

Research Article

Monte Carlo Registration and Its Application with Autonomous Robots

Christian Rink, Simon Kriegel, Daniel Seth, Maximilian Denninger, Zoltan-Csaba Marton, and Tim Bodenmüller

Institute of Robotics and Mechatronics, German Aerospace Center, 82234 Oberpfaffenhofen, Germany

Correspondence should be addressed to Christian Rink; christian.rink@dlr.de

Received 25 March 2016; Revised 28 June 2016; Accepted 10 July 2016

Academic Editor: Pablo Gil

Copyright © 2016 Christian Rink et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This work focuses on Monte Carlo registration methods and their application with autonomous robots. A streaming and an offline variant are developed, both based on a particle filter. The streaming registration is performed in real-time during data acquisition with a laser striper allowing for on-the-fly pose estimation. Thus, the acquired data can be instantly utilized, for example, for object modeling or robot manipulation, and the laser scan can be aborted after convergence. Curvature features are calculated online and the estimated poses are optimized in the particle weighting step. For sampling the pose particles, uniform, normal, and Bingham distributions are compared. The methods are evaluated with a high-precision laser striper attached to an industrial robot and with a noisy Time-of-Flight camera attached to service robots. The shown applications range from robot assisted teleoperation, over autonomous object modeling, to mobile robot localization.

1. Introduction

Pose estimation is needed in a lot of different robotic applications, such as object pose estimation for grasping or manipulating objects, mobile robot localization, or registration of submodels in 3D modeling. In many cases a streaming pose estimation is beneficial or mandatory. One example is autonomous object modeling: if an object is placed on a table or other objects are in the proximity, the bottom or occluded part cannot be modeled without repositioning the object. However, if the object is repositioned and a registration is performed with newly acquired data, the autonomous 3D modeling could continue in order to acquire a complete object model. A streaming pose estimation after object repositioning has various positive impacts on the entire approach. First, time is saved as the pose estimation is readily available after data acquisition. Second, streaming pose estimation has the potential of reporting convergence or failure during data acquisition, enabling an autonomous reaction of the robot, for example, switching to modeling or rescanning. Third, after convergence, acquired data can be passed on-the-fly to modeling modules, resulting in a seamless transition from pose estimation to modeling.

Existing methods do not satisfy the requirements for on-the-fly global pose estimation. On the one hand, particle filters for mobile robot localization are able to keep pace with data acquisition. Unfortunately, they typically work either locally, are specialized to a certain sensor type, or cope only with 2D pose estimation. On the other hand, many global pose estimation methods cannot be adopted to work streamingly. Recently, we tried to fill this gap by introducing a particle filter registration [1] and adapting it to streaming pose estimation [2]. The work is aimed at applying in autonomous 3D modeling of unknown objects with laser striper [3].

In this paper, we review these Monte Carlo methods and their performance. We focus especially on streaming registration, meaning that the pose is estimated on-the-fly during data acquisition. For autonomous object modeling, we propose to combine the registration method with the approach presented in [4] for creating complete object models. Further, we show various use cases, such as in robot assisted teleoperation, allowing for partial autonomy when grasping a power screwdriver as can be seen in Figure 1.

Compared to our previous works [1–3], new and extended experiments especially in autonomous 3D modeling and mobile robot localization are shown (Sections 8.3 and 8.4).

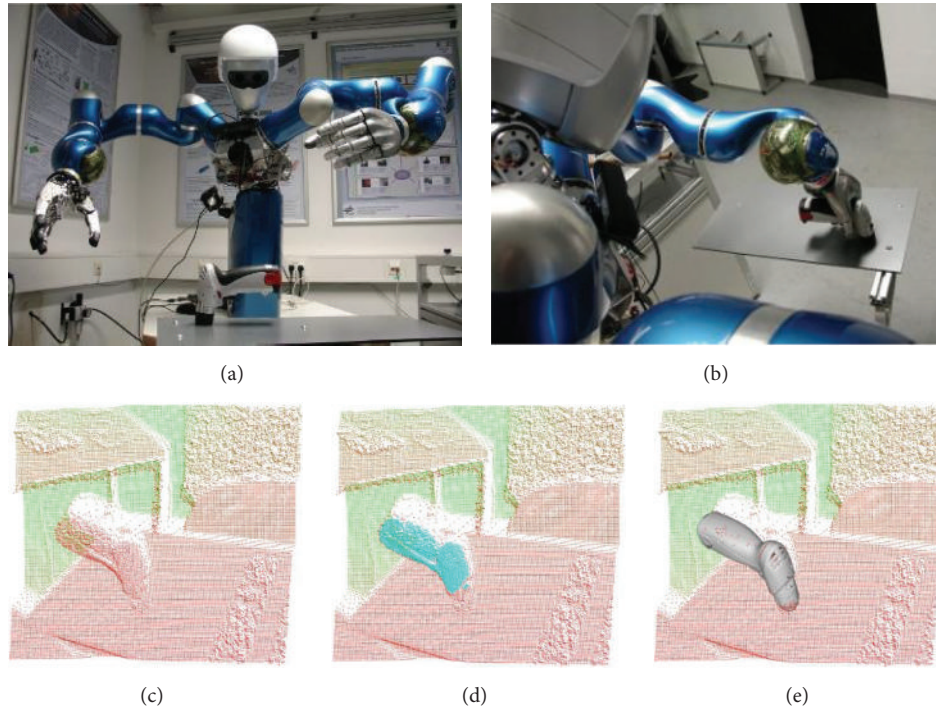


FIGURE 1: A humanoid robot (a) acquires a depth image of an object scene with a Time-of-Flight camera and registers the model of a power screwdriver ((c), (d), (e)) which allows for grasping the object (b). (c)–(e) depth values are color coded from red (close) to green (far), and the located model is shown in blue (points) and gray (CAD).

We enhance the streaming registration by providing a convergence criterion. Moreover, the influence of the sampling distribution, the introduced pose optimization during particle weighting, and the convergence criterion are investigated in detail. The main contributions in contrast to our previous works are

- (i) investigation of the influence of the proposed optimization step frequency on the robustness, reliability, and accuracy of results;
- (ii) investigation of the convergence behavior of the optimization frequency in comparison to the previous variants;
- (iii) definition of a convergence criterion;
- (iv) investigation of its influence on the robustness, reliability, and accuracy of results;
- (v) application in mobile robotics and comparison to a standard particle filter.

The remainder of this work is organized as follows. In Section 2, the related work concerning pose estimation and autonomous object modeling is reviewed. Then, in Section 3, the scalar features utilized in this work are introduced. Afterwards, particle filters are shortly reviewed and their application to registration is derived (Section 4). In the subsequent two sections, the sampling (Section 5) and scoring (Section 6) of particles are detailed. In Section 7, a feasibility study with the basic Monte Carlo method for global registration is performed. In Section 8, real experiments with

the streaming Monte Carlo registration and its results in the autonomous object modeling and mobile robot localization scenario are presented. The work is completed by a conclusion and a perspective on future work (Section 9).

2. Related Work

Registration is a crucial part of many technical applications such as reverse engineering, rapid prototyping, or manipulation tasks in manufacturing processes. It denotes the estimation of an unknown rigid motion between two 3D models of the same object.

Registration methods can be partitioned into fine (or local) and coarse (or global) ones. Fine methods require a good initial guess of the sought-after transformation. Coarse methods try to estimate the true rigid motion within a global search space. Both are reviewed in the following. For fine methods only a short overview including its application to mobile robot localization is given, as coarse methods are more closely related. Further, related work on real systems which autonomously model unknown objects is provided.

2.1. Fine Registration Methods. The iterative closest point (ICP) algorithm [5] is notably the most important representative of local methods (see [6] for an overview or [7] for a generalization). However, specialized methods have been proposed recently for RGB-D cameras [8–10] and image features [11, 12]. A further local technique related to our work is tracking with particle filters [13, 14], which is specialized for

fast local object movement estimation. Choi and Christensen [10] contribute a RGB-D object tracking approach implemented on a GPU and give an excellent general overview of particle filters in tracking problems, including edge based tracking, contour tracking, SLAM, and RGB-D based pose estimation. Therefore, the reader is referred to that work and the references cited therein for related tracking methods.

Localization and SLAM of mobile systems are also closely related but often specialized for local methods or methods restricted to 2D and could be categorized in between registration and tracking. A comprehensive overview of mobile robotics approaches is given by Sturm et al. [6]. Often particle filters based on pure depth images are used [15] which are not suitable for the stated problem, as the weighting is time consuming. Moreover, our estimation problem is not in 2D, and thus we need a much larger number of particles than classic mobile robotics particle filters. Additionally, we have to cope with higher update rates.

2.2. Coarse Registration Methods. In a large number of publications, contour and shape matching techniques are used, for which we refer the reader to the work of Ferrari et al. [16] and the literature cited therein. Note that we consider these techniques inappropriate for streaming pose estimation with laser stripers as they are slow and the contour or shape given by a few laser stripes is not very meaningful.

One of the most common methods in pose estimation is the random sampling consensus (RANSAC) introduced by Fischler and Bolles [17]. Its application to registration has been shown by Chen and Hung [18]. Commonly, subsets of points, point-normal pairs, or higher-dimensional features are sampled in the data sets to calculate unique rigid motions. Winkelbach [19] contributes an efficient way to sample point-normal pairs in order to build transformations. Drost et al. [20] also use point-normal pairs and a variant of the Generalized Hough Transform as voting scheme. Hillenbrand [21] contributes a robust cluster search in a set of transformations which are calculated from samples of either point-triples or point-normal pairs. Rusu et al. [22] use the Fast Point Feature Histogram in order to assign correspondences and use a sample consensus method in order to maximize the 3D overlap. Unfortunately, adapting these RANSAC-based methods to work with streaming data is not possible because a uniform sampling of points cannot be achieved before all data is acquired.

Another group of algorithms tries to group correspondences [23] or exploit salient points [24]. Aldoma et al. [23] evaluate various high-dimensional features for object recognition with a correspondence grouping method based on geometric consistency. A “center-star” variant of [24], followed by RANSAC-based filtering, yields a matching of similarly spaced point sets. In contrast to Rusu et al. [24] who search for salient points, points are sampled uniformly. The reduction to significant points is common. Gelfand et al. [25] reduce the data to a very small point set and apply a branch-and-bound algorithm to assign correspondences. The transformation is calculated by a least squares estimation. Cheng et al. [26] use a region growing algorithm to calculate

feature areas. In order to find correct correspondences, they use a relaxation labeling method. Both last methods rely on finding the unique and correct correspondences of feature points or areas. Again, this class of algorithms cannot be adapted to work with streaming data, since global data sets are needed, especially for feature calculation.

Barequet and Sharir [27] introduce a method that is based on the unique decomposability of rigid motions into a rotation and a translation. They iteratively search a discrete space of rotations by clustering the corresponding translations and finding the most definite cluster as the best rotation. In a later paper [28], they modify the method to work with directed features, that is, feature points with surface normals. They build ⟨feature point, normal⟩ pairs to directly get possible rotations (with 1 free DOF).

A similar approach has been proposed by Tombari and Di Stefano [29]. The features they use yield a complete reference frame, not only a sole surface normal (as in [28]). Therefore, a correspondence pair defines a rigid motion, in contrast to a set of pure rotations. However, the scoring/voting table is the same (up to a constant translation) as in [28]. In order to calculate the best translation, Tombari and Di Stefano apply an ICP iteration to the correspondence pairs after voting. Barequet and Sharir use the found transformation directly (ICP can be applied to the whole data set afterwards). Tombari and Di Stefano also use their method for object detection and prove that it is more robust and reliable than other standard methods that use a pose space clustering or geometric consistency.

Glover et al. [30] use a Bayes filter for pose estimation, where the rotational part of the transformations is represented by a Bingham distribution and extend their approach for multiple object detection in cluttered scenes [31].

Rink et al. [1] reformulate Barequet and Sharir’s approach as a particle filter and show that, in applications with very noisy data, relying on accurate surface normals or reproducible reference frames (as in [29]) can fail. Thus, scalar feature descriptors are proposed. Furthermore, a comparison to similar strategies [22, 23, 27, 29] is given and the advantages of explicit integration of prior knowledge about the searched transformation are presented. In a subsequent paper Rink et al. [2] advance the idea of particle filtering with scalar features to streaming pose estimation, adapting the search space to the space of rigid body transformations and giving a theoretically sound weighting of particles. The streaming feature calculation in that approach is based on a streaming principal component analysis used for tangential plane estimation in streaming mesh construction, proposed originally by Bodenmüller [32]. In a recent paper, Rink and Kriegel [3] optimize their method for the application in autonomous 3D modeling. An optimization in the particle weighting step is introduced and sampling pose particles according to a truncated Bingham/normal distribution is compared to uniform sampling. Further, they integrate the streaming pose estimation into an autonomous 3D modeling approach.

2.3. Autonomous Object Modeling. In autonomous object modeling, usually a robot-sensor system is utilized to plan a Next-Best-View (NBV) in order to acquire a 3D model of

an unknown object. Although the area of NBV planning has been widely explored [33, 34], there is little research on real systems for autonomous object modeling.

For the purpose of modeling cultural heritage objects, Karaszewski et al. [35] present a measurement system comprising a turntable and a vertically moveable pedestal. First, they select areas in the boundary area as viewpoint candidates. Then, low point density areas are selected. The digitization time is several hours, even for small objects. Khalfaoui et al. [36] combine an industrial robot, a turntable, and a very large and expensive fringe projection system. In order to plan NBVs, they define barely visible surfaces as viewpoint candidates. They perform a mean shift clustering for NBV selection. In order to avoid viewpoints being very close to each other, they use a minimal distance criterion. In the resulting model, several holes remain. Torabi and Gupta [37] use a smaller robot with 2D range sensor and focus on the exploration, not on the object modeling. Vasquez-Gomez et al. [38] autonomously reconstruct unknown objects with a mobile manipulator and a Kinect sensor. In order to avoid collisions, NBVs are sampled in configuration space and evaluated in Cartesian space.

Kriegel et al. [4] combine an industrial robot with a laser striper and focus on high quality model acquisition. Therefore, they define a quality criterion and consider the surface quality during Next-Best-Scan (NBS) planning. None of these approaches are able to obtain the bottom part of an object. In [39] the last approach [4] is used to create a pose estimation data set. There, initially occluded parts are also modeled. However, the objects have to be repositioned manually about a defined axis quite perfectly because an ICP is used for registration.

2.4. Distinction. In this work, we present Monte Carlo registration methods and apply it to different scenarios. The streaming registration works with pure depth measurements and is thus not specialized for a particular sensor. Notably it also works with laser stripers. It is a global method and works streamingly. To the best of our knowledge, there is so far no method satisfying these requirements. In addition to the results presented in [1–3], we perform a more in-depth analysis of the theoretical basis for the presented method. Further, we review the Monte Carlo registration approaches with more details concerning the used sampling methods. We improve the original methods, especially by equipping the streaming pose estimation method with a convergence criterion and perform extended experiments. Moreover, we combine the registration method with the approach presented in [4] which focuses on the modeling and plans NBSs for a laser striper for creating complete object models. Thus, we extend the autonomous modeling system by enabling the acquisition of complete object models by arbitrary repositioning of objects, which has not been done so far.

3. Features

For a streaming calculation only local features can be used, since regional or global features are computationally too

expensive. In the literature, various multidimensional features exist [22] that work well with a low or moderate level of noise. In this work scalar features are used for two reasons. On the one hand, if a small neighborhood radius is used for feature calculation, a high level of noise increases the number of false matches [23]. This limits the advantage of the higher expressiveness over scalar features. On the other hand, with a large neighborhood radius, an iterative calculation is difficult to achieve because an exhaustive neighborhood search has to be performed. Moreover, the scalar curvature features used in this work already proved to be suitable for an iterative streaming calculation; see [32] for an application of some of the features as a quality measure for streaming surface normal estimation.

Point clouds and triangle meshes are common representations for 3D sensor data and can be directly computed from range images or from streams [32]. Examples for angular features working on different data types are given by Barequet and Sharir [27]. However, as special data structures are presumed there, the proposed mean angles for meshes cannot be used for the situation considered here. In the following we present some applicable features for homogeneous triangle meshes and point clouds and show how to compute them streamingly. Note that every feature point $p = (c_p, n_p, v_p) \in \mathbb{R}^3 \times \mathcal{S}^2 \times \mathbb{R}$ comprises coordinates c_p , a surface normal n_p (\mathcal{S}^2 being the unit sphere), and a feature value v_p .

3.1. Normal Cosines in Polygon Meshes and Point Clouds. Let in the following p be a point with surface normal n_p and neighborhood $N(p)$. For a $q \in N(p) \setminus \{p\}$ we define

$$c(p, q) := \cos \left(n_p, \frac{q - p}{\|q - p\|} \right) \quad (1)$$

and call it normal cosine of p and q . Accordingly, the mean, maximum, and minimum of $\{c(p, q) \mid q \in N(p) \setminus \{p\}\}$ are called the mean normal cosine (MNC), maximum normal cosine (MaNC), and minimum normal cosine (MiNC) in p with neighborhood $N(p)$, respectively.

In this work we use two types of neighborhoods for the calculation of features, depending on the data structure that is used. A polygon mesh contains a set of vertices \mathcal{V} , a set of edges \mathcal{E} , and a set of polygons. Each edge $e \in \mathcal{E}$ is defined by two vertices $v_1, v_2 \in \mathcal{V}$, denoted by $\langle v_1, v_2 \rangle$. When dealing with homogeneous polygon meshes, it is reasonable to define the neighborhood of a vertex $p \in \mathcal{V}$ as all points that are connected with p via l edges. It can be defined recursively by $N_0(p) = \{p\}$ and

$$N_l(p) := N_{l-1} \cup \{q \in \mathcal{V} : \exists \bar{p} \in N_{l-1}(p) : \langle \bar{p}, q \rangle \in \mathcal{E}\}. \quad (2)$$

Figure 2 shows the MNC, the MaNC, and the MiNC of a triangle mesh of a wooden workpiece and their histograms. Note that points containing border vertices in their neighborhood are excluded from the feature calculation (depicted in white) because a robust feature calculation is not possible for this case. Since holes especially arise in high curvature areas,

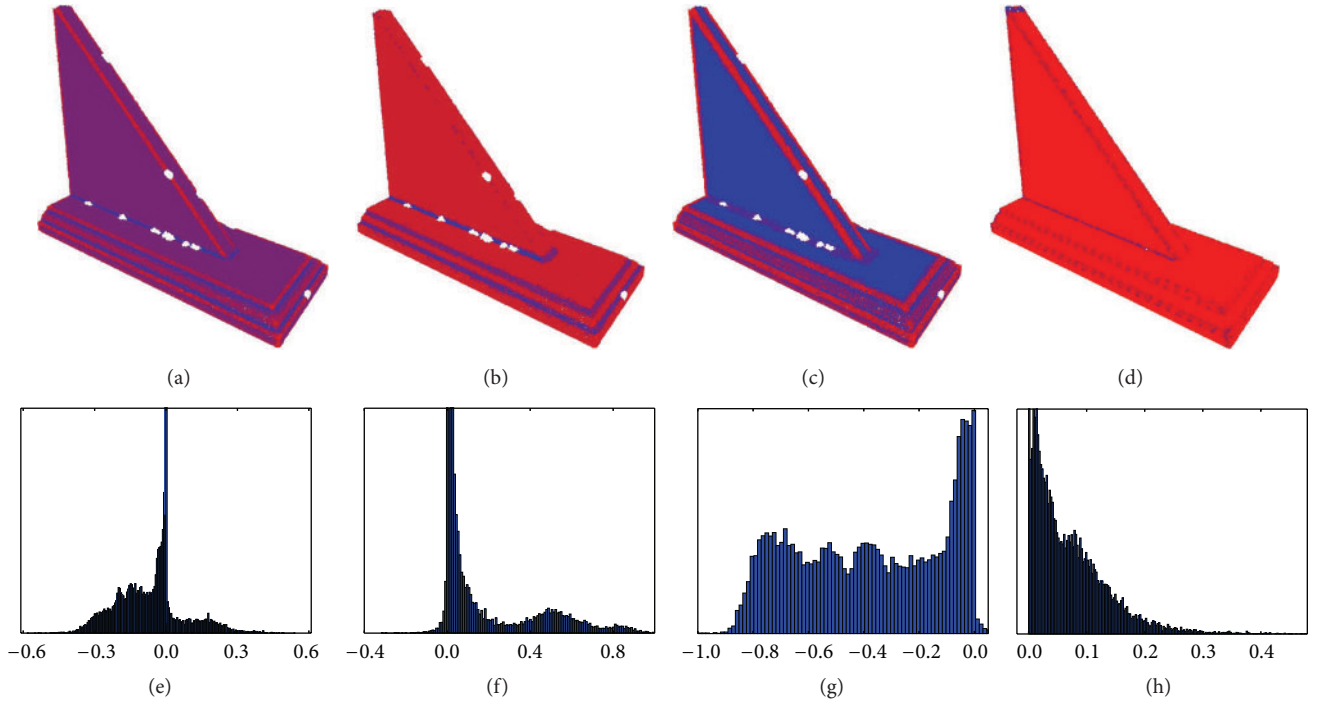


FIGURE 2: (a)–(d) The MNC, MaNC, MiNC, and EVQ13 of a wooden workpiece (high values are light red and low values are blue). (e)–(h) The corresponding histograms (most frequent value cut-off).

significant features could be incorrectly computed there. Note that in this case synthetic point data is used for mesh and feature generation. In nontechnical data sets, the distribution of the features is often closer to a normal distribution than in this example.

If point clouds are used or the polygon meshes are not homogeneous (i.e., edge lengths vary strongly), we use ball neighborhoods. For a point cloud P and a point $p \in P$ let

$$N_r(p) := \{q \in P : \|p - q\| \leq r\}. \quad (3)$$

The normal cosine used in practice depends on the objects that are expected. Our experience is that convex regions in objects can be extracted with MNC and MiNC and concave regions with MNC and MaNC. For the general case we propose the MNC. If mainly convex regions are expected to be more discriminative we propose the MiNC and if mainly concave regions are expected we propose the MaNC.

3.2. Point Clouds and Eigenvalues. If no homogeneous polygon mesh or reliable normal estimation is given in advance, alternative features can be calculated. Let $\lambda_1 \leq \lambda_2 \leq \lambda_3$ be the eigenvalues of a point neighborhood covariance matrix. Then λ_1/λ_3 defines a curvature measure and can be used as geometric feature, denoted as eigenvalue quotient of eigenvalues 1 and 3 (EVQ13). The literature on similar features exists [40–43]. Figures 2(d) and 2(h) show the example features of a wooden workpiece. Note that missing features do not emerge, but ambiguity between convex and concave regions arises.

Flat objects like metal sheets do not yield relevant curvature features, disabling a robust pose estimation. In such cases we use the feature value λ_2/λ_3 , denoted as EVQ23. This feature characterizes border points of the object (see Figure 13 for an example).

3.3. Streaming Feature Calculation. Feature-based streaming pose estimation requires streaming feature calculation which is implemented by a processing pipeline comprising three stages: the *density limitation*, the *normal estimation*, and the *feature generation* step. The depth points coming from a real-time data stream have to pass a limitation test in order to be inserted into the model: each newly acquired point that is closer than a distance r_r to any point already inserted into the model is rejected. Thus, the computational effort can be controlled because the entire Euclidean point density of the model is limited. For each point passing the *density limitation*, a surface normal is estimated using principal component analysis for all points within a ball neighborhood with radius r_n . Only points with a robust *normal estimation* (see [32] for details) are transferred to the subsequent *feature generation* step.

3.3.1. Eigenvalues. At the end of the *normal estimation* stage, the eigenvalues of the point neighborhood covariance matrix are readily available from the principal component analysis. Thus, the streaming feature calculation for EVQ13 and EVQ23 is straightforward: if a stable normal is ready, the corresponding feature point is calculated

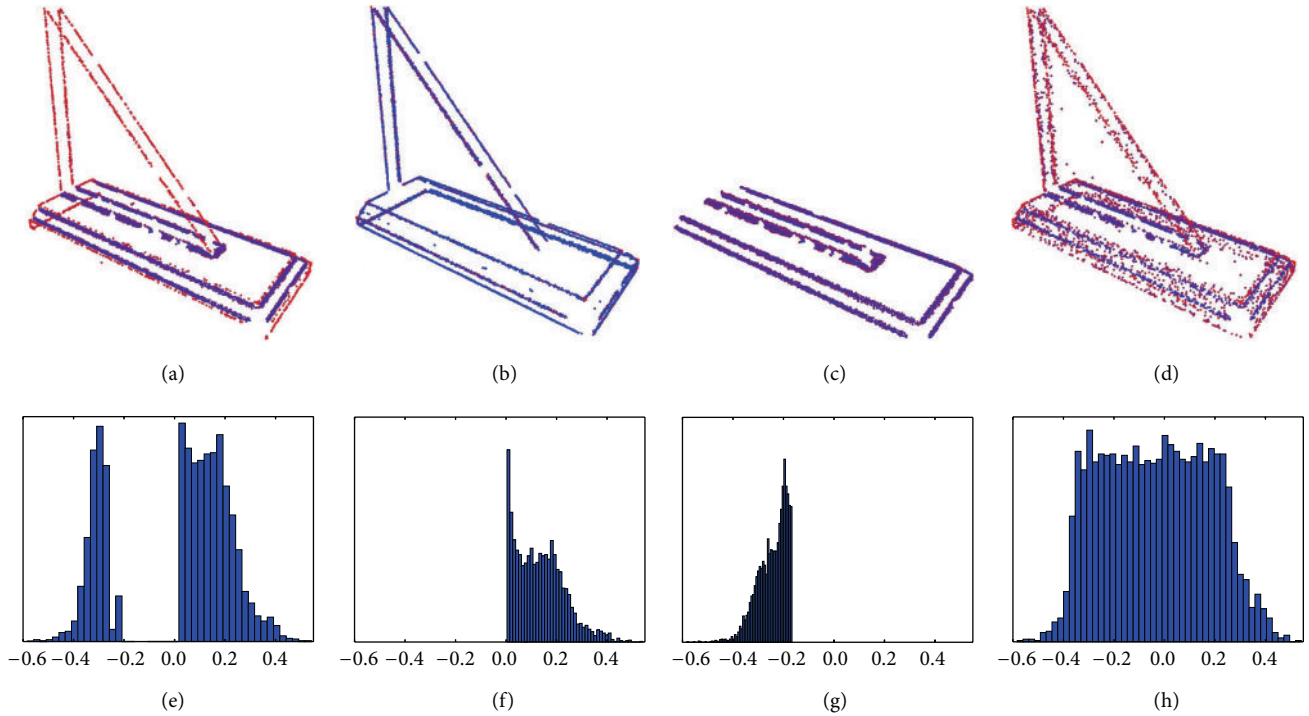


FIGURE 3: The reduction types (from left to right): removal of biggest bins, removal of leftmost bins, removal of rightmost bins, and randomized removal in biggest bins.

from the eigenvalues and inserted into the feature point stream.

3.3.2. Normal Cosines. The proposed angle features are calculated from the point surface normal and the neighborhood points in the model. Consequently, if a stable normal is ready and the new point is inserted into the module, the MNC, MaNC, or MiNC is calculated in the neighborhood of radius r_n (immediately available from the *normal estimation* stage). Also, all the points in the neighborhood are updated correspondingly.

3.3.3. Automatic Feature Type Selection. Theoretically, the most appropriate feature can be automatically selected by calculating the MNC and evaluating the histogram. For unimodal symmetric distributions, the MNC should then be used. If the distribution is skewed, the MiNC could be used for a heavy left tail and the MaNC for a heavy right tail. In practice, however, the automatic choice of the other parameters (like the neighborhood radius or meshing parameters) used for feature calculation poses a problem, especially if computation time is an issue.

3.4. Feature Point Reduction, Classification, and Correspondences. Besides the correspondence assignment features can be used to reduce the data to significant points. On the one hand this leads to lower computational costs. On the other hand correspondence assignment is concentrated on

characteristic areas of the object, reducing the number of false positive correspondences.

Therefore, the feature values are first summarized in a histogram. Then the histogram's bins can be removed iteratively. Either the biggest bin (see Figure 3) or the leftmost/rightmost bin is removed until only a given percentage of the original data remains. Alternatively, it can be useful to keep some points from the biggest histogram bins, for example, when dealing with great flat areas. Then, the smallest bins are kept and from the remaining bins a certain fraction is sampled so that the desired overall fraction remains in the end. Figure 3 shows the differences in these reduction strategies for the MNC. In ((a), (b), (c), (d)), the remaining feature points are depicted. In ((e), (f), (g), (h)), the corresponding histograms of feature values are shown. Clearly, the removal of the biggest bins results in remaining convex and concave extrema. The removal of the leftmost/rightmost bins results in remaining convex/concave extrema. In this case, weighted random reduction in the biggest bins results in evenly sized bins for the most frequent feature values.

In order to assign correspondences, we use discretized feature values, so-called feature classes, conferring some robustness to noisy sensor data. Given a number n of feature classes, the classes are defined uniquely by the maximum feature f_{\max} , the minimum feature f_{\min} , and an equal bin width $b = (f_{\max} - f_{\min})/n$. In this work, values of $n = 7$, $n = 5$, or $n = 3$ are used. Then, every feature point of one data set corresponds to every feature point with a feature of the same class in the other data set. This implicates a normalization and allows the data sets to be measured with different sensors.



FIGURE 4: The noise in a ToF cam depth image prevents reliable geometric feature calculation. The power screwdriver is colored light blue.

3.5. Features in Time-of-Flight Camera Data. Robust feature calculation from noisy Time-of-Flight camera (ToF cam) data is a special challenge. Figure 4 illustrates the noise with an exemplary depth image of a power screwdriver. In such cases, the selection of key points relying purely on geometry fails. Though, we take advantage of the additional intensity image provided by ToF cams because optical and geometric edges often coincide in the data of technical products. In this particular use case, we search features on the handle of the screwdriver. On the CAD data template, the fine geometric features can easily be detected; see Figure 5. In the real ToF cam depth image, these geometric edges cannot be found. However, the intensity gradients in the intensity image include the searched edge at the handle of the power screwdriver. As they also include the transition between the shape of the screwdriver and the background, edges in the range image must be subtracted. Therefore, we use an image filter combination consisting of dilation, blurring, and a Canny edge detector on the intensity image as well as on the depth image to acquire the edges (for filtering the software OpenCV 2.4 is used). Finally, the depth edge image is subtracted from the intensity edge image, as shown in Figure 6. In total, exactly the geometric edges we sought for are extracted. Section 7.2.2 shows an application and more details.

4. Monte Carlo Registration

In this section, we introduce the basic Monte Carlo registration methods. First, we review the general particle filter and its specialization for registration. Then, the proposed methods are derived.

4.1. Particle Filter. For a detailed description of particle filters and their applications the reader is referred to the literature [10, 13–15]. Here, just the notations used in subsequent sections are given and the most important points are reviewed.

Let X be a random variable and let $f(X | \theta)$ be the probability density function (pdf) of X conditioned on some parameter θ . Let further $p(\theta)$ be the a priori pdf of the parameter θ . According to Bayes' rule, after some observation x of X , the posterior of θ is given by

$$p(\theta | x) \propto f(x | \theta) p(\theta), \quad (4)$$

with \propto being the symbol for proportionality. If repeated measurements of X are carried out, Bayes' rule can be applied iteratively (if the measurements are statistically independent). Suppose that the state of the parameter θ changes between the observations x_i and x_{i+1} of X according to a transition function A_i :

$$\theta_{i+1} = A_i(\theta_i) + \varepsilon_{A_i}, \quad (5)$$

where $A_i(\cdot)$ describes a systematic change with an error ε_{A_i} that follows the pdf g_{A_i} . The transition A_i is changing over time in many applications. Moreover, ε_{A_i} often depends on A_i . Therefore, the pdf g_{A_i} is changing, too. In the case of registration, A_i as well as ε_{A_i} can be assumed to be constant. The assumption of first-order Markov chains and the independence of θ_{i+1} from x_i yields

$$\begin{aligned} p(\theta_{i+1} | x_{i+1}, \dots, x_1, \theta_i, \dots, \theta_1, A_i) \\ &= p(\theta_{i+1} | x_{i+1}, \theta_i, A_i) \\ &\propto f(x_{i+1} | \theta_{i+1}, \theta_i) p(\theta_{i+1} | \theta_i, A_i) \\ &= f(x_{i+1} | \theta_{i+1}) p(\theta_{i+1} | \theta_i, A_i). \end{aligned} \quad (6)$$

Now let the pdf of θ_i be represented by m_i particles:

$$(\theta_i^j, w_i^j)_{j=1, \dots, m_i} \quad (7)$$

with sampled states θ_i^j and corresponding weights w_i^j . After a transition according to A_i and a new observation of X , new particles are sampled according to (5). Note that sampling from the pdf $g(\varepsilon_{A_i})$ has to be possible, which is not generally the case. Therefore, ε_{A_i} is typically assumed to be distributed normally or uniformly. The new particles are each weighted with

$$w_{i+1}^j = f(x_{i+1} | \theta_{i+1}^j), \quad j = 1, \dots, m_i. \quad (8)$$

Finally, m_{i+1} particles are resampled according to these weights. Thus, particle filtering can be summarized as follows:

- (1) Initialize ($i = 0$) the m_0 particles θ_0^j .
- (2) Collect data x_i and weight the particles with

$$w_i^j := f(x_i^j | \theta_i^j). \quad (9)$$

- (3) Resample m_{i+1} particles with the weights w_i^j .
- (4) Apply the transition by $\theta_{i+1}^j = A_i(\theta_i^j)$.
- (5) Set $i := i + 1$ and return to Step (2).

4.2. Monte Carlo Registration. In the case of Monte Carlo registration, the particles can either be sampled in the space of rotations [1] or rigid body transformations [2]. Let \mathcal{T} be the corresponding state space in either case. Further, $T_i \in \mathcal{T}$ denotes the unknown transformation between two models P and Q of the same object at time step i . Then, every particle

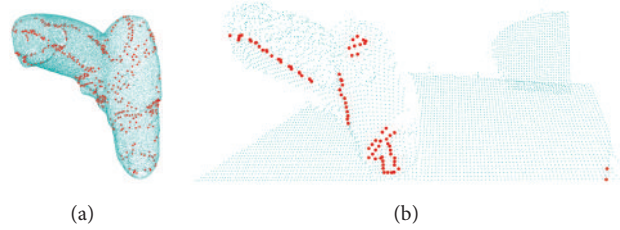


FIGURE 5: Point cloud models describing a screwdriver: (a) shows the template point cloud model (light blue) and the calculated feature points (red). (b) shows the cropped depth image with emphasized feature points (red) extracted from the ToF data (before plane deletion).



FIGURE 6: Edges detected in cropped ToF data. From (a) to (d) intensity, intensity edges, range edges, and edge difference image. In the intensity image the color gradient of the handle of the power screwdriver being used as a feature is conspicuous. This gradient can be recognized in the (d) difference image again.

comprises a transformation $T \in \mathcal{T}$ and a weight w . The state transition between two time steps is the identity in the case of registration because the observed object is not moving ($A_i = \text{id}$). The distribution of the error ε_{A_i} can be assumed to be a uniform or normal distribution. In each time step i , all particles (T, w) are weighted by the pdf $f(Q_i | T, P)$ of observed data Q_i conditional on the state T and the template model P . Then, particle filter registration can be summarized as follows:

- (1) sample pose particles initially;
- (2) weigh particles by $f(Q_i | T, P)$;
- (3) resample particles according to their weights;
- (4) optionally: adapt some parameters;
- (5) sample particles in neighborhoods of existing particles; return to Step (2) if not converged.

4.3. MCR and SMCR. The Monte Carlo registration method presented in [1] is denoted by MCR in this paper. It works offline and uses all feature points in every weighting step and the particles are rotations. The particle weighting is done by a heuristic cluster evaluation.

The streaming Monte Carlo registration method presented in [2] is denoted by SMCR in this paper. It works streamingly and can use either new incoming feature points or all accumulated feature points in the weighting steps. The particles are rigid body transformations. The particle weighting is done with respect to a truncated normal distribution. For SMCR, the number of particles, the sampling radii, and the radius for the score function from Section 6 are adapted in each Step (4) of Section 4.2. Each of these parameters has a maximum value and is reduced by a factor of 0.8 in Step (4), until a minimum value is reached.

5. Sampling Pose Particles

Initially the particles are sampled uniformly in a region according to the prior knowledge. In the subsequent iterations only neighborhood sampling is performed. This is done either with truncated uniform distributions or truncated normal/Bingham distributions. The radius of the truncated neighborhood can be adapted with convergence over time.

As sampling of translations uniformly or normally in neighborhoods is trivial, only the sampling of rotations is detailed here. Rigid body transformations are sampled by sampling the rotational and translational parts separately.

5.1. Sampling Unit Vectors Uniformly. A prerequisite for our approach to sample rotations uniformly is to sample uniformly distributed points in neighborhoods on the unit sphere \mathcal{S}^2 in \mathbb{R}^3 . Sampling a point on the unit sphere can be achieved by the following steps:

- (1) Sample v uniformly distributed in $[-1; 1]$.
- (2) Sample u uniformly distributed in $[-\pi; \pi]$.
- (3) Calculate $\mathbf{c} := \cos(\arcsin(v))$.
- (4) Build $p = (\mathbf{c} \cdot \cos(u), \mathbf{c} \cdot \sin(u), v)^T$.

According to the transformation theorem for densities, p will be uniformly distributed on the unit sphere [44].

Let the α -neighborhood $N_\alpha(a)$ of a vector $a \in \mathcal{S}^2$ be defined as

$$N_\alpha(a) := \{\bar{a} \in \mathcal{S}^2 \mid \angle(a, \bar{a}) \leq \alpha\}, \quad (10)$$

where $\alpha \in [0, \pi]$.

Then, the above approach can be adapted to sample unit vectors in an α -neighborhood of the first standard basis

vector e_1 with $\alpha \leq \pi/2$. Just modify steps one and two as follows:

- (1) Sample v uniformly distributed in $[-\sin(\alpha); \sin(\alpha)]$.
- (2) Sample u uniformly distributed in $[-\alpha; \alpha]$.

This specialization is biased for $\alpha > \pi/2$. In this case simple rejection sampling can be used.

5.2. Sampling Rotations in Neighborhoods Uniformly. In statistics, various strategies are common to choose priors that express ignorance about a parameter. In this work, uniform distributions are used for the initial sampling of transformations. The most common way to achieve a deterministic uniform sampling in a space is to build a homogeneous grid in that space. However, this leads to biased grids when dealing with the common representations of rotations. Sampling matrices straightforward in this manner is not even possible. There are sophisticated algorithms for deterministic uniform sampling of rotations [45–47], but unbiased deterministic sampling is not possible.

However, uniform sampling in a statistical sense can be achieved [48, 49] and is advantageous compared to simple grid sampling on Euler angles [1]. In the remainder of this section we detail our variant of Shoemake's method [49] to sample rotations uniformly in neighborhoods. Let \mathcal{R} be the space of rotations in the remainder. The α -neighborhood $N_\alpha(R)$ of a rotation R is defined as

$$N_\alpha(R) := \{\tilde{R} \in \mathcal{R} \mid d(R, \tilde{R}) \leq \alpha\}, \quad (11)$$

where $\alpha \in [0, \pi]$

with $d(R, \tilde{R})$ being the rotational difference between two rotations R, \tilde{R} , that is, the absolute value of the angle of the axis-angle representation of $R \circ \tilde{R}^{-1}$. Let e_1, e_2, e_3 be the basis vectors of the base coordinate system. In order to sample rotations on α -neighborhoods, we represent a rotation by the transformed coordinate system $\tilde{e}_1, \tilde{e}_2, \tilde{e}_3$, that is, the columns of the rotation matrix. Sampling rotations in a neighborhood $N_\alpha(R)$ of R is done by sampling a rotation R_α in $N_\alpha(\text{id})$ and calculating $\tilde{R} = R \circ R_\alpha$. In order to sample in $N_\alpha(\text{id})$, we propose the following procedure (see Figure 7):

- (1) Sample \tilde{e}_1 in the α -neighborhood of e_1 .
- (2) Calculate the vector

$$v_3 = \begin{cases} e_3, & \text{if } \tilde{e}_1 = \pm e_2, \\ \tilde{e}_1 \times e_2, & \text{else.} \end{cases} \quad (12)$$

- (3) Calculate the vector $v_2 = v_3 \times \tilde{e}_1$.
- (4) Rotate vectors v_2, v_3 around \tilde{e}_1 with a random angle φ onto \tilde{e}_2 and \tilde{e}_3 . If $\alpha < \pi/2$ sample φ in $[-\alpha; \alpha]$. Otherwise, sample φ in $]-\pi; \pi]$.
- (5) Build the rotation matrix $R := (\tilde{e}_1, \tilde{e}_2, \tilde{e}_3)$. If the rotation angle holds $d(R, \text{id}) \leq \alpha$ accept it; else return to Step (1).

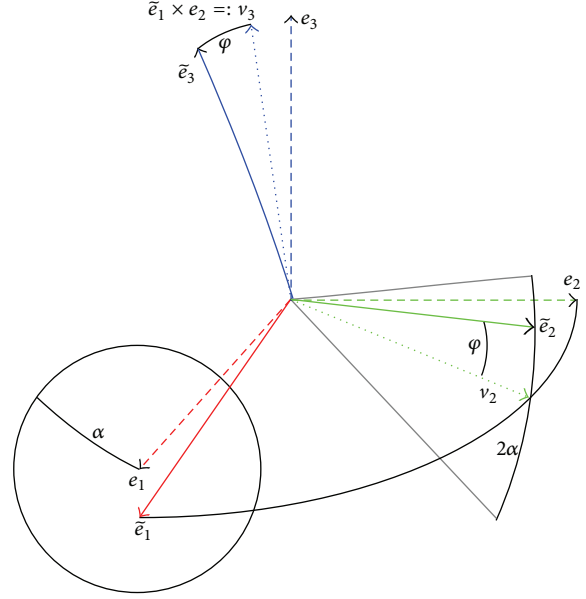


FIGURE 7: Uniform neighborhood sampling of rotations: \tilde{e}_1 is sampled uniformly in the α -neighborhood of e_1 . Then v_2 is chosen perpendicular to \tilde{e}_1 and in the plane spanned by \tilde{e}_1 and e_2 . Rotating around \tilde{e}_1 with a random angle smaller than α makes v_2 independent of \tilde{e}_1 .

Figure 7 illustrates these relations: v_2 is perpendicular to v_3 , which in turn is perpendicular to \tilde{e}_1 and e_2 . Therefore, v_2 lies in the plane spanned by \tilde{e}_1 and e_2 and thus v_2 is as close as possible to e_2 conditioned on a fixed \tilde{e}_1 . This assures an overall rotation angle as small as possible before applying the random rotation around \tilde{e}_1 in Step (4). Rotating around \tilde{e}_1 moves v_2 onto \tilde{e}_2 . For $\alpha < \pi/2$, rotations around \tilde{e}_1 by an angle greater than α cause v_2 to leave the α -neighborhood of e_2 . For $\alpha \geq \pi/2$, this is not necessarily the case. Together, this motivates Step (4) because each transformed basis vector \tilde{e}_i must not lie outside the α -neighborhood of the basis vector e_i if the resulting rotation is to lie in the α -neighborhood of the identity. Rotations that lie outside the neighborhood are removed by Step (5). Figure 8(a) shows uniformly sampled rotations in an α -neighborhood with $\alpha = 90^\circ$. The rotations are represented by the transformed unit vectors \tilde{e}_1 and \tilde{e}_3 , that is, the first and last columns of the corresponding rotation matrix. For better visibility, e_2 was left out.

5.3. Truncated Bingham Sampling. On the space of rotations, no normal distribution exists, though so-called projected Gaussians have been proposed [50]. Most similar are special cases of Bingham distributions, which have already been used for pose estimation [30]. We propose to sample from a truncated special case of a Bingham distribution, with rejection sampling. Shortly, after sampling a rotation R uniformly in an α -neighborhood of a rotation \tilde{R} , we do rejection sampling with the weight $\exp(d(R, \tilde{R})^2/2\sigma^2)$. This sampling strategy is easy to implement and if σ^2 is chosen appropriately (independent of α), the computation time is negligible compared to the weighting step of the particles.

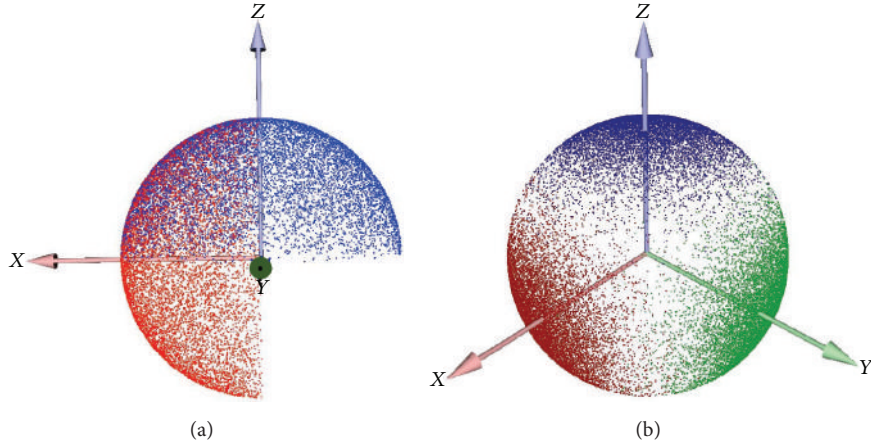


FIGURE 8: Uniformly (a) and truncated Bingham (b) distributed rotations: 10^5 images $R(e_1)$ (red), $R(e_2)$ (green, removed in (a) for better visibility), and $R(e_3)$ (blue) with neighborhood radius $\alpha = 90^\circ$ from different views. Variance of the normal is $\sigma^2 = (18^\circ)^2$.

Figure 8(b) shows truncated Bingham sampled rotations in a neighborhood of radius 90° . For better visibility, the normal distribution's variance is chosen as 18° in this example. In practice, we choose the standard deviation to be half the neighborhood radius.

6. Scoring Pose Particles

In this section, the scoring methods for rotations and rigid body transformations are presented. Scoring of a rotation is done by a cluster evaluation on the set of corresponding possible translations. This voting scheme is very similar to the generalized Hough transform. Scoring of a rigid body transformation is done by evaluating a truncated normal pdf. The former voting scheme is used in MCR and the latter in SMCR.

6.1. Scoring Rotations. In order to score a rotation, it is first applied to the corresponding data set. Then, all translational differences between all corresponding points are calculated (correspondences between the data sets are defined by equal feature classes). These differences define the set of all possible translations for the considered rotation. In order to find the one rotation with the most clustered set of such translations, two strategies are considered in this work. The first is to store the translations in a three-dimensional voting table and use the maximum number of elements in one bin as score. This method will be denoted in table in the remainder and is detailed in [27–29]. The second is to store the translations in a voxel space and use the maximum number of neighbors in a ball neighborhood as score; see Figure 9. This scoring method will be denoted as nb and was originally presented in [1]. The pdf $f(Q_i | R, P)$ (see Section 4.2) cannot be used as score function, since it is not known and there is no reasonable assumption about it.

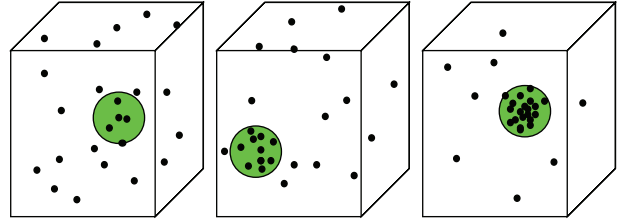


FIGURE 9: Scoring rotations: the maximum number of translations in a ball neighborhood in the set of translations defines a score for the clusteredness.

6.2. Scoring Rigid Body Transformations. The scores of the particles representing rigid body transformations are calculated according to a truncated normal distribution of the feature point coordinates, conditional on the pose. Let P, Q be the feature point sets of the template and the incoming data, correspondingly. Further, let the features be classified; that is, v_p is a discrete category for every $p \in P$ (and correspondingly for Q). Every particle describes a rigid body transformation $T = (R, t)$, defined by a rotation R and a translation t . In the following, $q \in Q$ is corresponding to $p \in P$. For a known transformation T between data and template, it is reasonable to assume that c_q follows a normal distribution with expectation $T(c_p)$ and a covariance matrix $\Sigma = \sigma^2 \cdot \text{id}$. If the errors are identically and independently distributed and a set of feature points $\mathbf{p} = \{p_1, \dots, p_n\}$ and a set of correspondences $\mathbf{q} = \{q_1, \dots, q_n\}$ are given, the conditional pdf of all feature point locations is

$$f(\mathbf{q} | T, \mathbf{p}) \propto \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^n \|T(c_{p_i}) - c_{q_i}\|^2\right). \quad (13)$$

The corresponding p_i are approximated by the nearest feature point to q_i with the same feature class:

$$p_i := \arg \min_{p \in P, v_p = v_{q_i}} \|T(c_p), c_{q_i}\|. \quad (14)$$

For a q with erroneous feature class, no correct corresponding p will be found in the template. The best we can do in this case is to assume a uniform distribution of the feature point location, conditional on a wrong corresponding p . For some distance threshold r_{\max} we define

$$d_i := \min \left\{ \|T(c_{p_i}) - c_{q_i}\|, r_{\max} \right\} \quad (15)$$

and adopt (13) to

$$f(\mathbf{q} | T, \mathbf{p}) \propto \exp \left(-\frac{1}{2\sigma^2} \sum_{i=1}^n d_i^2 \right) \quad (16)$$

which corresponds to (truncated) normally distributed errors if the correspondences are found within a radius r_{\max} and uniformly distributed errors if not (with the density equal to that of the truncated normal at its boundary). This is actually an improper pdf because its integral is unbounded. Nevertheless, sampling importance resampling [51] is possible with it. Thus, each transformation T is scored with feature points Q as follows. First, each point $q_i \in Q$ is classified according to the class borders of the template. It is transformed to $c_{q_i} := T^{-1}(c_{q_i})$ and the template model is searched for the nearest feature point p_i of the same feature class. The corresponding distance is denoted as $d_i := \|c_{q_i} - c_{p_i}\|$. If no such point is found within the search radius r_{\max} , the distance is set to this radius ($d_i := r_{\max}$). Based on the distances d_i of all available feature points p_i , the particle weight is defined by

$$w(T) := \exp \left(-\frac{1}{2\sigma^2} \sum_{i=1}^n d_i^2 \right). \quad (17)$$

6.2.1. Scoring Variants. Theoretically, a particle filter uses only statistically independent measurements for each update; that is, only new incoming feature points are utilized for Q . Let therefore Q_j be the incoming set of feature points at time step j . The results in Section 8.2 will show that $Q = Q_j$ leads to poor convergence behavior. The convergence can be improved by using all previously measured feature points for Q ; that is, $Q := \bigcup_{k=0}^j Q_k$. In order to distinguish the two scoring variants, we call the former *streaming particle filter registration* (SPFR) and the latter *streaming Monte Carlo registration* (SMCR) in the remainder. As shown in [2, 3] and reviewed in Section 8.2, SMCR has a better convergence behavior. Note that the offline variant MCR for global registration always uses all feature points.

6.2.2. Optimization. If a p_i with $d_i \leq r_{\max}$ according to (14) and (15) is found, it defines a correspondence to q_i . Thus, each particle yields a set of correspondences in the weighting step. In order to correct the pose particle, these correspondences can be used for an ICP iteration. If such an optimization step is applied, the method is denoted by SMCRO in the remainder. An additional subscript defines the frequency of such an optimization; for example, SMCRO₅ denotes an optimization in every fifth update.



FIGURE 10: The two data sets obtained by selecting parts of the Stanford Bunny.

6.2.3. Convergence Criterion. One advantage of streaming pose estimation is the possibility of stopping data acquisition as soon as the estimation converged. For this purpose a convergence criterion for SMCR is introduced. In every update step we calculate the rotational and translational difference of the highest rated transformation to the highest rated transformation from the last step. If the differences are below some thresholds c_r and c_t in five consecutive steps, the calculation is aborted. We combine the optimization in every step with this criterion and denote the method by SMCRC.

7. Feasibility Study with MCR

In this section, we briefly summarize the most important results that are obtained with MCR. For a more detailed review, the reader is referred to [1]. The results serve as feasibility study on the question, whether this kind of particle filter can compete with the state-of-the-art registration methods. Otherwise, the effort of developing a streaming variant would not be justified.

7.1. Validation with Artificial Data. For the validation with artificial data, two submeshes of the well-known Stanford Bunny are extracted; see Figure 10. The computing time \bar{t} , the mean rotational error $\bar{\rho}$, and the percentage of successful estimations sr are investigated. A successful estimation is defined by a rotational error less than 20° and reflects robustness as such errors can be equalized by ICP. All experiments in this section were performed by 1000 test runs. In each of these test runs, the underlying rotation was chosen randomly.

First, three different sampling strategies are examined. The first is the original one [27], denoted by *det* in the following. There, rotations are represented by Euler angles and sampled initially on a homogeneous grid with resolution ρ_0 . In each further step only the best rotation is kept and neighboring rotations are resampled on a local grid with 27 grid points, including the currently best rotation as center point. Subsequently, two stages are carried out: a coarse neighborhood search and a fine neighborhood search. The initial grid resolutions of the coarse and fine search are denoted by ρ_1 and ρ_2 in the following. Both in the coarse and in the fine search the grid resolution is adapted depending on whether a better rotation has been found in the last step or not. If a better rotation has been found in the coarse search, the local grid sampling is repeated around that rotation with the initial coarse resolution ρ_1 ; otherwise the local search

TABLE 1: Mean computing time \bar{t} , mean rotational error $\bar{\rho}$ in degree, and number of successful estimations in percent $n(\rho)$. Entries are for sampling types det/randdet/rand.

(ρ_0, ρ_1, ρ_2)	(5, 3, 1)	(20, 10, 5)	(50, 20, 10)
\bar{t} [s]	102/134/179	10/9/11	1.7/1.8/ 0.5
$\bar{\rho}$ [deg]	77/43/ 11	15/13/7	73/ 63/88
sr [%]	45/54/ 95	93/95.4/ 98.6	60/ 64.5/20

is repeated with with a doubled resolution. If a maximum resolution is exceeded, the coarse search is aborted. Then, the fine search is started with the initial resolution ρ_2 . If a better rotation can be found, the local grid sampling is repeated around that rotation with the initial fine resolution ρ_2 ; otherwise the current resolution is cut in half and the local search is repeated. If the resolution falls below a minimum, the search is finished.

The second sampling strategy (denoted by randdet) is a mixture of discrete and random sampling: the initial samples are drawn as before. The neighborhood search is performed by randomly sampling 27 new rotations in the neighborhood (with adapted radius as before) of the best rotation of the previous step.

The third method is denoted by rand and is an important resampling approach: in every step samples are drawn randomly and resampled according to the scores. Therefore, not only in the neighborhood of the best rotation new samples are drawn, but in the neighborhood of all rotations. In order to assure comparability, initially the same number of rotations is sampled as in the first method. This number is defined by the initial resolution ρ_0 . In each further step, the number of samples and the neighborhood radius are reduced by a factor of 0.5.

As convergence to the correct rotation is assumed, the resolution in the scoring (the bin width of table or the ball radius of nb in Section 6.1) is also adapted for each sampling stage. Sampling with det comprises three stages: an initial, a coarse, and a fine search. The bin widths of table in these stages are denoted as τ_0, τ_1, τ_2 , respectively, and are decreasing: $\tau_0 > \tau_1 > \tau_2$.

The results concerning sampling and scoring strategies are depicted in Tables 1 and 2 and can be summarized as follows: if the particle number is chosen properly, the proposed sampling rand outperforms the original method det concerning robustness and accuracy. The proposed scoring method nb yields slightly better results in accuracy and robustness compared to the original table. Though a computationally expensive neighborhood search in an octree has to be performed and thus computation time is much higher compared to computing a discretized 3D vote. Therefore, nb can only be recommended if computation time is irrelevant. Therefore, table and rand are used in the remainder.

Finally, a comparison of MCR to alternative approaches implemented in PCL was performed. These are the Hough voting method of Tombari and Di Stefano [29], the Geometric Consistency (GC) approach used by Aldoma et al. [23], and the SAC-IA method of Rusu et al. [22]. Figure 11 shows

TABLE 2: Mean computing time \bar{t} , mean rotational error $\bar{\rho}$ in degree, and number of successful estimations in percent. Entries are for scoring methods table and nb.

(τ_0, τ_1, τ_2)	(1, 0.8, 0.5)	(3, 2, 1)	(5, 3, 1)
\bar{t} [s]	10/60	5/41	5/69
$\bar{\rho}$ [deg]	53/56	47/35	39/36
sr [%]	61/ 63	65/73	69/73

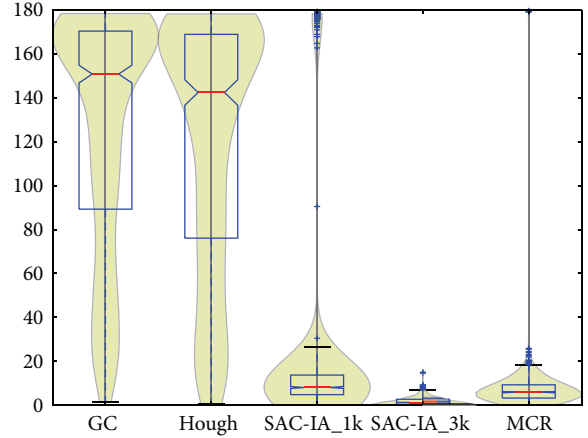


FIGURE 11: The rotational errors for the Stanford Bunny, comparison with methods available in PCL. A standard boxplot from python's matplotlib-package is overlaid with a density plot, in order to capture the multimodal distribution of the errors.

violin plots [52] of the results, including two versions of SAC-IA, first with the default 1000 (1k) iterations, then with an increased number of 3000 (3k) iterations. Notably, SAC-IA always runs up to the maximum allowed number (1000 or 3000 in this test). It maximizes the overlap quality directly and therefore performs well with low-dimensional features that are not as descriptive. Typically an increased number of iterations is needed to outperform MCR. MCR and SAC-IA clearly outperformed both GC and Hough when using the scalar features.

7.2. Experiments with Real Data. In the following, selected results with real data from different 3D sensors are presented, showing the effectiveness of MCR under hard conditions like small overlap areas and noisy data.

7.2.1. Registration with Laser Striper. In the first experiments, we use the DLR Multisensory 3D Modeler [53] that comprises a laser-line projector and a stereocamera system, implementing a laser-stripe range sensor. It is attached to a 6 DOF industrial robot, the KUKA KRI6-2.

Registration of a Wooden Workpiece. Figures 12(a), 12(b), 12(c), and 12(d) show the feature points and the reduced feature points used to match two scans of a wooden workpiece. Figures 12(e) and 12(f) show the result: the complete surface

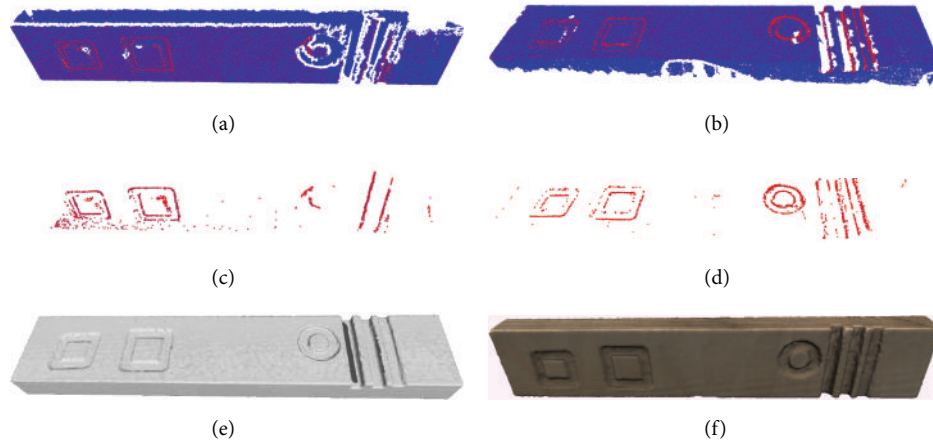


FIGURE 12: Modeling a wooden workpiece. Two sets of feature points ((a) and (b)), the reduced feature points ((c) and (d)), and the result after registration ((e) and (f)): the remeshed model (e) and the whole model with textures (f).

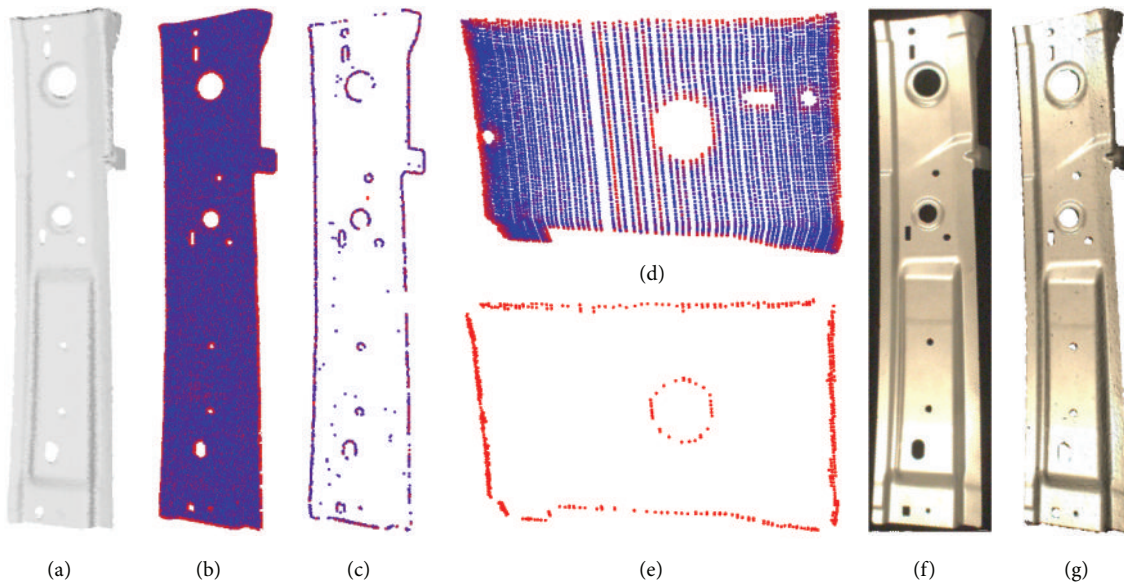


FIGURE 13: Modeling a steel sheet. ((a), (b), (c)) Reference triangle mesh, feature points, and reduced feature points (from left to right). ((d) and (e)) Partial scans of the object—feature points (d) and reduced feature points (e). ((f) and (g)) Photo/texture of the object and remeshed and texture-mapped (from left to right).

model after registration and remeshing on (e), with mapped textures on (f).

Registration of Steel Sheets. In this case, the problem is to map texture to a high quality 3D surface model. In order to overcome the sensor's problems with reflecting materials the object is sprayed with developer, losing the original texture information but gaining a high quality surface model. Afterwards a low quality surface model with correct texture (gained from the unsprayed object) is registered to the first model. Figure 13 shows (from (a) to (g)) the complete reference surface model, the feature points calculated from the point cloud (EVQ23, Section 3.2), and the reduced feature points (rightmost bins have been removed). In the middle the feature points before (d) and after (e) reduction of the second measurement are depicted. The texture information

(belonging to the second measurement) from a moncamera shot and the result, that is, the reference model with mapped texture, are shown in Figures 13(f) and 13(g).

7.2.2. CAD Model Registration with ToF Camera Images. Operators in telepresence systems (as in [54]) profit from semiautonomous functions, like grasping tools for manipulation. Here, we depict a use case where our method is employed to help the humanoid robot “Justin” grasp a power screwdriver, as shown in Figure 1. The pose estimation of the screwdriver is based on one frame of a SwissRanger SR4000 ToF cam fixed on one side of the torso. As the mounting is known, the screwdriver's position on a table is assumed to be known up to 10 cm. Further, two rotational degrees of freedom (DOF) are known up to a tolerance of 40° and 120° ,

respectively. In this setup the major challenge is the high noise of the ToF cam (Figure 4), which additionally produces an inaccurate extrinsic calibration.

The template feature points for global registration are calculated from CAD data. As the features we want to extract define a gap at the handle, concave regions have to be extracted. Therefore, points on the surface polygons of the CAD data are sampled and remeshed initially. On the resulting homogeneous triangle mesh, the MaNC(3) is calculated and the leftmost bins of the feature histogram are removed according to Section 3.4, until 30% of the feature points remain. Figure 5 shows the template point cloud model and the calculated feature points.

From the incoming data, that is, one frame of the ToF cam, the feature points are extracted as outlined in Section 3.5. Prior to that, the following strategy is used to handle the camera data. Considering the roughly known pose, the acquired depth image of the whole scene and its corresponding intensity image are 3D-cropped to the surrounding of the table first. Then, the table top is removed from the depth image with the help of a plane detection. Finally, the edges described in Section 3.5 are extracted to be used as features; see Figure 6.

As feature values do not correspond between template and measured data, only one feature class comprising all points is used. This works well as long as two conditions are met. First, visual and geometric features have to coincide, which is often the case with technical products. Second, in order to keep the number of false matches low, there should be not too many different edges.

Due to the possible incorporation of prior knowledge into MCR, the initial sample consists of 549 rotations on a grid, where the two rotational DOFs are sampled in steps of 2° and 5° , respectively. The pose estimation with MCR is fine adjusted with a subsequent ICP. The robot fulfills its task fluently, as the overall computing time is between 1 and 5 seconds. An example of a successfully fitted template in the original depth image is shown in Figure 1.

In order to prove the methods competitiveness it is tested against alternative methods as in Section 7.1. Here, 1000 random poses are tested using MCR and the methods from PCL. Obtaining 1000 real scans with ground truth pose is problematic. Therefore, the model is aligned to a scan manually. In order to get 1000 different poses, 1000 random rotations are applied to it. Figure 14 shows violin plots of the resulting rotational errors. Summarizing, MCR outperforms the other methods clearly. Note that the seemingly good results of SAC-IA are only due to rejected estimations outside the prior knowledge region. Only in 1% of the cases an acceptable estimation is obtained. The mean computation times for SAC-IA, SAC-IA_full, and MCR are 0.47 s, 3.54 s, and 0.14 s, respectively, with under 0.01 s added for MCR_ICP (if ICP correctly converged).

8. Experiments with SMCR

In this section, the main experiments with SMCR are carried out on an industrial robot and the method is applied to

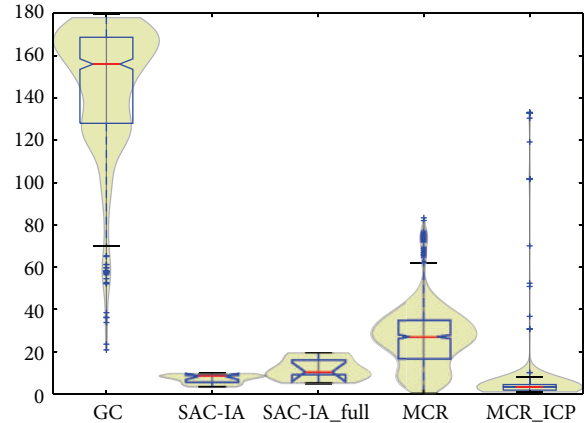


FIGURE 14: Violin plots of the rotational errors for the screwdriver compared to methods from PCL. Note that the Hough voting method never finds a solution, GC only in around 66% of the cases, and SAC-IA in around 1% of the cases (or less, when all points are used). MCR gives a result in all 1000 runs.

autonomous object modeling. In the following, the utilized hardware and experimental setup are described. Then, the results for SMCR are discussed, followed by the integration of SMCR with autonomous object modeling and an evaluation of its performance. Further, the application in mobile robot localization is depicted.

8.1. System Setup. Here, the utilized hardware, test objects, evaluation criteria, and parameters are described.

8.1.1. Hardware. For the experiments a 6 DOF industrial robot, the KUKA KR16-2, with mounted laser striper is utilized (see Figure 15). For the KR16-2, the absolute positioning error is in millimeter range. The streaming Monte Carlo registration and the autonomous object modeling are run on an external computer with Quad Xeon W3520 2.67 GHz CPUs and 6 GB RAM as the KUKA Robot Control 4 (KRC4) is not designed for additional modules. The communication between KRC4 and the external PC is performed at 250 Hz using the KUKA Robot-Sensor Interface. The laser striper is a Micro-Epsilon ScanControl 2700-100 which obtains a stripe of 640 depth points in a range of 0.3 m to 0.6 m at 50 Hz with a maximum measuring error of approx. 0.5 mm.

8.1.2. Test Objects and Data. All experiments in Sections 8.1–8.3 are performed for three objects: a Zeus bust, a bunny, and a wooden chevron (see Figure 16(a)). These represent different application domains, namely, cultural heritage, household, and manufacturing. The approximate height of the Zeus bust is 22 cm and that of the bunny and chevron 18 cm. Figure 16(b) shows the calculated features for these objects and Figure 16(c) the autonomously acquired 3d models (see Section 8.3).

8.1.3. Evaluation Criteria and Parameters. Similar to Section 7, the evaluations are done with respect to the median of rotational and translational error, denoted by m_t and m_R ,

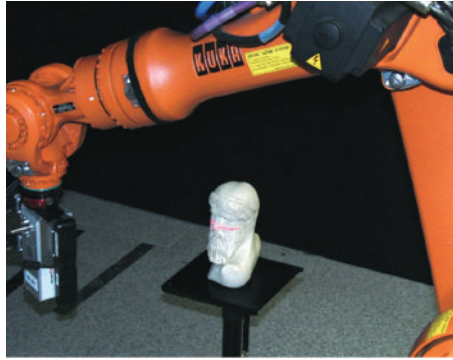


FIGURE 15: An industrial robot with attached laser striper performs a scan of an object placed onto a pedestal. The features are calculated in a real-time stream and are used to estimate the object's pose.

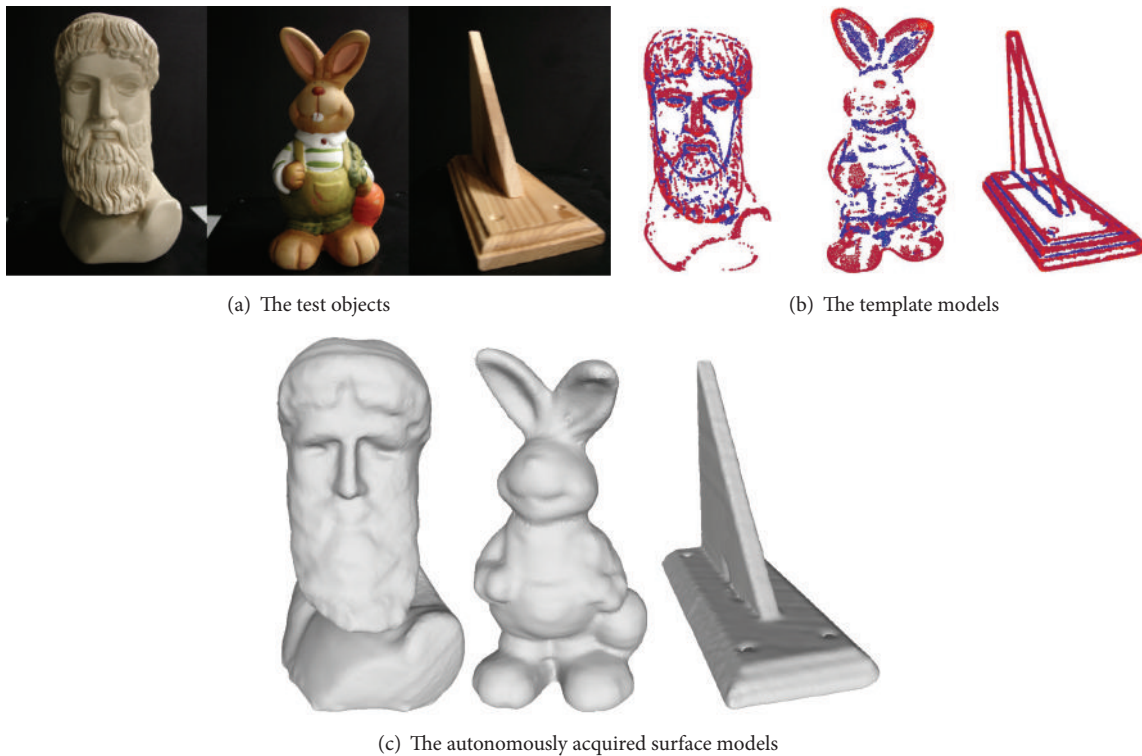


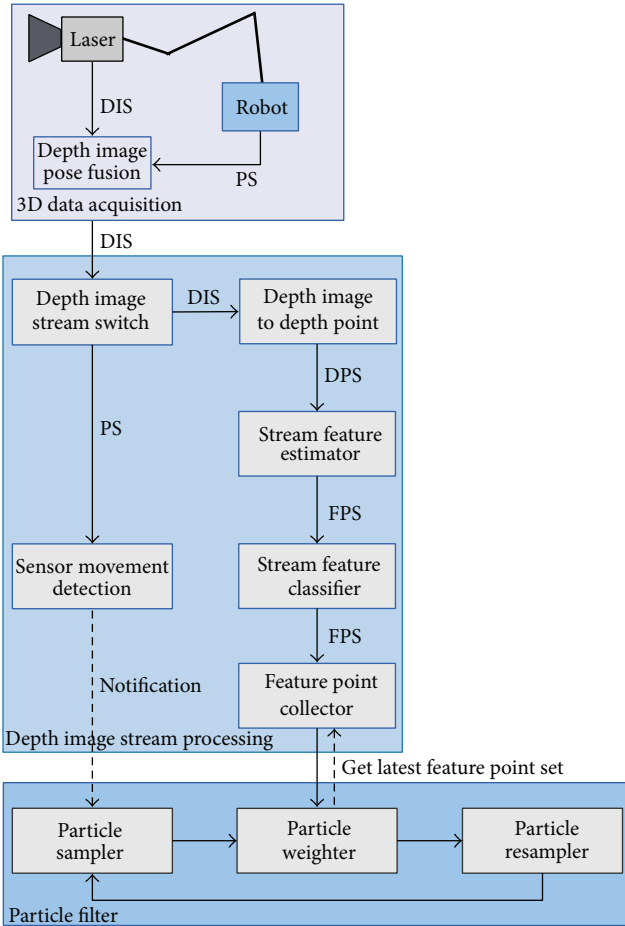
FIGURE 16: Zeus bust, bunny, and wooden chevron object (from left to right): the test objects (a) with corresponding template models (b) used for the experiments and surface models (c) acquired during autonomous object modeling (see Section 8.3.5). Colors in the template model describe the different feature classes: light/dark red occur in convex regions, blue/purple in concave regions (plane regions are removed).

respectively, and the success rate sr . Here, a success is defined, if the final error in translation and rotation is below 8 mm and 8° , which are tighter bounds than in Section 7.

As stated in Section 4, some parameters are adapted (by a factor of 0.8) in the iterative process. If not stated otherwise, we use a maximum number of 200 and a minimum number of 20 particles. The maximum scoring radius r_{\max} starts with 40 mm and is bounded from below by 4 mm. Neighborhood sampling of translations starts with a radius of 10 mm and is bounded by 1 mm.

8.2. Pose Estimation with SMCR. In this section, first an overview of the different modules of SMCR and its integration are given. Then, the influence of prior knowledge is investigated. Finally, the concept of SMCR is verified by comparison to offline global methods. Details on the results can also be found in [2].

8.2.1. Overview. The proposed SMCR method integrates three modules: the *3D Data Acquisition*, the *Depth Image*



DIS: depth image stream FPS: feature point stream
 DPS: depth point stream PS: pose stream

FIGURE 17: The SMCR workflow is divided into three main modules: the 3D Data Acquisition, the Depth Image Stream Processing, and the Particle Filter. Each module contains different components.

Stream Processing, and the Particle Filter module, as depicted in Figure 17.

The 3D Data Acquisition module synchronizes the depth images from the laser scanner with the pose information from the robot [55]. Therefore, the resulting depth images contain the sensor pose in robot coordinates, so that they need not be aligned to each other.

The depth image stream is handled to the Depth Image Stream Processing module. Here, the Depth Image Stream Switch component is dividing the stream again into a pose stream and a depth image stream. The pose stream is ending in the Sensor Movement Detection component, where the pose information is evaluated. If the translational or rotational difference to the last pose used for an update exceeds some predefined thresholds th_t and th_R , respectively, the sensor has moved significantly, and the particle filter is triggered to perform an update.

The depth image stream serves for feature point calculation. The depth images are converted to depth points and passed to the Stream Feature Estimator component (see

Section 3). The resulting feature point stream is handled to the Stream Feature Classifier component, which classifies the features according to the class borders of the template. Finally, the feature point stream is collected in the Feature Point Collector component from which the Particle Weighter component acquires the latest feature points on demand.

The Particle Filter module itself contains the Particle Sampler, the Particle Weighter, and the Particle Resampler components and works as described in Section 4. The Particle Sampler component starts sampling the particles when it is notified by the Sensor Movement Detection component. It performs a neighborhood sampling of the transformation particles (see Section 5). When finished, the Particle Weighter component acquires the latest feature points from the Feature Point Collector component. The particle weighting is carried out according to Section 6. After weighting, the particles are resampled with an importance resampling step by the Particle Resampler.

8.2.2. Data and Parameters. The method is evaluated with 10 different scan paths of the Zeus bust, 8 scan paths of the bunny, and 5 scan paths of the chevron. The difference in scan path numbers is due to the different object shapes and the chevron's symmetry. In order to ensure independent results, the scan paths are placed all around the objects. Each test is repeated 100 times to achieve a meaningful number of test runs. The tests are performed once on the real robot and the scan data is saved, as it is not feasible to repeat the whole scanning process so often. Then, the repeated tests are performed on the saved data.

The scans are registered to surface models, which are generated with a commercial 3D modeling system in advance. Feature points calculated from these models are depicted in Figure 16(b). Each feature point set is classified with 5 classes and the middle class is removed. The reduced feature point sets serve as template models and are used for registration during the laser scans. The template models of the bunny, the chevron, and the Zeus bust consist of 6714, 13075, and 8771 feature points, respectively.

The robot's pose error during a scan is usually negligible, concerning the quality of the acquired 3D models. Nevertheless, between two different scans, significant differences of robot configurations lead to considerable pose errors between the acquired 3D scan data. In between different scans, there typically occur gaps of up to 3 mm. Therefore, for each scan, a ground truth estimation is necessary because the resulting pose estimation accuracy is in the range of millimeters.

The ground truth estimations are calculated by utilizing MCR, followed by an ICP working on all acquired raw points. Correct results are assured by a visual inspection by a human operator. Moreover, the coordinate root mean square error after the ICP is checked to be lower than 0.2 mm.

8.2.3. Influence of Prior Knowledge. The prior knowledge about the searched transformation is separated into a translational and a rotational part. The translational part is expressed in terms of a cuboid. The volume of that cuboid is denoted by $V(t)$ in this section. The rotational part is explained by a

TABLE 3: sr , m_t , and m_R for different initial a priori knowledge. Fifth row: only rotations about the z -axis are sampled.

$V(t)$	r_p	sr	m_t/m_R
0.48 L	10°	0.69	3.8/4.6
0.48 L	20°	0.71	3.9/4.0
0.48 L	45°	0.68	4.0/4.2
0.48 L	90°	0.74	3.2/4.0
0.48 L	90° (z)	0.93	0.7/1.0
1.22 L	10°	0.69	4.1/4.4
1.22 L	20°	0.64	4.3/5.9
1.22 L	45°	0.63	4.0/5.5
1.22 L	90°	0.71	3.5/4.1
4.00 L	45°	0.48	7.2/8.1

mean rotation and a maximum rotational deviation from it. Additionally, the rotation axis can be assumed to be fixed in some cases, for example, if the object has been turned around the z -axis. The maximal difference from the mean rotation is denoted r_p . If not stated otherwise, the initial translations are sampled in a cuboid with an extension of 16 cm \times 10 cm \times 3 cm, which corresponds to the approximate position on the scanning pedestal.

In order to examine the influence of the prior knowledge, one scan of the bunny is registered to a ground truth surface model, which was acquired with a commercial scanning system. Table 3 shows the results. The initial sampling radius for the rotation seems to have little effect on the success rate and on m_t/m_R : for both fixed $V(t)$ of 0.48 L and 1.22 L neither the success rate nor the errors are clearly increasing or decreasing with increasing radius. The volume of the cuboid for the initial translation has a clear influence if increased, as the last row shows. Both the success rate and the errors are significantly worse for a translational volume of 4.00 L compared to that of 0.48 L and 1.22 L. This confirms the suitability in autonomous 3D modeling because there is often good a priori knowledge about the position of the object, whereas the rotation cannot be told beforehand. In our scenario the rotation axis is known approximately, and often the rotation angle is restricted within a known range.

8.2.4. Comparisons with Offline Methods. This section shows the comparison of SMCR to MCR and SAC-IA. We chose SAC-IA as reference because it performs best in the previous experiments in Section 7. In these experiments high quality scans are used. Therefore, SAC-IA yields the best results when applied to the complete point cloud (downsampled to a density of 3 mm) with the FPFH feature, a multidimensional feature [22], which has been used in these experiments. As SAC-IA is not taking any prior information into account, it needs to perform many trials, resulting in long computation times. Introducing prior information is possible, but the method slows down a lot by this, as discussed in Section 7.2.2. Table 4 shows the result for 100 runs with all scans. Again, SAC-IA works well with 3000 iterations (results can be improved by using even more trials, but we do not find that feasible). Z_i, B_i, C_i denote the scans of the Zeus bust, the

bunny, and the chevron, respectively. The highest success rate and lowest translational and rotational error are highlighted for each scan. Overall, MCR resulted in the lowest pose errors. However, MCR took up to 30 seconds and SAC-IA up to a few minutes in the worst case whereas SMCR did not require any additional computation time.

Figure 18 highlights one of the cases where both SMCR and MCR perform poorly but SAC-IA performs relatively well, in a violin plot in (a). The Z_6 scan captures a smooth surface at the back of the statue's head (on Figure 18(b)), which contains relatively few feature points. This results in larger errors of MCR and SMCR because they rely on local features. Contrary, SAC-IA uses all the points and FPFH and manages to find a good transformation in 44% of the cases. However, the distribution of the errors shows that while SAC-IA performs better, it fails completely in many cases, which influences the median not too much. In order to increase the performance of MCR and SMCR in such cases, more points could be considered. Though, more points lead to an increased computation time and introduces more uncertainty (due to a higher number of false matches).

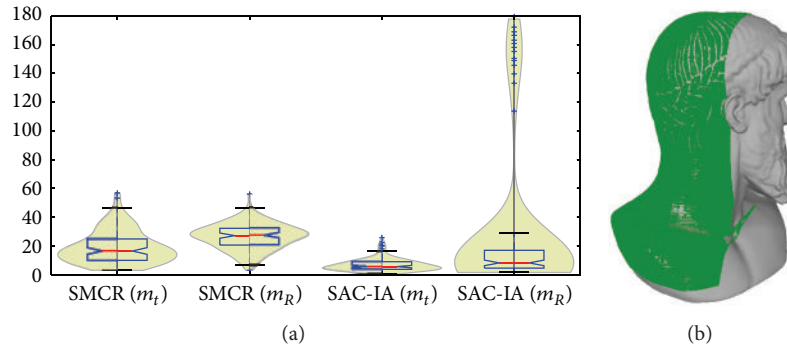
8.3. SMCR in Autonomous Object Modeling. In this section, first an overview is given on how the SMCR workflow (see Figure 17) is integrated into autonomous object modeling. Then, the convergence behavior and the performance of the different streaming registration variants and ICP are compared. Finally, the accuracy of autonomous modeling with integrated SMCR and repositioning is compared with the previous method.

Similar investigations can be found in [2, 3]. In contrast to those, here the particle optimization and the convergence behavior and the convergence criterion are investigated in detail. Basically, only the investigations of the uniform and Gaussian/Bingham sampling are covered by the previous publications. But for the sake of comparability with our new methods, we recomputed these tests as well in this work.

8.3.1. Overview. In this section, the integration of SMCR into the autonomous 3D modeling approach (see [4]) is presented. As the SMCR workflow has been described in detail in Section 8.2.1, here we concentrate on the modeling part as can be seen in the overview in Figure 19. Initially, an arbitrary laser scan of the unknown object is performed with the robot-sensor system (see Figure 15). The corresponding depth image stream contains the robot's pose information and is handled to three modules: *Mesh Update*, *Probabilistic Space Update*, and *Feature Calculation*. The features are calculated in real-time but will be used later for registration after the object has been repositioned. After updating the mesh and the probabilistic space, it is checked if the triangle mesh (surface model) has reached the desired quality, that is, if there are no holes except for not scannable parts. Then, *Next-Best-Scan Planning*, *collision-free Motion Planning*, and further laser scans are performed repeatedly, until the quality is reached. The mesh enables planning possible scan path candidates and selecting a Next-Best-Scan, in order to reach the desired surface model quality. The probabilistic space is a volumetric

TABLE 4: 100 runs: sr , m_t , and m_R and the mean computation time \bar{t} for SMCR, MCR, and SAC-IA.

Data	SMCR		MCR			SAC-IA		\bar{t}
	sr	m_t/m_R	sr	m_t/m_R	\bar{t}	sr	m_t/m_R	
Z_1	0.51	5.7/5.8	0.60	5.9/6.0	8.6	0.04	11.3/15.6	120.4
Z_2	0.59	6.0/5.7	0.92	3.8/3.3	14.1	0.26	7.9/11.1	113.1
Z_3	0.49	7.1/7.8	0.58	4.5/6.9	14.4	0.24	9.1/9.2	109.0
Z_4	0.50	7.0/7.0	0.80	2.7/4.2	14.1	0.47	5.5/7.8	131.7
Z_5	0.25	9.4/13.5	0.22	17.8/52.4	10.4	0.92	2.5/3.4	125.8
Z_6	0.00	16.5/27.1	0.00	19.2/176.9	7.5	0.44	5.6/8.2	114.1
Z_7	0.34	9.3/9.5	0.00	20.6/169.5	7.6	0.58	5.2/6.0	114.8
Z_8	0.00	28.3/22.0	0.82	6.2/3.8	8.9	0.11	9.3/12.8	143.2
Z_9	0.41	7.6/7.6	0.69	4.1/4.8	10.3	0.05	21.5/24.8	80.4
Z_{10}	0.04	27.3/ 11.8	0.00	23.7/124.3	6.9	0.06	14.3/143.3	79.0
B_1	0.77	3.7/4.4	0.90	2.9/2.3	6.1	0.02	9.8/18.0	28.4
B_2	0.71	3.9/4.0	0.57	2.2/7.4	7.4	0.06	8.2/13.0	30.3
B_3	0.61	4.6/5.8	0.93	2.4/2.5	6.8	0.42	5.4/8.4	28.5
B_4	0.44	3.9/9.2	0.78	2.7/6.1	4.3	0.22	6.8/11.9	23.0
B_5	0.26	6.7/10.5	0.22	5.3/9.0	4.1	0.70	4.1/5.8	28.8
B_6	0.15	13.9/10.1	0.58	4.6/4.9	4.4	0.20	7.5/10.4	27.7
B_7	0.69	3.8/5.6	0.37	4.2/11.2	4.4	0.17	7.5/12.3	27.6
B_8	0.43	6.0/ 8.5	0.45	4.6/8.7	4.7	0.02	11.0/28.0	23.4
C_1	0.15	14.8/10.5	0.66	3.2/4.1	30.8	0.15	13.9/8.1	82.5
C_2	0.33	11.8/7.6	0.96	1.4/1.7	23.7	0.14	13.3/9.7	77.0
C_3	0.22	14.7/6.6	0.99	1.6/3.9	16.4	0.04	19.2/13.2	63.5
C_4	0.17	18.7/8.6	0.52	7.5/4.3	15.9	0.01	39.7/170.0	59.2
C_5	0.06	21.8/11.0	0.32	11.8/6.0	15.1	0.00	43.0/26.8	54.5
Units		mm/deg		mm/deg	s		mm/deg	s

FIGURE 18: (a) Violin plots of exemplary translational errors in mm and rotational errors in degree for SMCR (left) and SAC-IA (right) for scan z_6 . (b) Scan z_6 (green) on ground truth model of Zeus.

model considering sensor uncertainties and giving a probability of occupancy for each voxel. It is used for exploration by *Next-Best Scan Planning* and collision avoidance during *Motion Planning*. For more details regarding the autonomous modeling and its modules we refer to [4].

As soon as the desired mesh quality has been reached, the object is repositioned in order to model previously occluded object parts. This is done by rotating it onto one of its sides. Then, a laser scan is performed along the region of interest. Again, the *Feature Calculation* is carried out on-the-fly. Synchronously, the *Particle Filter* component iteratively

performs a neighborhood sampling and weighting of the particles with the incoming feature points. After the laser scan has finished the pose estimation is instantly available. Finally, an ICP is used for fine registration, which results in a precise transformation between the original object position and the object position after repositioning. Then, the autonomous modeling is continued until a complete model is generated. In order to be able to model the object within the same coordinate system, all further generated laser scans are transformed by the resulting transformation from the registration.

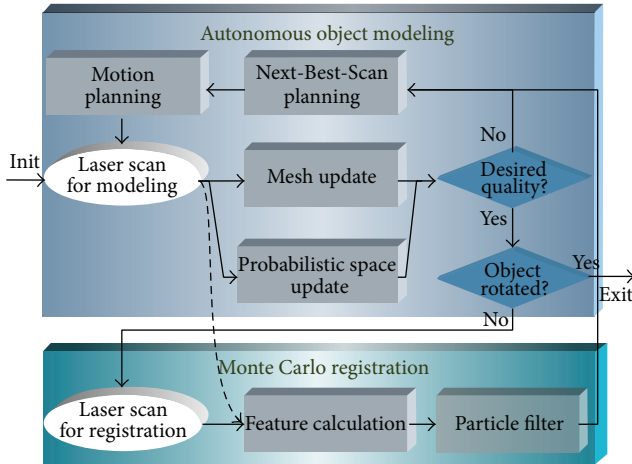


FIGURE 19: Overview of the integration of autonomous object modeling with streaming Monte Carlo registration. Gray boxes represent modules, white ovals robot-sensor system actions, and blue diamond boxes decisions.

8.3.2. SMCR and ICP with Partially Overlapping Submodels.

In order to show that local methods like ICP cannot be applied, we register two partially overlapping scans of the Zeus bust to each other, which poses a higher challenge than registering a submodel to a complete object model with low noise. For this purpose, we record 20 different pairs of overlapping scans (see Figure 20 for an example). To one of the scans in each pair we apply 10 different random transformations with translational and rotational variations. The translation vectors have a maximal norm of 20 mm and the rotations a maximum rotation angle of 45° , 90° , 120° , or 180° around the z -axis. The prior knowledge in SMCR is set accordingly, denoted with $Z45X$, $Z90X$, $Z120X$, $Z180X$, respectively in Table 5. Z stands for the object Zeus and X presents the different methods described below. After the SMCR, we apply an ICP with a small search radius (20 mm) and few iterations for fine fitting, denoted by SMCR-ICP. We compare the results to a pure ICP with a bigger search radius (50 mm), simply denoted by ICP. For each rotation range we test the original Uniform neighborhood sampling U , the proposed Gaussian sampling G , and the Gaussian sampling with Optimization step O , denoted by a capital U , G , or O_i in the data names, respectively. For instance, $Z90N$ denotes the case of uniformly sampled rotations with a maximal rotation angle of 90° for the Zeus bust, and $B120O_5$ denotes the case of optimization in every 5th step (normal/Bingham sampling) and a maximal rotation angle of 120° for the bunny. In the table, the highest success rate and lowest translational and rotational error are highlighted for each rotation angle over all methods. Overall, the accuracy in translation and the success rate are increased for SMCR when applying the optimization step. The rotational accuracy is not generally increased. The effects appear in all rotation ranges. Concerning pure ICP, it becomes clear that, with increasing rotation range, its performance significantly decreases, both in accuracy and reliability. SMCR-ICP outperforms pure ICP clearly, both in reliability and accuracy. Further,

the performance of pure SMCR is also significantly better than ICP. The frequency of the optimization steps has no clear effect on the results, neither on pure SMCR nor on the combination with ICP. Uniform sampling yields slightly better results than Bingham/Gaussian sampling.

Success rates of about 0.6 of pure SMCR appear to be pretty small for two reasons: on the one hand, the parameters are not tuned for the data sets. On the other hand, the partially overlapping scans are harder to register as it seems at first glance. The most descriptive features are not easy to scan and very similar features are spread over the object. Moreover, the descriptive features appear more or less randomly in the data sets because the scan paths are chosen arbitrarily. The results in the preceding sections show that with this kind of data other state-of-the-art methods perform even worse, even when registering to a complete high-precision ground truth surface model.

8.3.3. Convergence Behavior for SPFR, SMCR, and SMCRO.

In this section, the convergence of SMCRO, SMCR, and SPFR is investigated. In order to account for the application in autonomous 3D modeling, we first autonomously acquire a more or less complete model of the bunny and the Zeus bust (without bottom part). Then, we reposition them on the scanning pedestal and acquire one scan manually. With this scan, 1000 repetitions for pose estimation are performed.

The ground truth estimation is calculated by utilizing MCR, followed by an ICP working on all acquired raw points of the ten subscans. Correctness is assured by visual inspection of a human operator.

Therefore, we repeat estimations for one scan path of the bunny and the Zeus bust 1000 times and calculate the medians of the translational and rotational errors in each update step, as depicted in Figure 21. In the case of SMCRO, the optimization is carried out in every step and every 2nd, every 5th, or every 10th step. Clearly, SMCR yields better convergence behavior than SPFR, which does not reach the success criterion at all. SMCRO in turn converges faster than SMCR, and the faster the more optimization steps are performed. Note that, at the update steps, the error medians often visibly drop down.

However, the optimization needs to be carried out with caution as, in individual cases, for too early or too many optimization steps the method may tend to converge to the wrong transformation, especially for objects with many symmetries. Therefore we started optimization not before the 5th step in any case.

8.3.4. SMCR and ICP in Autonomous Modeling.

In this section, an extensive evaluation of SPFR, SMCR, and SMCRO and comparison to ICP are performed. Therefore, a more or less complete model (without bottom part) of the object is autonomously acquired initially. Then, the object is repositioned on the scanning pedestal and 10 different single scans are performed manually. Each of the manual scans is transformed by ten different random transformations, giving a total of 100 different test scans for each object. These are registered to the corresponding previously acquired complete

TABLE 5: m_t , m_R , and \bar{t} for SMCR with Gaussian (G)/uniform (U) sampling and with optimization (O) step and ICP for 200 tests of partially overlapping scans of the Zeus bust and 45°, 90°, 120°, and 180° rotations.

Data	SMCR		SMCR-ICP			ICP		
	m_t/m_R	sr	m_t/m_R	sr	\bar{t}	m_t/m_R	sr	\bar{t}
Z45G	6.3/5.0	0.5	1.7/2.5	0.7	2.1			
Z45U	6.3/6.3	0.5	1.5/2.7	0.7	2.0			
Z45O ₁	2.7/3.4	0.6	1.5/2.3	0.8	1.6	2.8/6.6	0.5	7.1
Z45O ₂	2.9/3.1	0.6	1.6/2.4	0.7	1.8			
Z45O ₅	2.7/3.2	0.6	1.4/2.4	0.8	1.7			
Z45O ₁₀	2.5/2.9	0.6	1.4/2.3	0.8	1.8			
Z90G	7.1/7.8	0.4	1.5/2.7	0.7	2.0			
Z90U	5.7/7.1	0.5	1.7/3.0	0.7	2.3			
Z90O ₁	3.1/3.8	0.6	1.7/2.5	0.7	1.7	12.6/41.9	0.3	7.2
Z90O ₂	3.2/4.7	0.5	1.6/2.4	0.7	2.0			
Z90O ₅	3.7/4.2	0.6	1.5/2.4	0.7	1.8			
Z90O ₁₀	3.8/4.5	0.5	1.4/2.3	0.8	1.9			
Z120G	6.0/5.3	0.5	1.6/2.8	0.7	2.2			
Z120U	5.7/6.9	0.5	1.6/2.7	0.7	1.9			
Z120O ₁	3.7/3.7	0.6	1.5/2.7	0.7	1.8	12.6/53.5	0.2	6.9
Z120O ₂	2.6/4.8	0.6	1.3/2.5	0.8	1.7			
Z120O ₅	2.7/2.7	0.6	1.3/2.3	0.8	1.6			
Z120O ₁₀	2.7/4.0	0.6	1.5/2.6	0.7	1.8			
Z180G	5.3/5.6	0.5	1.6/2.6	0.7	2.1			
Z180U	4.1/4.9	0.6	1.6/2.3	0.8	1.6			
Z180O ₁	2.9/4.0	0.6	1.7/2.2	0.7	1.8	15.2/71.6	0.2	3.7
Z180O ₂	4.0/5.6	0.5	1.6/2.5	0.7	1.9			
Z180O ₅	2.7/3.0	0.6	1.5/2.4	0.7	1.6			
Z180O ₁₀	1.8/3.4	0.6	1.4/2.4	0.7	1.7			
Units	mm/deg		mm/deg		s	mm/deg		s

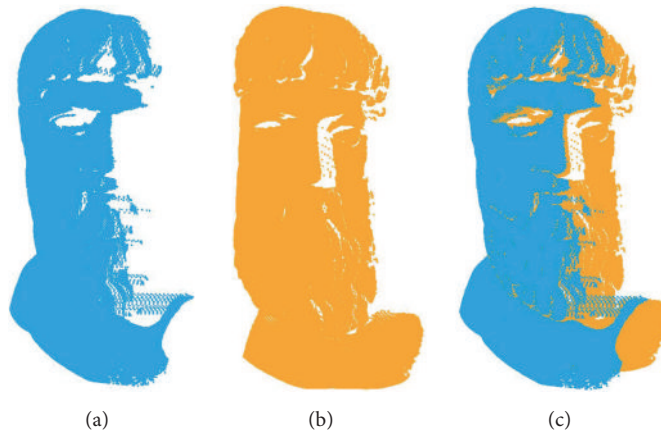


FIGURE 20: Two typical scans of Zeus bust before ((a) and (b)) and after alignment (c).

model. The ground truth estimation is calculated by MCR and an ICP working on the raw points of the ten subscons. Visual inspection of a human operator assures the correctness.

Goodness of Fit and Success Rates. Table 6 shows the results of the experiments concerning rotational and translational error as well as the success rate. In contrast to the experiments in

Section 8.3.2, the registration is performed based on the 3D model. Additionally we perform them for the bunny (denoted by a capital B) and the chevron (C) objects.

The results of the bunny and the Zeus bust clearly show that the proposed optimization step yields a higher accuracy and success rate. Moreover, in many cases, results get better with increasing optimization frequency. Finally, the errors

TABLE 6: m_t , m_R , and \bar{t} for SMCR with Gaussian (G)/uniform sampling (U), with optimization step (O), with aborting after convergence (C) and ICP for the Zeus bust (Z), the Bunny (B), and the Chevron (C) and 45°, 90°, 120°, and 180° rotations.

Data	SMCR		SMCR-ICP			ICP		\bar{t}
	m_t/m_R	sr	m_t/m_R	sr	\bar{t}	m_t/m_R	sr	
Z45G	3.1/2.3	0.7	0.4/0.4	0.9	1.6			
Z45U	2.8/1.8	0.7	0.3 /0.4	0.9	1.4			
Z45O ₁	0.9/0.4	0.8	0.4/0.4	0.9	2.6	0.9/1.8	0.6	10.9
Z45O ₂	1.1/0.6	0.8	0.4/ 0.3	1.0	2.9			
Z45O ₅	1.3/1.3	0.8	0.4/0.4	1.0	3.4			
Z45O ₁₀	1.3/1.1	0.8	0.4/ 0.3	0.9	3.5			
Z45C	1.3/0.9	0.9	0.3 /0.4	1.0	1.3		0.6	10.6
Z90G	5.3/4.3	0.6	0.5/0.5	0.8	1.7			
Z90U	2.7/1.8	0.6	0.4/0.4	0.9	1.7			
Z90O ₁	0.9/ 0.4	0.8	0.4/0.4	0.9	2.9	2.0/11.3	0.4	9.7
Z90O ₂	1.1/0.5	0.8	0.4/0.4	0.9	3.2			
Z90O ₅	1.5/1.4	0.8	0.4/0.4	0.8	3.5			
Z90O ₁₀	2.6/1.7	0.7	0.4/0.4	0.8	4.4			
Z90C	1.1/0.8	0.8	0.4/0.4	0.9	1.3		0.4	10.4
Z120G	5.3/4.6	0.5	0.5/0.5	0.8	1.8			
Z120U	4.5/3.0	0.6	0.6/0.4	0.8	1.4			
Z120O ₁	0.8/ 0.3	0.8	0.4/0.4	0.9	2.7	7.2/41.5	0.4	10.6
Z120O ₂	1.2/0.6	0.8	0.4/0.4	0.9	2.5			
Z120O ₅	1.3/1.4	0.7	0.4/0.4	0.9	2.4			
Z120O ₁₀	2.2/1.9	0.6	0.4/0.4	0.8	2.7			
Z120C	1.3/1.0	0.7	0.4/0.4	0.8	1.4			
Z180G	8.8/6.2	0.5	0.7/0.5	0.8	1.8			
Z180U	6.1/6.4	0.5	0.7/0.5	0.7	1.7			
Z180O ₁	1.0/0.5	0.7	0.4/0.5	0.8	2.4	10.8/51.9	0.3	10.3
Z180O ₂	1.3/0.9	0.7	0.4/0.5	0.8	2.4			
Z180O ₅	1.5/2.0	0.7	0.4/0.5	0.8	2.5			
Z180O ₁₀	2.6/2.0	0.6	0.4/0.5	0.8	2.7			
Z180C	1.6/1.1	0.6	0.5/0.4	0.8	1.5			
B45G	7.1/8.3	0.3	0.7/ 0.2	0.9	1.0			
B45U	6.7/7.0	0.6	0.7/ 0.2	1.0	0.9			
B45O ₁	0.7/0.3	0.9	0.7/ 0.2	1.0	2.0	0.6/0.3	1.0	2.6
B45O ₂	0.7/0.5	0.9	0.7/ 0.2	0.9	0.9			
B45O ₅	1.0/1.3	0.9	0.7/ 0.2	0.9	0.9			
B45O ₁₀	3.1/4.4	0.7	0.7/ 0.2	1.0	0.9			
B45C	0.8/0.8	0.9	0.7/ 0.2	0.9	1.0			
B90G	7.4/8.9	0.3	0.7/ 0.2	0.9	1.0			
B90U	6.7/8.5	0.4	0.7/0.3	0.9	0.9			
B90O ₁	0.7 /0.4	0.9	0.7/0.2	0.9	2.3	0.7/0.3	0.8	2.9
B90O ₂	0.7 /0.8	0.8	0.7/0.2	0.9	2.6			
B90O ₅	1.4/2.3	0.8	0.7/0.2	0.9	2.7			
B90O ₁₀	4.2/6.5	0.6	0.7/0.2	0.9	3.0			
B90C	0.8/0.8	0.8	0.7/0.2	0.9	1.1			
B120G	8.3/10.3	0.3	0.7/ 0.2	0.9	1.1			
B120U	6.9/7.6	0.4	0.7/ 0.2	0.9	1.0			
B120O ₁	0.7 /0.4	0.9	0.7/0.2	0.9	2.4	0.8/0.3	0.7	3.0
B120O ₂	0.8/0.7	0.8	0.7/0.2	0.9	2.4			
B120O ₅	1.9/2.7	0.8	0.7/0.2	0.9	2.9			
B120O ₁₀	4.2/6.1	0.6	0.7/0.2	0.9	3.3			
B120C	0.8/0.8	0.9	0.7/0.2	0.9	1.2			
B180G	9.3/11.9	0.2	0.7/ 0.2	0.9	1.2			
B180U	8.1/11.7	0.3	0.7/ 0.2	0.9	1.1			
B180O ₁	0.7/0.4	0.8	0.7/ 0.2	0.9	2.5	12.3/128.5	0.4	3.1
B180O ₂	0.8/0.7	0.8	0.7/ 0.2	0.9	2.6			
B180O ₅	1.6/2.2	0.7	0.7/ 0.2	0.8	2.7			

TABLE 6: Continued.

Data	SMCR		SMCR-ICP			ICP		
	m_t/m_R	sr	m_t/m_R	sr	\bar{t}	m_t/m_R	sr	\bar{t}
B180O ₁₀	5.4/9.0	0.4	0.7/0.2	0.9	3.4			
B180C	0.8/0.9	0.8	0.7/0.3	0.9	1.2			
C45G	12.3/0.9	0.4	2.2/0.9	0.6	1.9			
C45U	9.5/0.8	0.5	1.9/1.1	0.7	1.9			
C45O ₁	5.8/1.1	0.5	2.3/1.2	0.6	2.2	1.7/0.8	0.9	7.1
C45O ₂	5.1/1.3	0.5	2.2/1.2	0.6	2.3			
C45O ₅	8.9/1.8	0.5	2.2/1.2	0.6	2.5			
C45O ₁₀	5.8/1.5	0.6	1.8/1.1	0.8	2.6			
C45C	13.0/2.3	0.4	4.8/1.3	0.5	1.9			
C90G	11.4/1.0	0.4	2.2/1.0	0.6	1.9			
C90U	15.5/1.1	0.3	2.2/1.3	0.6	1.9			
C90O ₁	4.0/1.2	0.6	2.2/1.2	0.7	2.2	1.7/0.9	0.9	8.0
C90O ₂	4.9/1.1	0.6	2.2/1.2	0.6	2.3			
C90O ₅	7.0/1.9	0.5	2.6/1.3	0.6	2.5			
C90O ₁₀	9.8/1.4	0.5	2.2/1.1	0.6	2.7			
C90C	15.4/2.7	0.3	9.6/1.4	0.5	1.9			
C120G	15.3/1.1	0.3	7.4/1.3	0.5	1.7			
C120U	13.8/1.2	0.4	2.2/1.3	0.6	1.8			
C120O ₁	10.8/1.1	0.5	2.5/1.3	0.6	2.2	2.4/1.0	0.7	7.0
C120O ₂	6.4/1.2	0.5	2.2/1.2	0.6	2.2			
C120O ₅	11.8/1.8	0.4	3.9/1.3	0.5	2.3			
C120O ₁₀	14.2/1.4	0.4	2.6/1.0	0.5	2.5			
C120C	14.1/2.8	0.4	8.3/1.3	0.5	1.9			
C180G	18.4/5.2	0.1	12.3/2.1	0.4	1.7			
C180U	17.0/3.0	0.2	11.2/2.1	0.4	1.6			
C180O ₁	14.0/1.1	0.3	11.6/2.1	0.4	2.1	6.0/1.5	0.6	7.3
C180O ₂	13.5/1.5	0.4	9.5/ 1.3	0.5	2.3			
C180O ₅	16.5/2.3	0.2	12.0/2.0	0.4	2.5			
C180O ₁₀	13.6/1.9	0.3	2.5/1.3	0.6	2.6			
C180C	15.3/3.6	0.2	12.3/2.1	0.4	1.8			
Units	mm/deg		mm/deg		s	mm/deg		s

with SMCR-ICP are only slightly smaller than with the optimization in every step.

SMCR itself performs good, but accuracy is not comparable to the ICP (if both are successful). The proposed Gaussian sampling does not lead to higher accuracy or success rates. The accuracy as well as the success rate is not much influenced in the example of the bunny. The opposite is true for the Zeus bust. The pure ICP is working surprisingly good, especially with the chevron, though it gets unusable for rotation angles over 45° for the bust. SMCR-ICP works extremely good, even in cases when SMCR yields problematic results.

Concerning the chevron, the rotation is estimated pretty good by all methods and the translation very bad especially when allowing for large rotations. Note that the low median in the ICP results is misleading, as the success rates are pretty low, compared to the results with the Zeus bust and the bunny. An explanation could be that on the one hand the chevron has big flat surface areas which allow a robust estimation of rotations. On the other hand, this seems to allow the translation to slide along these areas, especially vertically along the triangular part. Additionally, there are a lot of spurious measurements, including the pedestal the object is placed on.

Convergence Criterion. The results obtained with the application of the convergence criterion (SMCRC) are denoted with a capital C as last letter in the table rows of Table 6; for example, Z45C denotes the experiments with the Zeus bust for an angle of 45° and with abortion due to convergence. In the tests, we used convergence thresholds of $c_t = 2$ mm and $c_r = 2^\circ$ (see Section 6.2.3). Concerning the Zeus bust and the bunny, Table 6 shows that the results with SMCRC are comparable with those of SMCRO₂, and after application of ICP comparable to SMCRO₁. Further, SMCRC clearly outperforms the variants with few optimization steps (SMCRO₅ and SMCRO₁₀). Concerning the chevron, SMCRC yields clearly worse results than SMCRO. Probably this is due to the geometry of the chevron, where initially a large translational error still allows for a very good matching of wrong correspondences.

8.3.5. Autonomous Object Modeling. Here, we compare the autonomous modeling results without repositioning the object as in [4] with the integration of the SMCR and repositioning of the object as presented in Figure 19. Therefore, the complete object modeling is performed 10 times each for

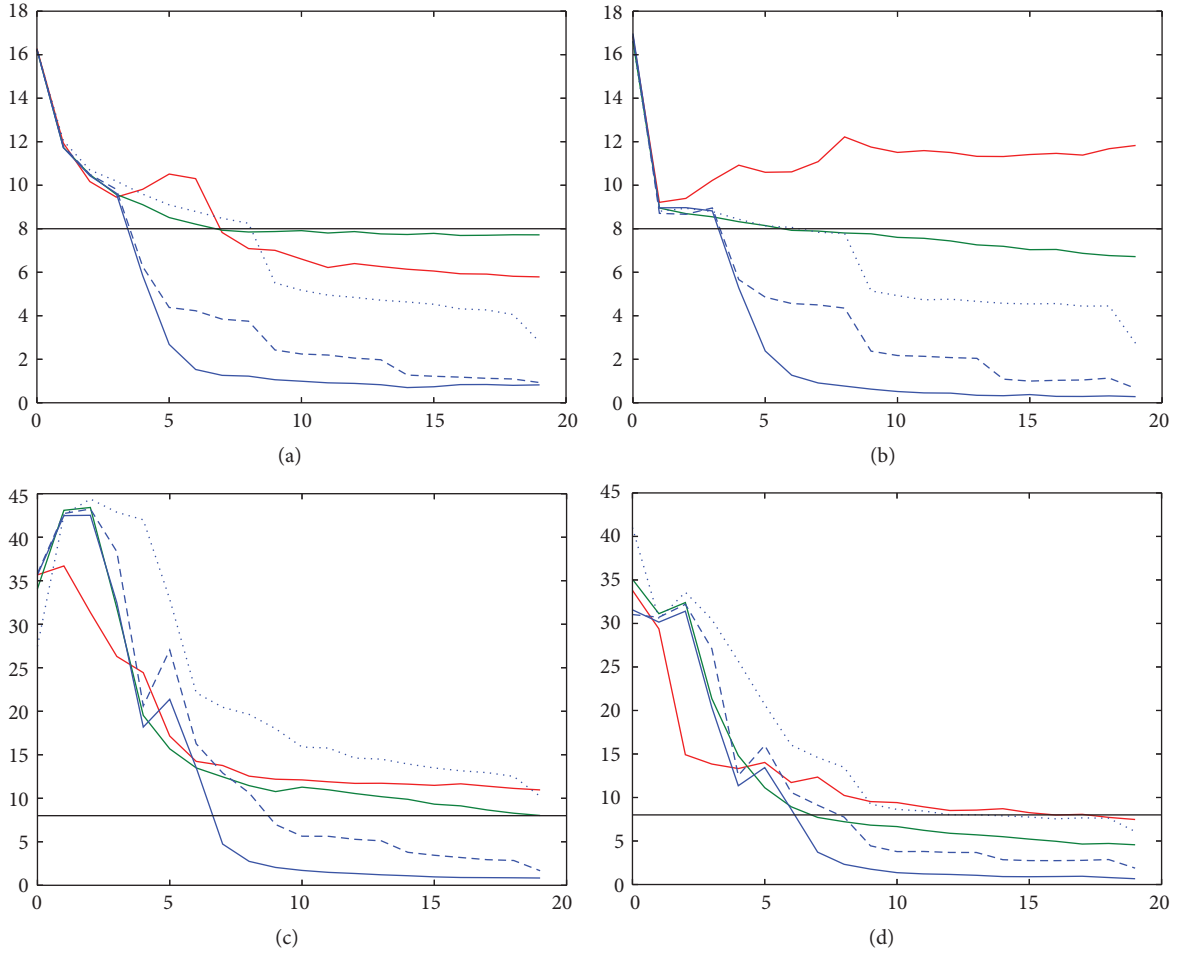


FIGURE 21: Exemplary error convergence of SPFR (red), SMCR (green), and SMCRO (blue) for 1000 runs on a bunny (top) and a Zeus (bottom) scan. Optimization is performed in every (solid) step and every 5th (dashed) and every 10th (dotted) step. Left: translational error in mm. Right: rotational error in degree. x-axis: step number. The black horizontal line represents the success threshold of 8 degrees or 8 mm.

the bunny, the Zeus bust, and the wooden chevron. For a comparison of the autonomous modeling method [4] with the other state-of-the-art methods concerning the algorithms for NBV planning, we refer to [56]. It has been shown that the NBV approach which plans NBVs based on the boundaries of the surface models and considers information gain and surface quality for the NBV selection outperforms the other methods.

During these experiments, the object is manually placed onto its side after the desired quality for the visible object parts has been reached. For the 10 runs, different arbitrary initial scans and variations in the repositioning object orientation are chosen. The average model completeness and coordinate root mean square (CRMS) error when comparing with ground truth models are given in Table 7. The completeness is evaluated by comparing a ground truth model with the generated triangle mesh. The CRMS gives a measure for the model error which is influenced by the fact that details in the object are not modeled perfectly as can be seen in Figure 16(c). The error is mainly a result of sensor noise, sensor calibration, and robot accuracy which for

the KUKA KR16-2 is in millimeter range. The completeness after repositioning is larger as the bottom parts have been filled. Figure 22 shows exemplarily for the Zeus bust how the bottom part is filled accurately with no major deviations due to the different object positions. The completeness still does not reach 100% which is due to the NBS planning which aborts based on a coverage estimation utilizing the current surface model which sometimes is noisy. However, these are just small holes which can easily be filled by a postprocessing technique. For the bunny and chevron, 100% is reached for some runs whereas for the Zeus bust a small part in the chin area below the beard could never be filled due to sensor restrictions as this area is very narrow. The CRMS shows that, due to object repositioning and SMCR, the model error does not increase. The CRMS is even slightly lower when the object is repositioned. One reason for this is probably due to the fact that along borders in the mesh larger errors occur due to incorrect matching (see Figure 22(a)). Further, the objects do not have many details on the bottom and thus the error is lower which influences the average error positively.

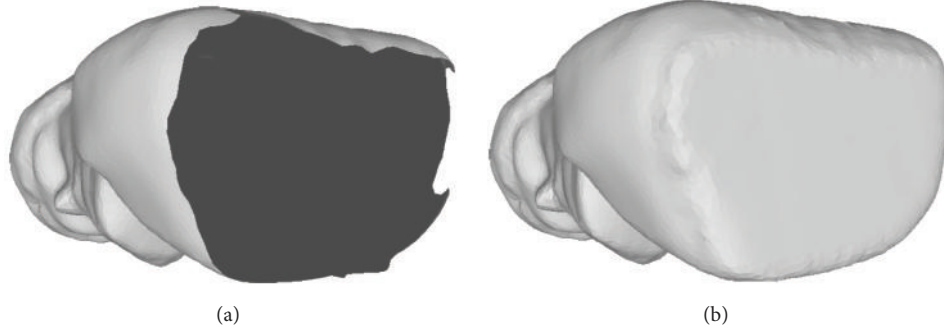


FIGURE 22: 3D model of Zeus bust from bottom view without (a) and with repositioning the object and performing SMCR (b).

TABLE 7: Comparison of modeling results without and with repositioning using SMCR (average of 10 runs).

Object	Repositioning	Completeness	CRMS
Zeus	No	88.0%	1.56 mm
	Yes	97.3%	1.46 mm
Bunny	No	91.7%	1.56 mm
	Yes	99.7%	1.37 mm
Chevron	No	97.7%	1.44 mm
	Yes	99.9%	1.34 mm



FIGURE 23: Kidnapped robot problem: a robot is randomly placed in a predefined area (green) in a known map. After self-localization the robot plans a path (yellow) to its goal.

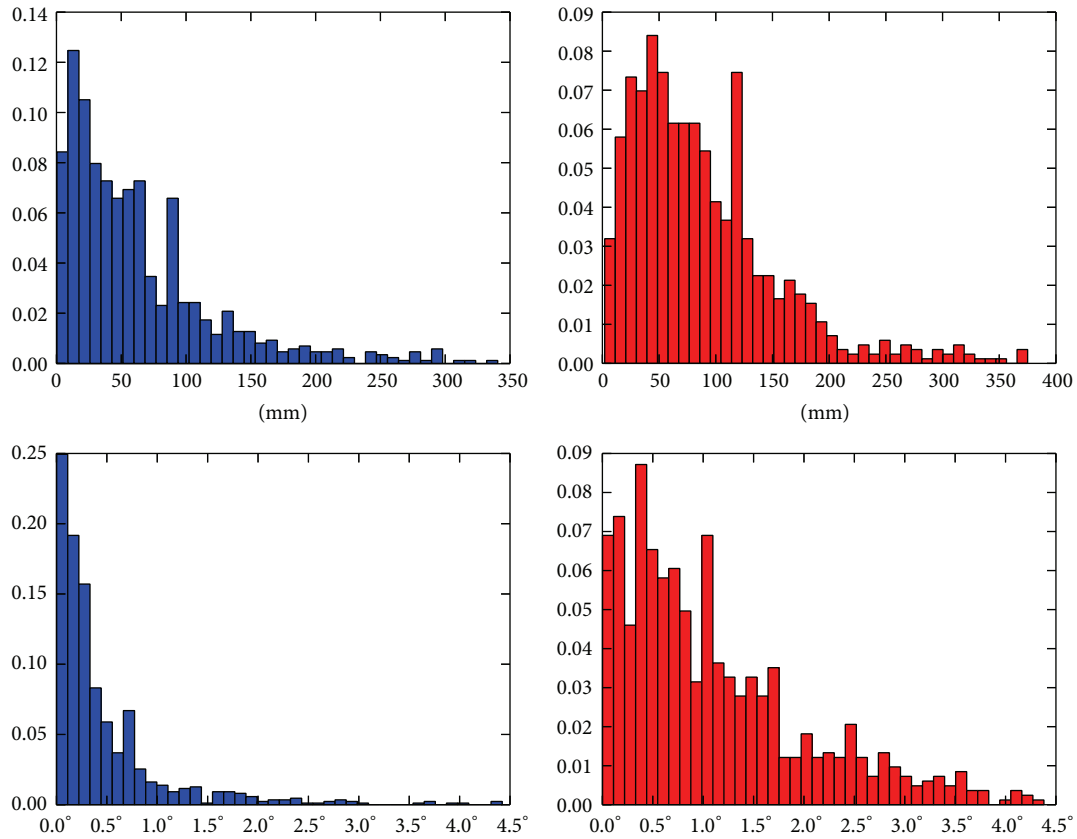
8.4. SMCRO in Mobile Robot Localization. In order to compare SMCRO with a standard particle filter based on 3D depth images (see [15]), we set up a simulated kidnapped robot scenario, where a mobile robot is randomly placed in a predefined area at an unknown pose $(x, y, \theta) \in \mathbb{R}^2 \times [-\pi, \pi)$ in a given global 3D map as shown in Figure 23. The particle filter weights the particles with a likelihood representing independently and identically normally distributed errors for

the depth values. Note that the map is 3D and the depth image simulation (for expected depth images in the weighting step) is also done in 3D, whereas the pose estimation is only in 2D. Our robot is moving omnidirectional on four wheels and is equipped with eight ToF cameras, each having a field of view of about $35^\circ \times 45^\circ$ and a resolution of 48×64 pixels. This setup is chosen such to represent the KUKA OmniRob, equipped with eight O3D100 ToF cameras of ifm.

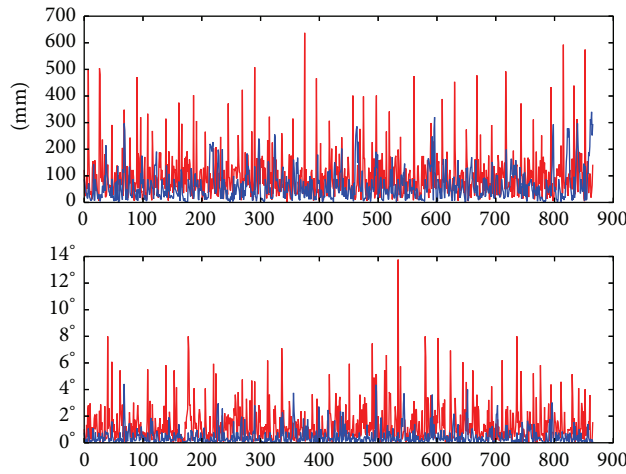
The robot’s task is to get “home” immediately, which can be divided into three subtasks: at first the robot actively localizes itself using the ToF cameras and a particle filter to get its current pose in the global map. After a successful localization the robot plans a piecewise linear path to reach the goal, containing about 20 waypoints. Finally the robot moves to the goal along the calculated path. For the odometry as well as the ToF data artificial noise is added. At every waypoint the robot corrects its odometrical pose using the ToF cameras to reduce the dead reckoning error. For correcting the pose, the standard particle filter is used. SMCRO is running in parallel, enabling a comparison of pose estimates, starting with the initial localization.

In order to achieve a better comparison the whole task has been repeated 40 times, resulting in a total number of 866 pose estimation steps. Figure 24(b) shows the errors of all these steps, starting with the first run. Clearly a periodic behavior is seen, which resembles the strong dependency of the pose estimation quality from the real pose in the environment. Note that the histograms (see Figure 24(a)) have been cut off on the right for better visibility. Both histograms and plots show that the accuracy of SMCRO outperforms that of the standard particle filter. The medians of translational and rotational errors are 76.4 mm and 0.94° for the standard particle filter and 47.6 mm and 0.26° for SMCRO. The maximum of translational and rotational errors are 637.5 mm and 13.8° for the standard particle filter and 340.5 mm and 4.4° for SMCRO.

Note that the mean computation times are 1.59 s for the standard particle filter and 2.1 s for SMCRO. In contrast to SMCRO, the standard particle filter has already been optimized for the mobile robot application. The number of beams from the depth images used for the updates are dependent on the number of particles in order to assure acceptable computation times.



(a) Histograms of errors of all iterations



(b) Errors of all update steps of all iterations, sorted by runs

FIGURE 24: Comparison between SMCRO (blue) and a standard particle (red) based on depth images. In the plots the errors are sorted by runs, resulting in periodic effects.

8.5. Discussion. The comparison with offline global methods yields no definite best method. But they clearly show that the method is competitive in accuracy and robustness with available state-of-the-art methods in many cases. The advantage of no extra computation time is clear, as even for these simple objects the offline methods need up to 2.5 minutes for getting similar results, whereas our method does not need any extra computation time. This effect will dramatically increase

with larger objects, which could not be investigated with the hardware setup in this paper, due to kinematic constraints.

The comparison between uniform and normal/Bingham sampling show that normal/Bingham sampling does neither improve accuracy nor robustness. The investigation of the different scoring variants (SPFR, SMCR, SMCRO, and SMCRC) shows that SMCR has a clearly better convergence behavior than SPFR. The optimization in SMCRO yields

faster convergence, higher accuracy, and higher success rate. In many cases, it is advantageous and poses no problem to optimize in every weighting step. However, this has to be done carefully, as convergence to false transformations can occur. In the data sets of this paper, no delay in updating occurred when updating in every step. However, in bigger data sets, it could lead to problems concerning computation time. The given convergence criterion proved to work efficiently when combined with optimization, with only slightly lower accuracy and robustness.

For autonomous modeling with SMCR, the results show that almost complete 3D models including object parts which are not visible in the initial pose can be created. Further, the average model error when comparing to ground truth is not increased by the object repositioning and SMCR which shows that the pose estimation is performed accurately for all runs.

First experiments in mobile robot localization show that SMCR is able to achieve a significantly higher pose accuracy than a tuned standard particle filter with only slightly higher computation time which can easily be optimized.

9. Conclusion and Future Work

In this work, Monte Carlo registration methods have been presented and advanced. The offline particle filter variant searching in the space of rotations outperformed the state-of-the-art algorithms, especially when prior knowledge is available. The proposed curvature features proved to be robust under sensor noise. For streaming application, the scoring of rotations is too time consuming. Thus, the space of rigid body transformations is searched in this case. Various real data experiments showed the competitiveness of the streaming variant. Thereby, convergence behavior and influence of prior knowledge have been investigated. The streaming variant has been enhanced with pose optimization and convergence criterion. The applicability in autonomous 3D modeling has been proven by various experiments with an industrial robot and a laser striper. The integration of the streaming registration into autonomous object modeling worked robustly and allowed for obtaining complete high quality 3D surface models of initially unknown objects. Finally, experiments in mobile robot localization showed that the straightforward application without any tuning yielded a higher accuracy than a standard particle filter (with comparable computation time).

Future work will focus on autonomous feedback of failure, in order to enable rescanning and detailed investigation of other convergence criteria. Furthermore, we want to apply the method during modeling of object scenes as presented in [56] where the template models contain less data as objects are occluded by others. Moreover, we want to apply the method to real mobile robots for localization and for modeling of larger indoor areas of buildings with big data sets. If data becomes too big to keep all feature points in memory, probably the most demanding challenge will be the combination with and the development of data structures that enable reloading and unloading afforded feature points for the weighting.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work has partly been supported by the European Commission under Grant no. H2020-ICT-645403-ROBDREAM and the Bavarian Research Foundation under Grant no. AZ-1104-15.

References

- [1] C. Rink, Z.-C. Marton, D. Seth, T. Bodenmüller, and M. Suppa, "Feature based particle filter registration of 3D surface models and its application in robotics," in *Proceedings of the 26th IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '13)*, pp. 3187–3194, Tokyo, Japan, November 2013.
- [2] C. Rink, S. Kriegel, J. Hasse, and Z. Marton, "Onthey particle filter registration for laser data," in *Proceedings of the IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR '16)*, Cluj-Napoca, Romania, 2016.
- [3] C. Rink and S. Kriegel, "Streaming Monte Carlo pose estimation for autonomous object modeling," in *Proceedings of the 13th Conference on Computer and Robot Vision (CRV '16)*, Victoria, Canada, 2016.
- [4] S. Kriegel, C. Rink, T. Bodenmüller, and M. Suppa, "Efficient next-best-scan planning for autonomous 3D surface reconstruction of unknown objects," *Journal of Real-Time Image Processing*, vol. 10, no. 4, pp. 611–631, 2015.
- [5] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *The IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.
- [6] J. Sturm, W. Burgard, and D. Cremers, "Evaluating egomotion and structure-from-motion approaches using the TUM RGB-D benchmark," in *Proceedings of the Workshop on Color-Depth Camera Fusion in Robotics (IROS '12)*, October 2012.
- [7] T. Stoyanov, M. Magnusson, H. Andreasson, and A. J. Lilienthal, "Fast and accurate scan registration through minimization of the distance between compact 3D NDT representations," *International Journal of Robotics Research*, vol. 31, no. 12, pp. 1377–1393, 2012.
- [8] C. Kerl, J. Sturm, and D. Cremers, "Robust odometry estimation for RGB-D cameras," in *Proceedings of the 2013 IEEE International Conference on Robotics and Automation (ICRA '13)*, pp. 3748–3754, Karlsruhe, Germany, May 2013.
- [9] T. Whelan, H. Johannsson, M. Kaess, J. J. Leonard, and J. McDonald, "Robust real-time visual odometry for dense RGB-D mapping," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '13)*, pp. 5724–5731, Karlsruhe, Germany, May 2013.
- [10] C. Choi and H. I. Christensen, "RGB-D object tracking: a particle filter approach on GPU," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1084–1091, Tokyo, Japan, November 2014.
- [11] K. H. Strobl, *A flexible approach to close-range 3-d modeling [M.S. dissertation]*, Technische Universität München, München, Germany, 2014.
- [12] E. Mair, K. H. Strobl, T. Bodenmüller, M. Suppa, and D. Burschka, "Real-time image-based localization for hand-held 3D-modeling," *Künstliche Intelligenz*, vol. 24, no. 3, pp. 207–214, 2010.

- [13] M. Isard and A. Blake, "Contour tracking by stochastic propagation of conditional density," in *Computer Vision—ECCV '96: 4th European Conference on Computer Vision Cambridge, UK, April 15–18, 1996 Proceedings, Volume I*, vol. 1064 of *Lecture Notes in Computer Science*, pp. 343–356, Springer, Berlin, Germany, 1996.
- [14] W. Sepp, S. Fuchs, and G. Hirzinger, "Hierarchical featureless tracking for position-based 6-DoF visual servoing," in *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '06)*, pp. 4310–4315, IEEE, Beijing, China, October 2006.
- [15] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, MIT Press, Cambridge, Mass, USA, 2005.
- [16] V. Ferrari, F. Jurie, and C. Schmid, "From images to shape models for object detection," *International Journal of Computer Vision*, vol. 87, no. 3, pp. 284–303, 2010.
- [17] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [18] C.-S. Chen and Y.-P. Hung, "RANSAC-based DARCES: a new approach to fast automatic registration of partially overlapping range images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 11, pp. 1229–1234, 1999.
- [19] S. Winkelbach, "Efficient methods for solving 3D-Puzzle-Problems," *it-Information Technology*, vol. 50, no. 3/2009, pp. 199–201, 2008.
- [20] B. Drost, M. Ulrich, N. Navab, and S. Ilic, "Model globally, match locally: efficient and robust 3D object recognition," in *Proceedings of the 23rd IEEE Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 998–1005, San Francisco, Calif, USA, June 2010.
- [21] U. Hillenbrand, "Consistent parameter clustering: definition and analysis," *Pattern Recognition Letters*, vol. 28, no. 9, pp. 1112–1122, 2007.
- [22] R. B. Rusu, N. Blodow, and M. Beetz, "Fast Point Feature Histograms (FPFH) for 3D registration," in *Proceedings of the 2009 IEEE International Conference on Robotics and Automation (ICRA '09)*, pp. 3212–3217, Kobe, Japan, May 2009.
- [23] A. Aldoma, Z.-C. Marton, F. Tombari et al., "Tutorial: point cloud library: three-dimensional object recognition and 6 dof pose estimation," *IEEE Robotics & Automation Magazine*, vol. 19, no. 3, pp. 80–91, 2012.
- [24] R. B. Rusu, N. Blodow, Z. Marton, and M. Beetz, "Aligning point cloud views using persistent feature histograms," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '08)*, Acropolis Convention Center, Nice, France, September 2008.
- [25] N. Gelfand, N. J. Mitra, L. J. Guibas, and H. Pottmann, "Robust global registration," in *Proceedings of the 3rd Eurographics Symposium on Geometry Processing (SGP '05)*, M. Desbrun and H. Pottmann, Eds., pp. 197–206, Eurographics Association, 2005.
- [26] P. Li, P. Cheng, M. A. Sutton, and S. R. McNeill, "Three-dimensional point cloud registration by matching surface features with relaxation labeling method," *Experimental Mechanics*, vol. 45, no. 1, pp. 71–82, 2005.
- [27] G. Barequet and M. Sharir, "Partial surface and volume matching in three dimensions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 9, pp. 929–948, 1997.
- [28] G. Barequet and M. Sharir, "Partial surface matching by using directed footprints," *Computational Geometry: Theory and Applications*, vol. 12, no. 1-2, pp. 45–62, 1999.
- [29] F. Tombari and L. Di Stefano, "Hough voting for 3D object recognition under occlusion and clutter," *IPSN Transactions on Computer Vision and Applications*, vol. 4, pp. 20–29, 2012.
- [30] J. Glover, R. Rusu, and G. Bradski, "Monte Carlo pose estimation with quaternion kernels and the Bingham distribution," in *Proceedings of the Robotics: Science and Systems Conference*, Los Angeles, Calif, USA, June 2011.
- [31] J. Glover and S. Popovic, "Bingham procrustean alignment for object detection in clutter," in *Proceedings of the 26th IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '13)*, pp. 2158–2165, Tokyo, Japan, November 2013.
- [32] T. Bodenmüller, *Streaming surface reconstruction from real time 3D measurements [M.S. thesis]*, Technische Universität München, Munich, Germany, 2009.
- [33] W. R. Scott, G. Roth, and J.-F. Rivest, "View planning for automated three-dimensional object reconstruction and inspection," *ACM Computing Surveys*, vol. 35, no. 1, pp. 64–96, 2003.
- [34] S. Chen, Y. Li, and N. M. Kwok, "Active vision in robotic systems: a survey of recent developments," *International Journal of Robotics Research*, vol. 30, no. 11, pp. 1343–1377, 2011.
- [35] M. Karaszewski, R. Sitnik, and E. Bunsch, "On-line, collision-free positioning of a scanner during fully automated three-dimensional measurement of cultural heritage objects," *Robotics and Autonomous Systems*, vol. 60, no. 9, pp. 1205–1219, 2012.
- [36] S. Khalfaoui, R. Seulin, Y. Fougerolle, and D. Fofi, "An efficient method for fully automatic 3D digitization of unknown objects," *Computers in Industry*, vol. 64, no. 9, pp. 1152–1160, 2013.
- [37] L. Torabi and K. Gupta, "An autonomous six-DOF eye-in-hand system for in situ 3D object modeling," *International Journal of Robotics Research*, vol. 31, no. 1, pp. 82–100, 2012.
- [38] J. I. Vasquez-Gomez, L. E. Sucar, and R. Murrieta-Cid, "View/state planning for three-dimensional object reconstruction under uncertainty," *Autonomous Robots*, 2015.
- [39] U. Thomas, S. Kriegel, and M. Suppa, "Fusing color and geometry information for understanding cluttered scenes," in *Proceedings of the International Conference on Intelligent Robots and Systems IROS: Robots in Clutter Workshop (IROS '14)*, Chicago, Ill, USA, September 2014.
- [40] K. Bae and D. D. Lichti, "Automated registration of unorganized point clouds from terrestrial laser scanners," in *International Archives of Photogrammetry and Remote Sensing*, vol. 35 of *Proceedings of ISPRS Working Group V/2*, pp. 222–227, 2004.
- [41] M. Pauly, M. Gross, and L. P. Kobbelt, "Efficient simplification of point-sampled surfaces," in *Proceedings of the IEEE Visualization (VIS '02)*, pp. 163–170, Boston, Mass, USA, November 2002.
- [42] M. Pauly, R. Keiser, and M. H. Gross, "Multi-scale feature extraction on point-sampled surfaces," *Computer Graphics Forum*, vol. 22, no. 3, pp. 281–289, 2003.
- [43] S. Gumhold, X. Wang, and R. Macleod, "Feature extraction from point clouds," in *Proceedings of the 10th International Meshing Roundtable*, pp. 293–305, October 2001.
- [44] E. W. Weisstein, *CRC Concise Encyclopedia of Mathematics*, Chapman & Hall/CRC, 2nd edition, 2002.
- [45] F. Zacharias, C. Borst, and G. Hirzinger, "Capturing robot workspace structure: representing robot capabilities," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '07)*, pp. 3229–3236, San Diego, Calif, USA, November 2007.

- [46] J. C. Mitchell, "Sampling rotation groups by successive orthogonal images," *SIAM Journal on Scientific Computing*, vol. 30, no. 1, pp. 525–547, 2008.
- [47] A. Yershova, S. Jain, S. M. Lavalle, and J. C. Mitchell, "Generating uniform incremental grids on SO_3 using the hopf fibration," *International Journal of Robotics Research*, vol. 29, no. 7, pp. 801–812, 2010.
- [48] J. Arvo, "Fast random rotation matrices," in *Graphics Gems III*, D. Kirk, Ed., pp. 117–120, Academic Press Professional, San Diego, Calif, USA, 1992.
- [49] K. Shoemake, "Uniform random rotations," in *Graphics Gems III*, D. Kirk, Ed., pp. 124–132, Academic Press Professional, San Diego, Calif, USA, 1992.
- [50] W. Feiten, P. Atwal, R. Eidenberger, and T. Grundmann, "6D pose uncertainty in robotic perception," in *Advances in Robotics Research*, pp. 89–98, Springer, Berlin, Germany, 2009.
- [51] G. H. Givens and J. A. Hoeting, *Computational Statistics*, Wiley, Hoboken, NJ, USA, 2005.
- [52] J. L. Hintze and R. D. Nelson, "Violin plots: a box plot-density trace synergism," *The American Statistician*, vol. 52, no. 2, pp. 181–184, 1998.
- [53] M. Suppa, S. Kielhöfer, J. Langwald, F. Hacker, K. H. Strobl, and G. Hirzinger, "The 3D-modeller: a multi-purpose vision platform," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '07)*, pp. 781–787, Rome, Italy, April 2007.
- [54] P. Kremer, T. Wimb, J. Artigas et al., "Multimodal telepresent control of DLR's rollin' justin," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '09)*, pp. 1601–1602, Kobe, Japan, May 2009.
- [55] T. Bodenmüller, W. Sepp, M. Suppa, and G. Hirzinger, "Tackling multi-sensory 3D data acquisition and fusion," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '07)*, pp. 2180–2185, San Diego, Calif, USA, November 2007.
- [56] S. Kriegel, *Autonomous 3D modeling of unknown objects for active scene exploration [Ph.D. thesis]*, Technische Universität München (TUM), 2015.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

