicly available KITTI dataset [44]. The KITTI dataset is a large-scale outdoor vehicle-based mobile mapping dataset. The dataset includes the data of a GNSS/INS unit, a 3D laser scanner and a stereo camera system.

To test the calibration approach with a more inaccurate pose estimation sensor compared to the highly accurate MCS which is used for the small-scale indoor dataset, we use the images of the stereo camera system in a visual SLAM framework instead of the highly accurate GNSS/INS unit for this analysis. The visual SLAM framework we use is called ORB-SLAM2 [45]. Thus, in this case, the extrinsic calibration aims to determine the transformation between a 3D laser scanner and a stereo camera system.

The purpose of this experiment is to evaluate the robustness of the proposed extrinsic self-calibration approach in terms of different environments and sensors. The environment mainly differs from the environment in the small-scale indoor data set by larger dimensions of the point cloud, a larger number of points and a larger percentage of vegetation. The sensors differ from those used in the small-scale indoor data set in terms of the operating principle of the pose estimation sensor, the number of scanned lines per rotation of the laser scanner, and the accuracies of both sensors.

The laser scanner is a *Velodyne HDL-64E*. It has a measurement range of 120 m and a range measurement accuracy of 5 cm ( $1\sigma$ ). It has a vertical field of view of  $26.8^\circ$  and a horizontal field of view of  $360^\circ$ . The vertical angular resolution is  $0.4^\circ$  and the horizontal angular resolution is  $0.09^\circ$ . The laser scanner measures up to approximately one million points per second.

Each single camera of the stereo camera system has a  $1/2''$  CCD sensor without color filters, 1.4 megapixel, a pixel size of  $3.75\ \mu\text{m}$  and a global shutter. For the pose estimation by ORB-SLAM2 on the KITTI dataset, we assume a translation accuracy of 1.15% and a rotation accuracy of  $0.0027^\circ$  [46].

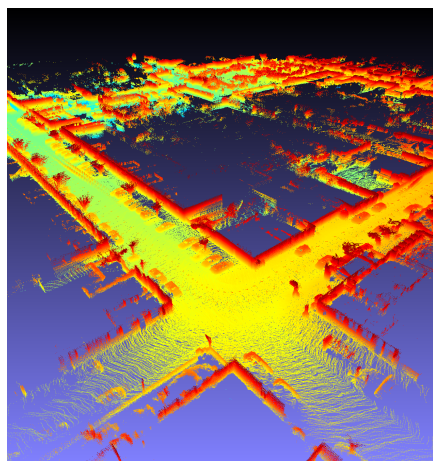
We use the first of 22 sequences of the KITTI dataset. In order to accelerate the calibration process, we use a subset of this sequence and randomly downsample the laser scanner points. However, to ensure an accurate pose-estimation of ORB-SLAM2, we wait for the first loop closure. Consequently, we use every third of the first 1650 poses and associated laser scans.

We compare the estimated parameters against the parameters of the reference calibration (RC), which is provided with the dataset. We falsify the values of the RC about  $5^\circ$  for each rotation angle and 5 cm for each translation parameter and use these values as initial calibration parameters. We compare the RC qualitatively and quantitatively with the optimized parameters after performing the self-calibration.

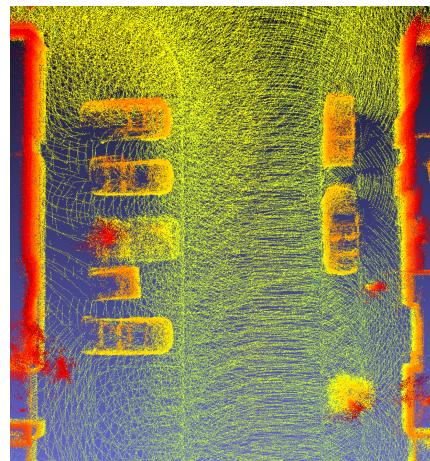
The RC is a two-step semi-automatic procedure. In the first step, coarse calibration parameters are determined based on the automatic single-shot calibration of range and camera sensors utilizing checkerboards [47]. Their algorithm proceeds a segmentation to identify the checkerboard planes in the data of the range sensor. In the following, transformation hypotheses are generated by random plane associations. The best hypotheses are then refined and verified. A final non-maxima suppression step determines all acceptable solutions. In the second step, the result of this automatic calibration is refined manually. Therefore, the number of disparity outliers is minimized jointly with the reprojection error of manually selected correspondences [44].

The accuracy of the calibration results of the automatic step is given in [47]. The median rotation error without Gaussian noise is given with approximately  $10^\circ$  with a 25th percentile of approximately  $1^\circ$  and a 75th percentile of approximately  $18^\circ$ . The median translation accuracy is given with approximately 0.2 m, the 25th percentile with a few centimeters and the 75th percentile with approximately 0.5 m. The accuracy of the manually refined calibration (after the second step) is not mentioned.

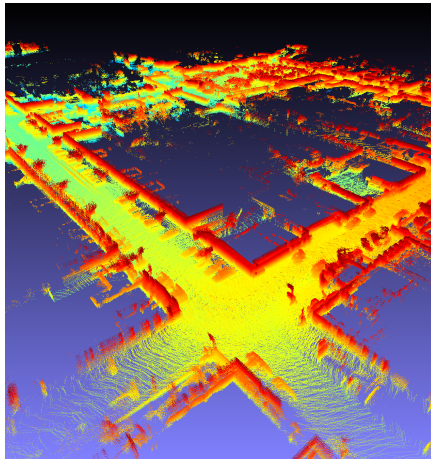
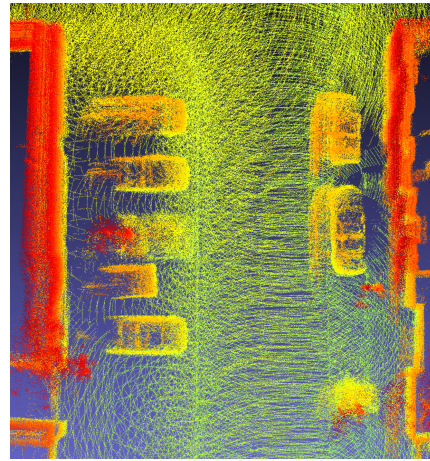
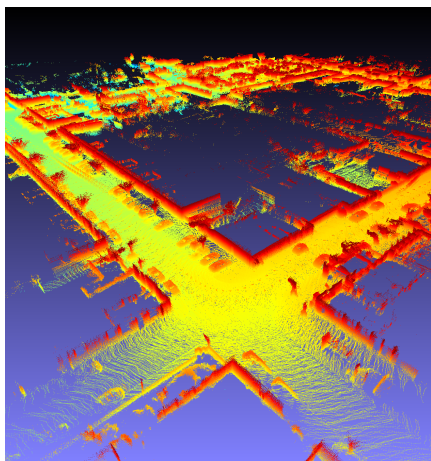
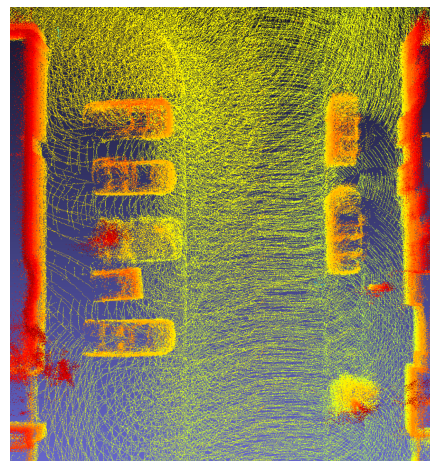
Figure 9 shows the point clouds associated with this experiment. The color of each point represents its height. Figure 9a,c,e show an overview of the point clouds computed with the parameters of the RC, the self-calibration based on the *RQE* and the self-calibration based on the feature of omnivariance, respectively. Figure 9b,d,f show detailed top views of these point clouds.



(a) Slant view, RC



(b) Top view, RC

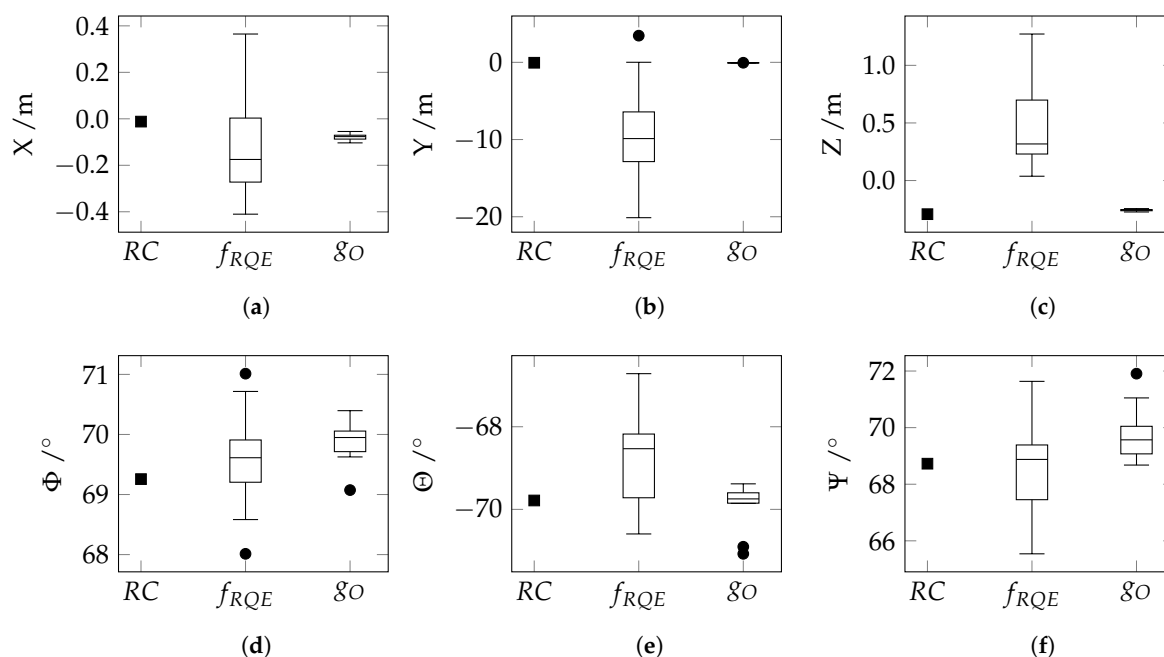
(c) Slant view,  $f_{RQE}$ (d) Top view,  $f_{RQE}$ (e) Slant view,  $g_O$ (f) Top view,  $g_O$ 

**Figure 9.** Results on the publicly available KITTI dataset [44]. The color of each point depicts its height. Both calibrations improved the initial calibration parameters such that objects and basic structures are identifiable in the point clouds. Moreover, the RC and the calibration based on the feature of omnivariance lead to very similar point clouds. The calibration based on the  $f_{RQE}$ , however, results in a more noisy point cloud, which can be seen especially at the vertical walls and at artifacts next to the two cars on the right.

Both calibrations improved the initial calibration parameters significantly such that objects and basic structures are identifiable in the point clouds. The RC and the calibration based on the feature of omnivariance lead to very similar point clouds. They only differ in details but quality differences are difficult to verify based on visual inspection. The calibration based on the  $f_{RQE}$ , however, results in a more noisy point cloud, which can be seen especially at the vertical walls and the artifacts next to the two cars on the right.

Table 3 shows mean point-to-plane distances for different parts of the point cloud in analogy to Section 4.2.1. Planes 1–3 are excerpts from the street level at different positions, Planes 4–6 are excerpts from walls of different buildings. RC is the calibration performed by [44] including the manual refinement.  $f_{RQE}$  is the self-calibration based on the RQE and  $g_O$  is our self-calibration based on the feature of omnivariance. The differences between the mean point-to-plane distances are in the range of a few centimeters for most planes. However, for three out of six planes the RC achieves the best results. For the other three planes the self-calibration based on the feature of omnivariance achieves the best results. This supports the visual impressions of Figure 9 that the results of the RC and  $g_O$  are of similar quality whereas the quality of  $f_{RQE}$  is lower.

Finally, Figure 10 shows the results of the RC, the self-calibration based on the RQE and our self-calibration based on the the feature omnivariance for the large-scale outdoor dataset. Figure 10a–c show the translation parameters and Figure 10d–f shows the rotation parameters. The calibration parameters must be interpreted with respect to the orientation of the navigation frame and the mapping frame. For the large-scale outdoor dataset, the X-axis of the navigation frame points to the right, the Y-axis to the bottom and the Z-axis to the front. For the mapping frame, the X-axis points to the left, the Y-axis to the top and the Z-axis to the front.



**Figure 10.** Calibration results of the reference calibration (RC), the self-calibration based on the RQE and our self-calibration based on the the feature of omnivariance for the large-scale outdoor dataset. (a–c) show the translation parameters, (d–f) show the rotation parameters. The estimation of the vertical component Y of the translation fails for the self-calibration based on RQE. The estimation of the calibration parameters is more precise for the self-calibration based on the omnivariance  $g_O$ , however,  $\Phi$  and  $\Psi$  is estimated with a larger deviation from the reference calibration.

**Table 3.** Mean point-to-plane distances for different parts of the point cloud of the large-scale outdoor dataset (in cm). The smallest point-to-plane distance in each column is marked in bold. For three out of six planes the RC achieves the best results. For the other three planes the self-calibration based on the feature of omnivariance achieves the best results.

Approach	Plane 1	Plane 2	Plane 3	Plane 4	Plane 5	Plane 6
RC	5.4	<b>3.0</b>	5.4	<b>8.3</b>	<b>8.2</b>	16.4
$f_{RQE}$	11.4	10.3	7.9	8.6	10.0	23.1
$\xi_O$	<b>5.0</b>	4.5	<b>4.1</b>	8.7	9.4	<b>13.3</b>

The estimation of the vertical component  $Y$  of the translation fails for the self-calibration based on  $RQE$  as the error is about 10 m for this parameter. Nevertheless, objects like cars are identifiable in the resulting point cloud shown in Figure 9d. The reason is that a calibration error of  $-10$  m for the vertical component causes the entire point cloud to be shifted 10 m upwards without affecting the position of one point relative to another. This point is discussed in more detail in Section 5.4.1. The estimation of the translation parameters is more precise as well as more accurate for the calibration based on the feature of omnivariance.

The result of the rotation parameters is more precise for the self-calibration based on the omnivariance, however,  $\Phi$  and  $\Psi$  are estimated with a larger deviation from the reference calibration. As the self-calibration approach based on the  $RQE$  is not precise, a single calibration is likely to be less accurate than the result shown in Figure 10. However, the median of multiple calibrations shows errors of less than  $1^\circ$  for all rotation parameters.

## 5. Discussion

In the following, we discuss the results of each performed experiment in detail. First, we discuss the results of the investigations concerning the different cost functions, the radius of convergence, and the multi-scale approach. Subsequently, we discuss the experiments with real data in more detail.

### 5.1. Suitability of the Cost Function

The experiments in Section 4 have shown that it is possible to optimize initial calibration parameters with the self-calibration approach and most investigated cost functions. The cost functions based on linearity and planarity have proven to be unsuitable for the self-calibration. However, some of the cost functions lead to significantly better results than others. The experiments show that the cost functions based on the features omnivariance and eigenentropy outperform the other cost functions including the one based on the  $RQE$ . These results could be shown for synthetic as well as for real data.

When comparing Equations (4)–(10), it is noticeable that, in contrast to all other features, no quotient is formed from the eigenvalues for the features omnivariance and eigenentropy. In these cases, no ratio between multiple eigenvalues is taken into account. Instead, the omnivariance becomes small if at least one of the three eigenvalues is small. This ensures that linear and planar structures in the point cloud as well as isolated points lead to small values of the omnivariance. Moreover, a point that belongs to two perpendicular planes like the edge of a room also leads to a small value of the omnivariance. In contrast, structures which cause all three eigenvalues to be considerably larger than zero result in large values of the omnivariance. So disordered structures which result for example from wrong calibrations or from vegetation are penalized. The eigenentropy behaves similar to the omnivariance. However, structures that deviate from an ideal linear or planar structure are penalized more severely. So the feature eigenentropy is supposed to be less suited to calibrate inaccurate sensors with noisy measurements. The omnivariance and the eigenentropy lead to a more generic approach compared to other features and are, thus, supposed to be suitable for more kinds of environments. This is supposed to be the reason that these features are more suitable for self-calibration. Hence, it means that the only restriction is that the environment should not predominantly consist of regions where each eigenvalue of the structure tensor is large like for vegetation. In this case, the cost

function is supposed to have a small gradient such that the optimization likely converges to a local minimum. Another reason might be, that the features omnivariance and eigenentropy have proven to be robust to noise, whereas the features linearity and planarity are very sensitive to noise [34].

Compared to the cost function based on the feature of omnivariance, the cost function based on the *RQE* leads to less accurate and less precise results. The cost function based on the *RQE* represents the sum over the weighted exponentiated distances between all possible points in the specified neighborhood [8]. Smaller distances are weighted higher. The distances are computed along the coordinate axis. The cost is therefore not only dependent on the extent of the considered points, but also on the orientation of the points with respect to the coordinate axes. The omnivariance essentially represents the volume of the 3D ellipsoid around the considered points oriented along its principal axes, since the eigenvalues can be interpreted as the extent of the 3D ellipsoid along the principal axes. Consequently, the extent is not determined along the coordinate axes, but along the principal axes of the considered points. This cost function is therefore independent of the orientation of the considered points with respect to the coordinate axes.

### 5.2. Radius of Convergence

Figure 4 shows that radius of convergence of the self-calibration is with more than 2 m and 30° impressively large. Consequently, the approach is robust against large initial translation and rotation errors. It is therefore sufficient in most cases to roughly estimate the initial translation and initial rotation parameters. Although the distortion of the point cloud increases for large initial calibration errors, the experiments show that approximating the angles between the sensors, measuring it with simple instruments or using the design drawings is accurate enough for the optimization to converge to the global minimum. It is also possible to compute the initial point cloud with Equation (2), tune the parameters manually until basic structures are identifiable and perform the optimization afterwards to obtain accurate calibration parameters.

### 5.3. Single-Scale vs. Multi-Scale

Figure 5a shows, that the fine-scale approach is robust to initial translation errors and achieves accurate results. However, for large initial translation errors, the estimation of the translation parameters fails. The coarse-scale approach is more robust to initial translation errors, but not as accurate as the fine-scale approach. The multi-scale approach achieves the most accurate results and is robust concerning the tested initial translation errors.

In contrast, Figure 5b shows that the fine-scale approach is not very robust to initial rotation errors. The calibration achieves less accurate results than the coarse-scale approach for initial rotation errors of more than 3° in this simulation. Even if the exact numbers of this investigation refer only to the given sensor noise, Figure 5b shows that the coarse-scale approach achieves better calibration results from a certain size of initial rotation error. The multi-scale approach also achieves the most accurate results for the tested initial rotation errors.

The coarse-scale approach is more robust to large initial calibration errors than the fine-scale approach. However, the fine-scale approach, is more accurate for small initial calibration errors. The multi-scale approach combines the advantages of both approaches, which means it is robust as well as accurate, i.e., if the initial rotation parameters are not known in advance, the multi-scale approach should be chosen. The only disadvantage of the multi-scale approach is an increased runtime. However, as the calibration is usually not time-critical, the self-calibration should be performed with multiple scales.

### 5.4. Real data

The real-data experiments show that the self-calibration is in principle applicable to different scenarios such as different sensor combinations (2D laser scanner + MCS and 3D laser scanner + a stereo camera together with ORB-SLAM2) and different environments (small-scale indoor and large-scale



outdoor). The experiments with real data also show that the self-calibration approach is sufficiently robust against objects with high scattering characteristics like vegetation. This robustness is achieved by filtering or down-weighting the residuals that have large values.

#### 5.4.1. Small-Scale Indoor Dataset

The sensor setup used for the small-scale indoor dataset is the most challenging for calibration. In contrast to 3D laser scanners, where two different 3D scans can be registered using well-known techniques like the ICP algorithm, the registration of lines of a 2D laser scanner in 3D is not possible solely based on the scanning data. Thus, a wide range of approaches is not applicable. However, we can compare our results to a state-of-the-art target calibration, which is based on RADLOCC and described in more detail by [18]. However, this target calibration can not be seen as ground truth as its result has proven to be less accurate than some of the self-calibrations.

Figures 6–8 and Tables 1 and 2 show, that the parameters estimated by our self-calibration lead to a sharper point cloud than the parameters that are estimated by the target calibration. However, this is not a proof of outperforming the reference calibration in general. In contrast to the target calibration, our self-calibration is universally applicable, completely data-driven, and does not require a calibration target.

In comparison to the self-calibration based on the RQE, the self-calibration based on the feature of omnivariance leads to a sharper point cloud as well. This can be seen particularly at the qualitative comparison of Figures 6f and 7b and the quantitative comparison in Table 2.

#### 5.4.2. Large-Scale Outdoor Dataset

For the vehicle-based large-scale outdoor dataset, the result of our self-calibration is very similar to the reference calibration. However, neither the qualitative evaluation which is based on the appearance of the point cloud nor the quantitative evaluations are sufficient to assess which calibration is better. Anyway, the experiment with this dataset has proven the applicability of our self-calibration to vehicle-based mobile mapping with a 3D laser scanner and a stereo camera together with ORB-SLAM2 for pose estimation. Although the characteristics of the environment and the sensors are very different from the small-scale indoor dataset, the self-calibration improves the initial calibration parameters. The experiment thus shows the robustness of the self-calibration against these factors. In particular, the experiment also shows the suitability for a pose estimation approach that utilizes on-board sensors. However, systematic errors in pose estimation may have a negative impact on self-calibrations in general and should therefore be reduced to a minimum. The more inaccurate pose estimation and measurements of the laser scanner are supposed to be the main reasons that the results for the large-scale outdoor dataset are not as accurate as for the small-scale indoor dataset.

Compared to the automatic single-shot calibration of range and camera sensors [47] that was performed as the first step of the reference calibration of this dataset, our self-calibration is more accurate. The translation parameters are estimated with comparable precision. However, the estimation of the rotation parameters, is with approximately a factor of 10 much more accurate for our approach. One reason for this might be that our approach makes use of the stereo camera system, whereas the reference calibration is based on the measurements of a single camera. Another reason might be that our approach uses much more observations due to the nature of a self-calibration approach. Compared to the second step of the reference calibration, namely after manual refinement, the quality of our self-calibration is comparable.

The self-calibration based on the RQE again leads to less precise results compared to the self-calibration based on the feature of omnivariance. However, using the medians of multiple calibrations leads to point clouds with acceptable quality for most use cases.

## 6. Conclusions and Outlook

In this article, we propose a novel self-calibration approach for extrinsic calibration of Mobile Mapping Systems that include a mapping sensor and any kind of pose estimation sensor. Since it is a self-calibration, there is no need to involve calibration targets or manipulate the environment. Therefore, it can be performed based on the data that shall be collected for the 3D mobile mapping task. The approach uses geometric features that are computed based on the local neighborhood around each point of the cloud.

The experiments with synthetic data as well as with two different real datasets show the applicability to several sensor combinations in different environments. The investigations of the suitability of different cost functions demonstrate that the cost function based on the feature of omnivariance lead to more accurate results than the other investigated cost functions, including the cost function based on the Rényi quadratic entropy [8]. The self-calibration approach based on this feature has no assumptions about the environment. Coarse initial calibration parameters are assumed to be known. However, as the experiments with synthetic data show, the initial calibration parameters only need to be known approximately and can be roughly guessed in many cases, since the radius of convergence of the self-calibration is impressively large. Further, the experiments show that the use of a multi-scale approach considerably improves the calibration results. In contrast to single-scale approaches, the multi-scale approach is more robust as well as more accurate.

For all experiments with real sensor data, the accuracy of the calibration is about  $1^\circ$  for the rotation parameters. The accuracy of the translation parameters depends on the utilized sensors. For accurate sensors like in the small-scale indoor dataset, the accuracy of the translation is in the range of a few millimeters.

Future work focuses on the impact of different environments on the self-calibration. Further, it is theoretically possible to extend the presented self-calibration approach so that intrinsic parameters like a range offset of a laser scanner are estimated together with the extrinsic parameters.

In the current implementation, each point of the point cloud is used multiple times due to the neighborhood search. This is a disadvantage in terms of computational performance. In principle, there is potential here to make the approach more efficient. This could allow to use the basic ideas of our approach for on-line self-calibrations.

**Author Contributions:** Conceptualization, M.H., M.W., M.S.M. and B.J.; Formal analysis, M.H. and M.W.; Investigation, M.H.; Methodology, M.H. and B.J.; Project administration, B.J.; Software, M.H.; Supervision, B.J.; Validation, M.W. and M.S.M.; Visualization, M.H.; Writing—original draft, M.H. and M.W.; Writing—review & editing, M.H., M.W., M.S.M. and B.J.

**Funding:** This research received no external funding.

**Acknowledgments:** We thank Jochen Meidow for helpful discussions, which have contributed to a deeper understanding of the topic. We also thank Marcus Hebel and Joachim Gehring for the preparation of the *MLS 1—TUM City Campus dataset* [48] so that it could be used for testing purposes in the context of our work. We acknowledge support by the KIT-Publication Fund of the Karlsruhe Institute of Technology.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Zhang, Q.; Pless, R. Extrinsic Calibration of a Camera and Laser Range Finder (Improves Camera Calibration). In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Sendai, Japan, 28 September–2 October 2004; Volume 3, pp. 2301–2306.
2. Kassir, A.; Peynot, T. Reliable automatic camera-laser calibration. In Proceedings of the 2010 Australasian Conference on Robotics and Automation (ARAA), Brisbane, Australia, 1–3 December 2010.
3. Vasconcelos, F.; Barreto, J.P.; Nunes, U. A Minimal Solution for the Extrinsic Calibration of a Camera and a Laser-Rangefinder. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 2097–2107. [[CrossRef](#)] [[PubMed](#)]

4. Weinmann, M.; Jutzi, B.; Hinz, S.; Mallet, C. Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. *ISPRS J. Photogramm. Remote Sens.* **2015**, *105*, 286–304. [[CrossRef](#)]
5. Hackel, T.; Wegner, J.D.; Schindler, K. Fast semantic segmentation of 3D point clouds with strongly varying density. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2016**, *3*, 177–184.
6. Bueno, M.; Bosché, F.; González-Jorge, H.; Martínez-Sánchez, J.; Arias, P. 4-Plane congruent sets for automatic registration of as-is 3D point clouds with 3D BIM models. *Autom. Constr.* **2018**, *89*, 120–134. [[CrossRef](#)]
7. Frome, A.; Huber, D.; Kolluri, R.; Bülow, T.; Malik, J. Recognizing objects in range data using regional point descriptors. In Proceedings of the European Conference on Computer Vision, Prague, Czech Republic, 11–14 May 2004; pp. III:224–III:237.
8. Sheehan, M.; Harrison, A.; Newman, P. Self-calibration for a 3D laser. *Int. J. Robot. Res.* **2012**, *31*, 675–687. [[CrossRef](#)]
9. Kong, J.; Yan, L.; Liu, J.; Huang, Q.; Ding, X. Improved Accurate Extrinsic Calibration Algorithm of Camera and Two-dimensional Laser Scanner. *J. Multimed.* **2013**, *8-6*, 777–783. [[CrossRef](#)]
10. Yu, L.; Peng, M.; You, Z.; Guo, Z.; Tan, P.; Zhou, K. Separated Calibration of a Camera and a Laser Rangefinder for Robotic Heterogeneous Sensors. *J. Adv. Robot. Syst.* **2013**, *10-10*, 367–377. [[CrossRef](#)]
11. Hu, Z.; Li, Y.; Li, N.; Zhao, B. Extrinsic Calibration of 2-D Laser Rangefinder and Camera From Single Shot Based on Minimal Solution. *IEEE Trans. Instrum. Meas.* **2016**, *65*, 915–929. [[CrossRef](#)]
12. Sim, S.; Sock, J.; Kwak, K. Indirect Correspondence-Based Robust Extrinsic Calibration of LiDAR and Camera. *Sensors* **2016**, *16*, 933. [[CrossRef](#)]
13. Dong, W.; Isler, V. A novel method for the extrinsic calibration of a 2-D laser-rangefinder & a camera. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 5104–5109.
14. Li, N.; Hu, Z.; Zhao, B. Flexible extrinsic calibration of a camera and a two-dimensional laser rangefinder with a folding pattern. *Appl. Opt.* **2016**, *55*, 2270. [[CrossRef](#)]
15. Chen, Z.; Yang, X.; Zhang, C.; Jiang, S. Extrinsic calibration of a laser range finder and a camera based on the automatic detection of line feature. In Proceedings of the 2016 9th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), Datong, China, 15–17 October 2016; pp. 448–453.
16. Zhou, L. A New Minimal Solution for the Extrinsic Calibration of a 2D LIDAR and a Camera Using Three Plane-Line Correspondences. *IEEE Sens. J.* **2014**, *14*, 442–454. [[CrossRef](#)]
17. Tulsuk, P.; Srestasathien, P.; Ruchanurucks, M.; Phatrapornnant, T.; Nagahashi, H. A novel method for extrinsic parameters estimation between a single-line scan LiDAR and a camera. In Proceedings of the 2014 IEEE Intelligent Vehicles Symposium, Dearborn, MI, USA, 8–11 June 2014; pp. 781–786.
18. Hillemann, M.; Jutzi, B. UCalMiCeL—Unified intrinsic and extrinsic calibration of a Multi-Camera-System and a Laserscanner. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2017**, *4*, 17–24. [[CrossRef](#)]
19. Talaya, J.; Alamus, R.; Bosch, E.; Serra, A.; Kornus, W.; Baron, A. Integration of a terrestrial laser scanner with GPS/IMU orientation sensors. In Proceedings of the XXth ISPRS Congress, Istanbul, Turkey, 12–23 July 2004; Volume 35-B5, pp. 1049–1055.
20. Gräfe, G. High precision kinematic surveying with laser scanners. *J. Appl. Geod.* **2007**, *1-4*, 185–199. [[CrossRef](#)]
21. Heinz, E.; Eling, C.; Wieland, M.; Klingbeil, L.; Kuhlmann, H. Development, calibration and evaluation of a portable and direct georeferenced laser scanning system for kinematic 3D mapping. *J. Appl. Geod.* **2015**, *9-4*, 227–243. [[CrossRef](#)]
22. Jung, J.; Kim, J.; Yoon, S.; Kim, S.; Cho, H.; Kim, C.; Heo, J. Bore-sight calibration of multiple laser range finders for kinematic 3D laser scanning systems. *Sensors* **2015**, *15*, 10292–10314. [[CrossRef](#)] [[PubMed](#)]
23. Skaloud, J.; Lichti, D. Rigorous approach to bore-sight self-calibration in airborne laser scanning. *ISPRS J. Photogramm. Remote Sens.* **2006**, *61*, 47–59. [[CrossRef](#)]
24. Habib, A.F.; Kersting, A.P.; Shaker, A.; Yan, W.Y. Geometric calibration and radiometric correction of LiDAR data and their impact on the quality of derived products. *Sensors* **2011**, *11*, 9069–9097. [[CrossRef](#)]
25. Besl, P.; McKay, N.D. A method for registration of 3-D shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* **1992**, *14*, 239–256. [[CrossRef](#)]

26. Chen, Y.; Medioni, G. Object modelling by registration of multiple range images. *Image Vis. Comput.* **1992**, *10*, 145–155. [[CrossRef](#)]
27. Ravi, R.; Shamseldin, T.; Elbahnasawy, M.; Lin, Y.J.; Habib, A. Bias Impact Analysis and Calibration of UAV-Based Mobile LiDAR System with Spinning Multi-Beam Laser Scanner. *Appl. Sci.* **2018**, *8*, 297. [[CrossRef](#)]
28. Maddern, W.; Harrison, A.; Newman, P. Lost in translation (and rotation): Rapid extrinsic calibration for 2D and 3D LIDARs. In Proceedings of the 2012 IEEE International Conference on Proceedings of the Robotics and Automation (ICRA), Saint Paul, MN, USA, 14–18 May 2012; pp. 3096–3102.
29. Wang, F.; Syeda-Mahmood, T.; Vemuri, B.C.; Beymer, D.; Rangarajan, A. Closed-form Jensen-Renyi divergence for mixture of Gaussians and applications to group-wise shape registration. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, London, UK, 20–24 September 2009; pp. 648–655.
30. Maddern, W.; Pascoe, G.; Linegar, C.; Newman, P. 1 year, 1000 km: The Oxford RobotCar dataset. *Int. J. Robot. Res.* **2017**, *36*, 3–15. [[CrossRef](#)]
31. Jutzi, B.; Gross, H. Nearest neighbour classification on laser point clouds to gain object structures from buildings. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2009**, *38*, 4–7.
32. West, K.F.; Webb, B.N.; Lersch, J.R.; Pothier, S.; Triscari, J.M.; Iverson, A.E. Context-driven automated target detection in 3-D data. In *Automatic Target Recognition XIV*; International Society for Optics and Photonics: Orlando, FL, USA, 2004; Volume 5426; pp. 133–143.
33. Pauly, M.; Keiser, R.; Gross, M. Multi-scale feature extraction on point-sampled surfaces. *Comput. Graph. Forum* **2003**, *22*, 281–289. [[CrossRef](#)]
34. Dittrich, A.; Weinmann, M.; Hinz, S. Analytical and numerical investigations on the accuracy and robustness of geometric features extracted from 3D point cloud data. *ISPRS J. Photogramm. Remote Sens.* **2017**, *126*, 195–208. [[CrossRef](#)]
35. Munoz, D.; Bagnell, J.A.; Vandapel, N.; Hebert, M. Contextual classification with functional max-margin Markov networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 975–982.
36. Waldhauser, C.; Hochreiter, R.; Otepka, J.; Pfeifer, N.; Ghuffar, S.; Korzeniowska, K.; Wagner, G. Automated classification of airborne laser scanning point clouds. *Solving Computationally Expensive Engineering Problems: Methods and Applications*; Koziel, S., Leifsson, L., Yang, X.-S., Eds.; Springer: Cham, Switzerland, 2014; pp. 269–292.
37. Johnson, A.E.; Hebert, M. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Trans. Pattern Anal. Mach. Intell.* **1999**, *21*, 433–449. [[CrossRef](#)]
38. Tombari, F.; Salti, S.; Di Stefano, L. Unique signatures of histograms for local surface description. In Proceedings of the European Conference on Computer Vision, Heraklion, Greece, 5–11 September 2010; pp. III:356–III:369.
39. Salti, S.; Tombari, F.; Di Stefano, L. SHOT: unique signatures of histograms for surface and texture description. *Comput. Vis. Image Underst.* **2014**, *125*, 251–264. [[CrossRef](#)]
40. Osada, R.; Funkhouser, T.; Chazelle, B.; Dobkin, D. Shape distributions. *ACM Trans. Graph.* **2002**, *21*, 807–832. [[CrossRef](#)]
41. Rusu, R.B.; Blodow, N.; Beetz, M. Fast point feature histograms (FPFH) for 3D registration. In Proceedings of the IEEE International Conference on Robotics and Automation, Kobe, Japan, 12–17 May 2009; pp. 3212–3217.
42. Rusu, R.B. Semantic 3D object maps for everyday manipulation in human living environments. *Künstl. Intell. (KI)* **2010**, *24*, 345–348. [[CrossRef](#)]
43. Huber, P.J. Robust statistics. In *International Encyclopedia of Statistical Science*; Springer: Cham, Switzerland, 2011; pp. 1248–1251.
44. Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for autonomous driving? The KITTI vision benchmark suite. In Proceedings of the 2012 IEEE Conference on Proceedings of the Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–21 June 2012; pp. 3354–3361.
45. Mur-Artal, R.; Tardós, J.D. ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras. *IEEE Trans. Robot.* **2017**, *33*, 1255–1262. [[CrossRef](#)]
46. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Visual Odometry/SLAM Evaluation 2012. Available online: [http://www.cvlibs.net/datasets/kitti/eval\\_odometry.php](http://www.cvlibs.net/datasets/kitti/eval_odometry.php) (accessed on 23 July 2019).

47. Geiger, A.; Moosmann, F.; Car, O.; Schuster, B. Automatic camera and range sensor calibration using a single shot. In Proceedings of the International Conference on Robotics and Automation (ICRA), Saint Paul, MN, USA, 14–18 May 2012; pp. 3936–3943.
48. Gehring, J.; Hebel, M.; Arens, M.; Stilla, U. An Approach to Extract Moving Objects from MLS Data Using a Volumetric Background Representation. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2017**, *4*, 107–114. [[CrossRef](#)]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).