

Streaming Monte Carlo Pose Estimation for Autonomous Object Modeling

Christian Rink and Simon Kriegel

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

Email: christian.rink@dlr.de, simon.kriegel@dlr.de

Abstract—This work contributes the optimization of a streaming pose estimation particle filter and its integration into an autonomous object modeling approach. The particle filter is advanced by an additional pose optimization in the particle weighting step. By integrating the method into the autonomous object modeling approach, the repositioning of objects is enabled, which is often necessary in order to acquire complete models. Experiments show that the usage of iterative closest point is too restrictive for general transformations. The used Monte Carlo method enables a robust pose estimation without loss of time and with high precision. Further, it is shown that the overall modeling results are improved clearly.

Keywords—Pose estimation; Laser scanning; 3D modeling; Active sensing;

I. INTRODUCTION

In robotics, 3D models are usually required in order to recognize, locate, and manipulate real objects. Nowadays, if no CAD data is given, 3D models of unknown objects are typically generated manually. If robots are supposed to handle unknown objects themselves in future, they need to be able to acquire an object model fully autonomously. The autonomous modeling approach presented in [1] is not able to generate a 3D model of all objects parts. For instance, if the 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. Thus, pose estimation is needed to realign the data.

The development of pose estimation techniques has been fortified by the evolution of low-cost 3D sensors. Thus, many recent techniques are designed for Kinect-like sensors. However, laser scanners yield higher robustness against shiny or reflecting surfaces and significantly higher data accuracy and density. Most pose estimation techniques can be applied to all kinds of 3D point data. However, most methods work locally or are not able to work streamingly with laser strippers. Recently, a streaming Monte Carlo pose estimation method has been developed to fill this gap [2]. It performs as good as other state-of-the-art algorithms and estimates poses during data acquisition, if a priori knowledge is available. The situation in autonomous 3D modeling, where the object must be placed within a predefined region with a more or less arbitrary orientation, fits perfectly for this method. To the best of our knowledge, this is the only feature based global streaming pose estimation method in the literature, that is based on pure geometrical information and working with laser strippers.

In this paper, we thus propose the usage of this method in combination with the autonomous 3D modeling approach

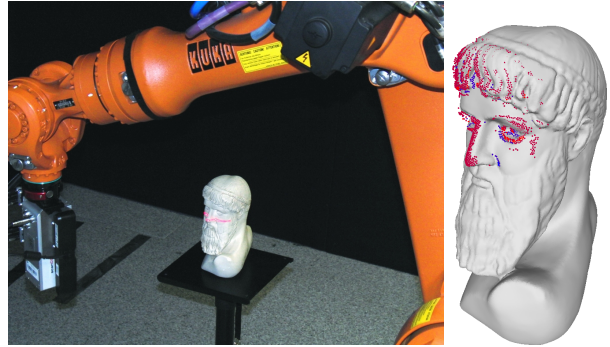


Fig. 1. Streaming Monte Carlo Filter Registration for Autonomous Object Modeling: *Left*: a robot with attached laser stripper performs multiple scan for autonomously modeling an initially unknown object. *Right*: After the object is repositioned, features are incrementally calculated to estimate the object pose using the current model and continue the autonomous modeling.

of Kriegel et al. [1] (see Fig. 1) as follows: initially a 3D model of an unknown object is generated. Then, the object is repositioned and the current model is directly used to estimate the new pose of the object. Finally, the autonomous modeling is continued until all parts have been covered.

Our contributions include the seamless integration of the pose estimation into autonomous 3D-modeling and a pose optimization in its weighting step. Additionally, we investigate sampling particles according to Gaussian/Bingham distributions and analyze the influence of a priori knowledge with respect to reliability and accuracy. Numerous experiments with an industrial robot and laser stripper are carried out to show the enhanced method's performance, compared to the previous approach. Finally, we investigate the impact to the modeling results: We show the improvements concerning completeness of models and prove that the overall accuracy of the obtained models is sustained. In contrast to [2] where data was registered to a complete ground truth reference model, here real scan data was used for the reference models.

II. RELATED WORK

In this section, we first summarize recent work on pose estimation and then briefly on autonomous object modeling.

A. Pose Estimation

For details of strongly related mobile robotics topics such as localization and SLAM, we refer to the comprehensive overview given by Sturm et al. [3]. However, particle filters based purely on depth images as used for mobile robot

localization are not suitable for the stated problem, since depth measurement simulation is too time consuming (even on a GPU) in comparison to our method. Moreover, methods in the realm of mobile robotics typically rely on local registration or focus on 2D (3 DOF) self localization.

Concerning object pose estimation, Fischler et al. [4] introduced the well-known random sampling consensus (RANSAC), and successively Chen et al. [5] demonstrated its application to registration. Variants typically calculate rigid motions from subsets of points or point-normal pairs that are sampled in the data sets. Winkelbach [6] uses point-normal pairs and concentrates on an efficient sampling strategy. Drost et al. [7] use a voting scheme similar to the Generalized Hough Transform. Rusu et al. [8] use the Fast Point Feature Histogram, a higher dimensional feature, to define correspondences. Then, a RANSAC method is utilized for selecting those correspondences with maximal 3D overlap. To evaluate multiple high-dimensional features for object recognition, Aldoma et al. [9] use a correspondence grouping method based on geometric consistency. By using a “center-star” variant of Rusu et al. [10], similarly-spaced point groups are matched. However, in contrast to [10], the points are evenly sampled instead of searching for salient points.

Barequet et al. [11] exploit the unique decomposability of rigid motions into rotations and translations. The discretized space of rotations is searched iteratively by clustering the corresponding translations and finding the most definite cluster, defining the best rotation. Tombari et al. [12] have proposed a similar approach that uses features yielding a complete reference frame in contrast to a sole surface normal. A correspondence pair defines a complete rigid motion, not only a set of rotations as in [13]. Nevertheless, up to a constant translation the voting table is the same. The correspondence pairs contributing to the best translation are used for a least squares estimation. In contrast, Barequet et al. use the found transformations directly (ICP is applied to the complete data set in a postprocessing step). Tombari et. al prove their method to be more robust and reliable than other standard methods based on pose space clustering or geometric consistency.

Rink et al. [14] reformulated Barequet’s approach as a particle filter and showed that in applications with very noisy data, relying on accurate surface normals or reproducible reference frames (as in [12]) can fail. Thus, scalar feature descriptors were used. Furthermore, a comparison to similar strategies [11], [8], [12], [9] was given and the advantages of explicit integration of prior knowledge about the searched transformation were 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 [15].

B. Autonomous Object Modeling

Autonomous object modeling usually makes use of a robot-sensor system and by iterative Next-Best-View (NBV) planning acquires a 3D model of an unknown object. The area of NBV planning has been widely explored [16], [17]. However, little research on real autonomous object modeling systems has been carried out. Khalifaoui et al. [18] utilizes an industrial robot, a turntable, and a very large and expensive fringe projection system to plan NBVs using barely visible surfaces as viewpoint candidates. However, several holes in the models remain. Torabi et al. [19] use a smaller robot with 2D range sensor but the focus lays more in the exploration and not in the object modeling. In [20], autonomous 3D reconstruction of unknown objects is performed with a mobile manipulator and Kinect sensor. The approach samples NBVs in configuration space and evaluates these in Cartesian space to avoid viewpoints that would lead to collisions. Kriegel et al. [1] use an industrial robot and consider the surface quality during Next-Best-Scan (NBS) planning for high-quality model acquisition. The planning is called NBS instead of NBV as the utilized sensor, a laser striper, needs to be moved in order to acquire 2D depth images. All of these approaches are not able to obtain the bottom part of the objects. In [21], the last approach [1] has been applied to also scan initially occluded parts, e.g. the bottom of an object for creating a pose estimation data set. However, here the object had to be manually repositioned about a defined axis quite perfectly (deviations in the manual repositioning were not possible). In this work, we propose to use the approach presented in [1] which focuses on the modeling and plans NBSs for a laser striper.

III. SYSTEM OVERVIEW

This section gives an overview (see Fig. 2) of how the autonomous modeling method as presented in [1] is combined with Streaming Monte Carlo Registration (SMCR). Initially,

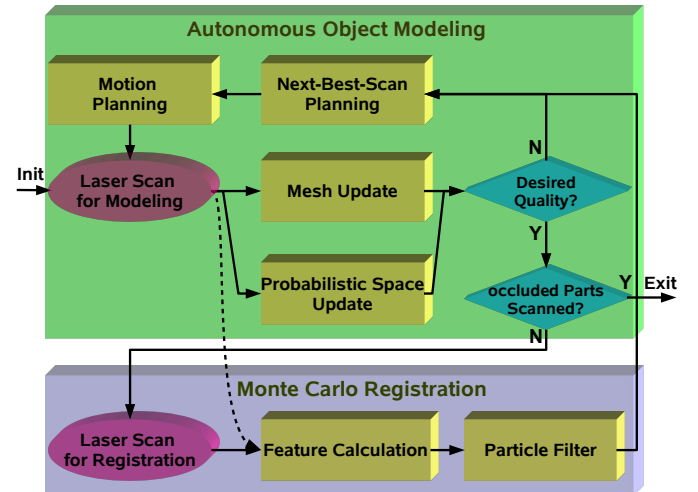


Fig. 2. Overview of the integration of autonomous object modeling with streaming Monte Carlo registration. Yellow boxes represent modules, purple ovals robot-sensor system actions, and blue diamonds decisions.

an arbitrary laser scan of the unknown object is performed utilizing a robot-sensor system. Here, robot-sensor system refers to an industrial robot with a laser stripier attached to the robot’s flange (see Fig. 1). When data acquisition is started, a depth image stream containing the robot’s pose information is handled to three modules: *Mesh Update*, *Probabilistic Space Update*, and *Feature Calculation*. The features are also calculated in real-time here but will only be used later for registration after the object has been repositioned. After the updates, we check if the triangle mesh (surface model) has reached the desired quality meaning if it is complete apart from not scanable parts. Then *Next-Best-Scan Planning*, collision-free *Motion Planning*, further laser scans, and real-time updates are iteratively performed until the quality is reached. The mesh is utilized for planning possible scan path candidates and selecting a NBS with the goal to reach the desired surface model quality. The probabilistic space represents a volumetric model which considers sensor uncertainties and gives a probability of occupancy for each voxel. It is used for exploring the unknown environment by *Next-Best-Scan planning*, and avoiding collisions during *Motion Planning*.

Once the desired quality has been reached, the object is repositioned meaning it is rotated onto one of its sides in order to model previously occluded object parts. Therefore, a laser scan is performed along the region of interest for which again the *Feature Calculation* (see Sect. V) is carried out on-the-fly. Here, features are also classified according to the class borders of the template (feature points before the object was repositioned). Synchronously, the *Particle Filter* component iteratively performs a neighborhood sampling and weighting of the particles with incoming feature points. The estimated pose is instantly available after the *Laser Scan for Registration* has finished. Then, the Monte Carlo registration is fine adjusted by applying the ICP which results in a precise transformation between the original object position and the object position after it has been rotated onto one of its sides. After the registration, the autonomous modeling is continued until a complete model is generated. Thereby, the resulting transformation from the registration is applied to the robot pose of all further generated laser scans in order to be able to model the object within the same coordinate system. For more details on the autonomous modeling we refer to [1] and concerning the particle filter registration to [2] and a summary in the following section.

IV. MONTE CARLO REGISTRATION

The particles are sampled in the space of rigid body transformations, denoted \mathcal{T} . The unknown transformation between two models P and Q of the same object at time step i is denoted $T_i \in \mathcal{T}$. Each particle consists of a transformation $T \in \mathcal{T}$ and a weight w . The state transition A_i between two time steps can be considered the identity in the case of registration, and the observed model is assumed to be not moving ($A_i = \mathbf{id}$). The error ε_{A_i} is assumed to be distributed uniformly in a neighborhood of the identity. Each particle (T, w) in each time step i is weighted by the conditional

probability density function (**pdf**) $f(Q_i|T, P)$ of observed data Q_i conditional on the state T and the template model P . In short, the particle filter registration can be described by:

- 1) sample pose particles initially
- 2) weight particles by $f(Q_i|T, P)$
- 3) resample particles according to their weights
- 4) optionally: Adapt some parameters
- 5) sample particles in neighborhood of existing particles, return to step 2 if not converged

In this work, we adapt the number of particles, the sampling radii and the radius for the score function from Sect. VI-B. 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.

V. FEATURES

In this section, we clarify how features originally introduced in [14] are calculated streamingly.

A. Robust Scalar Features

Every feature point $p = (c_p, n_p, v_p) \in \mathbb{R}^3 \times \mathcal{S}^2 \times \mathbb{R}$ consists of a coordinate c_p , a surface normal n_p , (\mathcal{S}^2 being the unit sphere) and a feature value v_p . We shortly present slight variants of scalar features proposed in [14], that proved to be robust under noisy depth measurements.

1) *Normal Cosines*: Let in the following p be a point with surface normal n_p and a neighborhood $N(p)$ and define

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

for a neighbor $q \in N(p)$. Then, the mean, maximum and minimum of $\{c(p, q) | q \in N(p)\}$ are called the mean normal cosine (MNC), maximum normal cosine (MaNC) and minimum normal cosine (MiNC) in p with neighborhood $N(p)$, respectively.

2) *Point Clouds and Eigenvalues*: If no stable normal estimation is given, features can be calculated with the help of the eigenvalues $\lambda_1 \leq \lambda_2 \leq \lambda_3$ of a point neighborhood covariance matrix: the scalar curvature feature defined by $\frac{\lambda_1}{\lambda_3}$ is denoted eigenvalue quotient of eigenvalues 1 and 3 (EVQ13). When no relevant curvature features can be exploited for registration, e.g. with flat objects like a metal sheet, the feature value $\frac{\lambda_2}{\lambda_3}$ can be used (EVQ23), indicating whether a point is near the border of the object [14].

B. Streaming Feature Calculation and Classification

The processing pipeline for streaming feature calculation consists of three stages: the *density limitation*, the *normal estimation* and the *feature generation* step. Depth points coming from a real-time data stream are incrementally inserted into the model if they pass a limitation test: each newly acquired point, that is closer than a distance r_r to any point already inserted to the model, is rejected. Thus, the entire Euclidean point density of the model is limited and the computational effort can be controlled. For each point that passes the *density limitation*, a surface normal is estimated using principal component analysis for all points within a spherical neighborhood with

radius r_n . Only points, for which the *normal estimation* is considered robust (see [15] for details), are transferred to the subsequent *feature generation* step.

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 straight forward: if a stable normal is ready, the corresponding feature point is calculated from the eigenvalues and inserted into the feature point stream.

2) *Normal Cosines*: The proposed angle features are calculated from the point surface normal and the neighborhood points in the model. Consequently, the original *mesh generation* stage is adapted to serve as a *feature generation* module as follows. If a stable normal is ready and the new point is inserted into the module the MNC, MaNC, or MiNC are 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.

C. Reduction

Besides defining correspondences, the features are used for data reduction [14]. In this work, five feature classes are defined on the template data set. The middle class which represents low curvature areas is removed. During the streaming feature calculation, only features are inserted into the data that are not in the middle class.

VI. SAMPLING POSE-PARTICLES

We follow the widely used approach to initially sample our pose particles uniformly in a predefined region, as detailed in [2]. Contrary to [2], the neighborhood sampling in the successive iterations will be performed with truncated Normal or truncated Bingham distribution for translations and rotations, respectively.

A. Sampling Rigid Body Transformations in Neighborhoods

Sampling translations uniformly or normally in neighborhoods is trivial. Sampling rotations uniformly is outlined in [14]. If uniformly sampled rotations are available, importance resampling can be used to allow rotations to be distributed according to a truncated Normal or Bingham. This works efficient as long as the Normal/Bingham distribution has a standard deviation fitting the truncation. In this work, the standard deviation is chosen to be half of the sampling radius.

B. Scoring Transformations

The weights of the particles are calculated according to a Normal distribution of the feature point locations, conditional on the pose. Let P, Q be the feature points of the template and the incoming data feature points, correspondingly. Further we assume the features to be classified (according to [14]), i.e. v_p is a discrete category for every $p \in P$ (and correspondingly for Q). Now consider a particle describing a transformation T . Let q be the feature point in the data corresponding to the feature point p of the template model. If the underlying

transformation T between data and template is known, it is reasonable to assume c_q to follow a Normal distribution with expectation $T(c_p)$ and some covariance matrix $\Sigma = \sigma^2 \cdot \mathbf{id}$. Then, if the errors are identically and independently distributed and we consider a set of feature points $\mathbf{p} = \{p_1, \dots, p_n\}$ and a set of correspondences $\mathbf{q} = \{q_1, \dots, q_n\}$, we end up at the conditional **pdf** of all feature point locations to be:

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

In practice, the corresponding p_i are approximated by the nearest feature point to q_i with the same feature class:

$$p_i = \arg \min_{p \in P, v_p = v_{q_i}} d(T(c_p), c_{q_i}). \quad (3)$$

If the feature class is erroneous for some q , no correct corresponding p will be found in the template. The best we can do is to assume that the feature location is distributed uniformly, conditional on a wrong corresponding p . For some distance threshold r_{\max} we define

$$d_i := \min \{d(T(c_{p_i}) - c_{q_i}), r_{\max}\} \quad (4)$$

and adapt Equation (2) to

$$f(\mathbf{q}|T, \mathbf{p}) \propto \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^n d_i^2\right) \quad (5)$$

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). Although this is an improper **pdf**, it can be used for calculating the importance weights in sampling importance resampling [22].

Therefore, each particle's transformation T is scored with incoming feature points Q as follows. Each point q_i of Q is classified according to the class borders of the template, and transformed to $T^{-1}(c_{q_i})$. Then, the template model is searched for the nearest feature point p_i of the same feature class and the distance d_i is saved. If no such point is found within the search radius r_{\max} , the distance is set to that radius ($d_i = r_{\max}$). When all distances d_i are calculated for the n feature points, we determine

$$w(T) = \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^n d_i^2\right). \quad (6)$$

Scoring Variants: There are two scoring variants: either only the newly inserted feature points are used for Q_i , denoted SPFR, or all previously acquired points are used ($Q_i := \bigcup_{j < i} Q_j$), denoted SMCR, which has a better convergence behavior [2].

Optimization: A found p_i with $d_i \leq r_{\max}$, according to Equation (3), can be used as correspondence for q_i . In each weighting step, we thus get a set of correspondences for each particle. This can be used for an ICP-iteration in order to correct the corresponding pose particle. If we use such an optimization step, we denote the method SMCRO



Fig. 3. The test objects used for the experiments: bunny, Zeus bust, and wooden chevron (from left to right).

in the remainder. Note that we use this optimization only in every tenth update step of the particle filter in order to avoid overfitting, especially for the first updates.

VII. EXPERIMENTS AND EVALUATION

In this section, experiments with real data are demonstrated. First, hardware, experimental setup, and parameters are described. Then, the influence of a priori knowledge for the pose estimation is investigated, and the performance of 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.

A. System Setup

Here, a 6 DOF industrial robot, the Kuka KR16-2, with mounted laser striper is utilized (see Fig. 1). For the KR16-2, the absolute positioning error is in millimeter range. The SMCR and autonomous 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. During laser scans, the robot pose and range data are synchronized.

The experiments were performed for three objects: a bunny, a Zeus bust, and a wooden chevron (see Fig. 3). These represent the application domains household, manufacturing and cultural heritage. The approximate height of the bunny and chevron is 18 cm, and of the bust 22 cm. The objects are placed onto a pedestal as can be seen in Fig. 1.

B. Parameters and Evaluation Criteria

As stated in Sect. IV, 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.

TABLE I
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_ρ	sr	m_t/m_R
0.48 dm ³	10°	69 %	3.8/4.6
0.48 dm ³	20°	71 %	3.9/4.0
0.48 dm ³	45°	68 %	4.0/4.2
0.48 dm ³	90°	74 %	3.2/4.0
0.48 dm ³	90° (z)	93 %	0.7/1.0
1.22 dm ³	10°	69 %	4.1/4.4
1.22 dm ³	20°	64 %	4.3/5.9
1.20 dm ³	45°	63 %	4.0/5.5
1.22 dm ³	90°	71 %	3.5/4.1
4.00 dm ³	45°	48 %	7.2/8.1

The evaluations are done with respect to the median of rotational and translational error, denoted m_t and m_R , respectively, and a success rate. A success is defined, if the final error in translation and rotation is below 8 mm and 8°. The ratio between successful runs and total runs is called success rate and denoted sr. The given bounds proved necessary for the ICP to converge [2].

C. Influence of a priori Knowledge

The a priori knowledge about the searched transformation is explained in terms of a cuboid for the translational part. In the following, the volume of that cuboid is denoted $V(t)$. The rotational part is expressed by a mean rotation and a maximum rotational difference from it. In some cases the rotation axis can be assumed to be fixed, for example if it is known, that the object has been turned about the z-axis. The maximal deviation from the mean rotations is denoted r_ρ .

In order to investigate the influence of the a priori knowledge, represented by $V(t)$ and r_ρ , one scan of the bunny has been registered to a known ground truth surface model, acquired with a commercial scanning system. The results are depicted in Tab. I. The initial sampling radius for the rotation seems to have little effect on the success rate and on m_t/m_R . Apparently, the volume of the cuboid for the initial translation has more influence. This confirms the suitability in autonomous 3D modeling, as it often yields good a priori knowledge about the position of the object, whereas the rotation cannot be estimated beforehand.

D. ICP vs. SMCR with partially overlapping submodels

In these experiments, we tried to register two partially overlapping scans of the Zeus bust to each other, which poses a higher challenge than registering a submodel to a complete ground truth model with few noise. For this purpose, we recorded 20 different pairs of overlapping scans (see Fig. 4 for an example). To one of the scans in each pair we applied 10 different random transformations with translational and rotational variations. The translation vectors had a maximal norm of 20 mm and the rotations a maximum rotation angle of 45°, 90°, 120° or 180° around the z-axis. The a priori knowledge in SMCR was set accordingly, denoted with Z45X, Z90X, Z120X, Z180X respectively in Tab. II. Z stands for the object Zeus and X presents the different sampling methods described below. After the SMCR, we applied

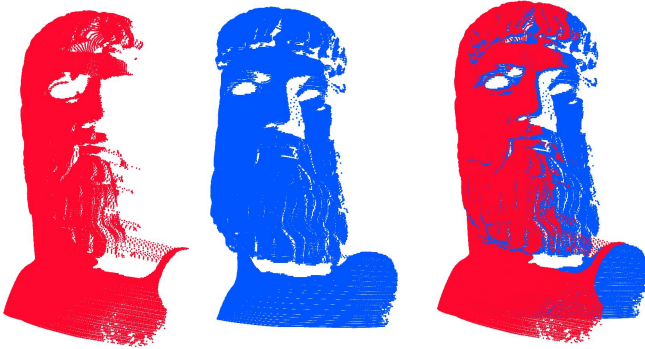


Fig. 4. Typical scans of Zeus bust (aligned on the right)

TABLE II

m_t , m_R AND \bar{t} FOR SMCR WITH (G)AUSSIAN/(U)NIFORM SAMPLING AND WITH (O)PTIMIZATION 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	\bar{t}	m_t/m_R	sr	\bar{t}	m_t/m_R	sr	\bar{t}
Z45G	4.8 / 8.3	0.47	1.6 / 2.6	0.69	1.8				
Z45U	5.1 / 6.4	0.5	1.8 / 2.9	0.66	1.8	2.0 / 2.7	0.64	6.7	
Z45O	3.5 / 6.3	0.51	1.3 / 3.1	0.71	1.8				
Z90G	5.8 / 7.0	0.48	1.8 / 3.1	0.64	3.1				
Z90U	5.3 / 6.8	0.49	1.8 / 2.9	0.65	2.9	9.2 / 31.2	0.32	6.7	
Z90O	3.7 / 7.1	0.49	1.3 / 2.9	0.71	2.9				
Z120G	5.9 / 9.3	0.41	1.8 / 3.1	0.65	2.0				
Z120U	5.4 / 8.0	0.45	1.8 / 2.9	0.64	2.0	12.4 / 50.9	0.27	5.1	
Z120O	3.9 / 9.7	0.48	1.5 / 3.1	0.65	2.1				
Z180G	7.3 / 11.9	0.37	2.5 / 3.4	0.60	2.1				
Z180U	6.7 / 11.9	0.40	2.6 / 3.4	0.6	2.1	15.7 / 87.8	0.13	3.9	
Z180O	5.9 / 12.2	0.42	2.2 / 3.4	0.61	2.1				
units	mm / deg	%	mm / deg	%	s	mm / deg	%	s	

an ICP with a small search radius (20 mm) and few iterations for fine fitting, denoted SMCR-ICP. We compared the results to a pure ICP with a bigger search radius (50 mm), simply denoted ICP. For each rotation-range we tested the original (U)niform neighborhood sampling, the proposed (G)aussian sampling and the Gaussian sampling with (O)ptimization step, denoted by a capital U, G or O 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. In the described tests, Gaussian sampling increases both accuracy and reliability. The optimization increases accuracy in translation and the success rate. The rotational accuracy is not increased. The effects appear in all rotation ranges. Concerning the performance of pure ICP, it becomes clear, that with increasing rotation range, results get too bad, both in accuracy and reliability. SMCR-ICP outperforms pure ICP clearly, both in reliability and accuracy.

Success rates of pure SMCR appear to be pretty small for two reasons: On the one hand, we did not tune the parameters for the data sets. On the other hand, the partially overlapping scans are not as easy to register as it seems at first glance. Very similar features are spread over the object and the most descriptive features are not easy to scan. As the scan paths are chosen arbitrarily, they appear randomly in the data sets. Preceding papers [14], [2] showed that other state-of-the-art methods performed even worse for this kind of data, even with a registration to a complete high precision ground truth

TABLE III
 m_t , m_R AND \bar{t} FOR SMCR WITH (G)AUSSIAN/(U)NIFORM SAMPLING AND WITH (O)PTIMIZATION STEP AND ICP FOR THE (Z)EUS BUST, THE (B)UNNY AND THE (C)HEVRON AND 45°, 90°, 120° AND 180° ROTATIONS.

data	SMCR			SMCR-ICP			ICP		
	m_t/m_R	sr	\bar{t}	m_t/m_R	sr	\bar{t}	m_t/m_R	sr	\bar{t}
Z45G	7.9 / 4.5	0.5	0.4 / 0.4	0.9	1.8				
Z45U	3.1 / 2.0	0.6	0.4 / 0.4	0.9	1.6	0.6 / 0.8	0.8	12.5	
Z45O	4.0 / 3.0	0.6	0.4 / 0.5	0.9	1.8				
Z90G	15.0 / 15.2	0.3	0.7 / 0.5	0.8	2.0				
Z90U	12.3 / 9.1	0.4	0.4 / 0.4	0.8	1.8	4.7 / 38.3	0.3	9.7	
Z90O	7.0 / 5.3	0.6	0.4 / 0.4	0.8	1.6				
Z120G	21.6 / 21.1	0.2	0.7 / 0.6	0.7	2.3				
Z120U	19.2 / 19.0	0.3	0.9 / 0.6	0.7	2.0	6.5 / 25.2	0.4	10.6	
Z120O	16.2 / 12.7	0.4	0.9 / 0.6	0.7	2.0				
Z180G	23.9 / 28.1	0.2	0.8 / 0.6	0.7	2.0				
Z180U	18.6 / 19.1	0.4	0.5 / 0.5	0.7	1.9	16.8 / 51.8	0.3	10.3	
Z180O	12.2 / 13.9	0.4	0.9 / 0.6	0.6	1.7				
B45G	7.8 / 8.0	0.4	0.7 / 0.2	1.0	0.9				
B45U	6.5 / 7.8	0.4	0.7 / 0.2	0.9	0.9	0.6 / 0.2	1.0	2.6	
B45O	4.7 / 5.8	0.6	0.7 / 0.2	0.9	0.9				
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	0.7 / 0.3	0.9	2.9	
B90O	4.9 / 6.2	0.6	0.7 / 0.2	0.9	0.9				
B120G	9.1 / 13.1	0.2	0.7 / 0.2	0.9	1.0				
B120U	7.2 / 8.5	0.4	0.7 / 0.3	0.9	1.0	0.7 / 0.4	0.7	3.0	
B120O	5.7 / 6.3	0.5	0.7 / 0.2	0.9	1.0				
B180G	8.9 / 10.4	0.2	0.7 / 0.2	0.9	1.0				
B180U	7.4 / 9.1	0.3	0.7 / 0.2	0.9	1.0	11.1 / 60.7	0.4	3.1	
B180O	7.3 / 7.6	0.4	0.7 / 0.2	0.9	0.9				
C45G	13.0 / 0.8	0.3	6.0 / 1.0	0.6	2.8				
C45U	12.3 / 0.7	0.4	6.1 / 1.0	0.6	2.5	4.9 / 0.7	0.9	11.3	
C45O	13.3 / 1.3	0.4	6.2 / 0.9	0.6	2.7				
C90G	18.7 / 0.8	0.3	11.1 / 1.3	0.5	2.6				
C90U	12.6 / 0.8	0.3	6.4 / 1.1	0.6	2.6	5.0 / 0.9	0.9	12.7	
C90O	15.8 / 1.3	0.3	9.5 / 1.2	0.5	2.5				
C120G	14.4 / 1.3	0.3	6.3 / 1.1	0.6	2.6				
C120U	15.2 / 1.1	0.3	6.6 / 1.2	0.6	2.4	4.8 / 0.8	0.8	11.1	
C120O	18.0 / 2.2	0.3	11.1 / 1.3	0.5	2.3				
C180G	19.9 / 2.5	0.2	11.9 / 1.5	0.5	2.6				
C180U	17.5 / 3.1	0.2	11.4 / 1.4	0.5	2.3	14.1 / 26.9	0.5	11.2	
C180O	19.0 / 2.3	0.2	16.7 / 2.0	0.4	2.3				
units	mm / deg	%	mm / deg	%	s	mm / deg	%	s	

surface model.

E. ICP vs. SMCR in Autonomous Modeling

In this section, we show experiments concerning the later use in autonomous 3D modeling. Therefore, we first autonomously acquired a more or less complete model of the objects (without bottom part). Then, we repositioned them on the scanning pedestal and acquired 10 different single scans manually. Each of these manual scans was transformed by ten different random transformations. In total, resulting in 100 different test scans for each object which were registered to the corresponding previously acquired complete model.

Each ground truth estimation was calculated by utilizing the global method from [14], followed by an ICP working on all acquired raw points of the ten subscans. Correctness was assured by visual inspection of a human operator.

1) *Goodness of Fit and Success Rates:* Tab. III shows the results of the experiments concerning rotational and translational error as well as the success rate. In contrast to the experiments in the previous section, the registration was performed based on the 3D model and additionally 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 yield a higher accuracy and success rate. SMCR itself performs good, but accuracy

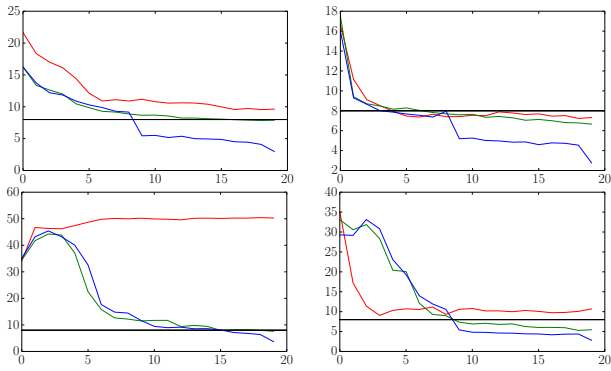


Fig. 5. Exemplary error convergence of SPFR (red), SMCR (green) and SMCRO (blue) for 100 runs on a bunny (top) and a Zeus (bottom) scan. Left: translational error in mm. Right: rotational error in degree. X-axis: step number. The black horizontal line represents the success threshold of 8 degrees or 8 mm.

is not comparable to the ICP (if both are successful). In contrast, the proposed Gaussian sampling does not lead to higher accuracy or success rates. Moreover, the results partly confirm the experiments on the influence of a priori knowledge (see Sect. VII-C) in some cases. 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 bunny, 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. 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 was placed on. Consequently, we excluded the Chevron from the modeling experiments in Sect. VII-F, as the scan data errors were too large to robustly acquire complete and accurate models.

2) *Convergence Behavior*: In this section, we show the convergence behavior of the proposed SMCRO, in comparison to SMCR and SPFR. Therefore, we repeated estimations for one scan path of the bunny and the Zeus bust 100 times and calculated the median translational and rotational errors in each update step, as depicted in Fig. 5. Clearly, SMCRO yields a faster convergence than SMCR and SPFR. Note that the optimization step is only performed in every tenth update, i.e. in update step 9 and 19. At these update steps, the error medians visibly drop down. However, the optimization needs to be carried out with caution as for too early or too many optimization steps the method may tend to converge to the wrong transformation, especially for objects with many symmetries.

TABLE IV
COMPARISON OF MODELING RESULTS WITHOUT AND WITH REPOSITIONING USING SMCR (AVERAGE OF 10 RUNS).

Object	Bunny		Zeus	
	No	Yes	No	Yes
Repositioning				
Completeness	91.7 %	99.7 %	88.0 %	97.3 %
CRMS	1.56 mm	1.37 mm	1.56 mm	1.46 mm

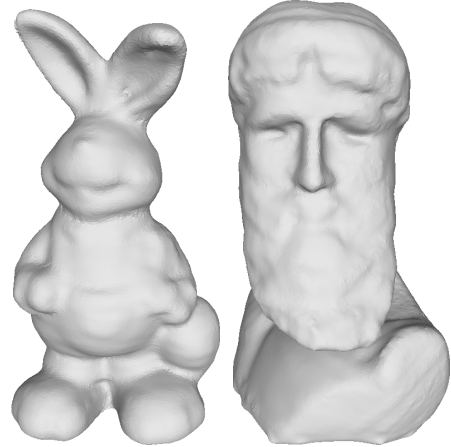


Fig. 6. 3D models of bunny and Zeus object acquired during the autonomous object modeling.

F. Autonomous Modeling with SMCR

Here, we compare the autonomous modeling results without repositioning the object as in [1] with the integration of the SMCR and repositioning of the object as presented in Fig. 2. Therefore, the complete object modeling is performed 10 times for the bunny and Zeus object.

After the desired quality for the visible object parts has been reached, the object is manually placed onto its side. 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 Tab. IV. 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 Fig. 6. 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. Fig. 7 shows exemplary 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, 100 % was 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

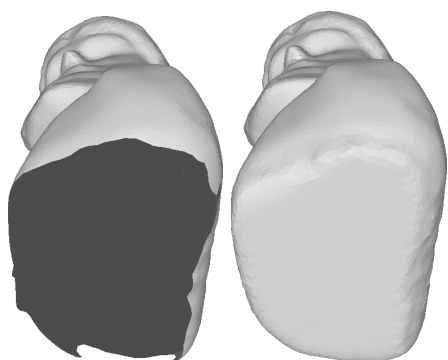


Fig. 7. 3D model of Zeus bust from bottom view without (left) and with repositioning the object and performing SMCR (right).

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 Fig. 7 left). Further, the objects do not have many details on the bottom and thus the error is lower which influences the average error positively.

The modeling results show that by using SMCR 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 was performed accurately for all runs.

VIII. CONCLUSIONS AND FUTURE WORK

In this paper, a streaming pose estimation method has been optimized and successfully integrated into an autonomous 3D modeling approach. The influence of a priori knowledge for pose estimation has been examined and the registration performance has been compared to the previous method and to ICP. We proved our concept of autonomous 3D modeling to work robustly and obtained complete high quality 3D surface models with real data experiments. The results show that the streaming pose estimation method can be successfully applied to fully model unknown objects autonomously.

Future work will focus on autonomous feedback of convergence or failure, in order to enable scan interruption or rescanning. If convergence is reported, the processing pipeline could also immediately pass incoming data to the streaming modeling modules, such that the scan does not need to be interrupted, but can directly be used for modeling. Further, we want to apply the method during modeling of scenes containing several objects as presented in [23] where objects are occluded by others resulting in template models with less data. Moreover, we want to apply the method in mobile robot localization and modeling of larger indoor areas of buildings.

ACKNOWLEDGMENT

This work has partly been supported by the European Commission under contract number H2020-ICT-645403-ROBDREAM.

REFERENCES

- [1] 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.
- [2] C. Rink, S. Kriegel, J. Hasse, and Z. Marton, "On-the-fly particle filter registration for laser data," in *IEEE AQTR*, Cluj-Napoca, Romania, May 2016, accepted for publication.
- [3] J. Sturm, W. Burgard, and D. Cremers, "Evaluating egomotion and structure-from-motion approaches using the TUM RGB-D benchmark," in *Proc. of the Workshop on Color-Depth Camera Fusion in Robotics at the IROS*, Oct. 2012.
- [4] 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, Jun. 1981.
- [5] C. Chen, Y. Hung, and J. Cheng, "RANSAC-based DARCES: A new approach to fast automatic registration of partially overlapping range images," *IEEE Transactions PAMI*, vol. 21, pp. 1229–1234, 1999.
- [6] S. Winkelbach, "Effiziente Methoden zum Lösen von 3D-Puzzle-Problemen," *it - Information Technology*, vol. 50, no. 3, pp. 199–201, 2008.
- [7] B. Drost, M. Ulrich, N. Navab, and S. Ilic, "Model globally, match locally: Efficient and robust 3D object recognition," in *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, San Francisco, CA, USA, Jun. 13–18, 2010, pp. 998–1005.
- [8] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *ICRA*, Kobe, Japan, May 12–17, 2009.
- [9] A. Aldoma, Z. Marton, F. Tombari, W. Wohlkinger, C. Potthast, B. Zeisl, R. B. Rusu, S. Gedikli, and M. Vincze, "Tutorial: Point cloud library – three-dimensional object recognition and 6 dof pose estimation," *Robotics & Automation Mag.*, vol. 19, no. 3, pp. 80–91, 2012.
- [10] R. B. Rusu, N. Blodow, Z. Marton, and M. Beetz, "Aligning point cloud views using persistent feature histograms," in *IROS*, Acropolis Convention Center, Nice, France, Sep. 22–26, 2008.
- [11] G. Barequet and M. Sharir, "Partial surface and volume matching in three dimensions," *IEEE Transactions PAMI*, vol. 19, pp. 929–948, 1994.
- [12] F. Tombari and L. D. Stefano, "Hough voting for 3D object recognition under occlusion and clutter," *IPSI Transactions on Computer Vision and Applications*, vol. 4, pp. 20–29, 2012.
- [13] G. Barequet and M. Sharir, "Partial surface matching by using directed footprints," *Computational Geometry*, vol. 12, pp. 45–62, 1999.
- [14] C. Rink, Z. Marton, D. Seth, T. Bodenmüller, and M. Suppa, "Feature based particle filter registration of 3D surface models and its application in robotics," in *IROS*, Tokyo, Japan, Nov. 3–7, 2013, pp. 3187–3194.
- [15] T. Bodenmüller, "Streaming surface reconstruction from real time 3D measurements," Dissertation, Technische Universität München, Munich, 2009.
- [16] W. R. Scott, G. Roth, and J. Rivest, "View planning for automated three-dimensional object reconstruction and inspection," *ACM Comput. Surv.*, vol. 35, no. 1, pp. 64–96, Mar. 2003.
- [17] S. Chen, Y. Li, and N. M. Kwok, "Active vision in robotic systems: A survey of recent developments," *Int. J. of Robotics Research*, vol. 30, no. 11, pp. 1343–1377, Sep. 2011.
- [18] 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.
- [19] L. Torabi and K. Gupta, "An autonomous six-DOF eye-in-hand system for in situ 3D object modeling," *IJRR*, vol. 31, no. 1, pp. 82–100, 2012.
- [20] J. I. Vasquez-Gomez, L. E. Sucar, and R. Murrieta-Cid, "View/state planning for three-dimensional object reconstruction under uncertainty," *Autonomous Robots*, pp. 1–21, 2015.
- [21] U. Thomas, S. Kriegel, and M. Suppa, "Fusing color and geometry information for understanding cluttered scenes," in *IROS, Robots in Clutter Workshop*, Chicago, Illinois, USA, Sep. 14–18, 2014.
- [22] G. H. Givens and J. A. Hoeting, *Computational Statistics*. Wiley, 2005.
- [23] S. Kriegel, "Autonomous 3D modeling of unknown objects for active scene exploration," Ph.D. dissertation, Technische Universität München (TUM), 2015.