Combining Object Modeling and Recognition for Active Scene Exploration

Simon Kriegel, Manuel Brucker, Zoltan-Csaba Marton, Tim Bodenmüller and Michael Suppa

***Abstract*— Active scene exploration incorporates object recognition methods for analyzing a scene of partially known objects and exploration approaches for autonomous modeling of unknown parts. In this work, recognition, exploration, and planning methods are extended and combined in a single scene exploration system, enabling advanced techniques such as multi-view recognition from planned view positions and iterative recognition by integration of new objects from a scene. Here, a geometry based approach is used for recognition, i.e. matching objects from a database. Unknown objects are autonomously modeled and added to the recognition database. Next-Best-View planning is performed both for recognition and modeling. Moreover, 3D measurements are merged in a Probabilistic Voxel Space, which is utilized for planning collision free paths, minimal occlusion views, and verifying the poses of the recognized objects against all previous information. Experiments on an industrial robot with attached 3D sensors are shown for scenes with household and industrial objects.**

I. INTRODUCTION

Real world tabletop scenes are usually partially known, which means that models are available for some, but not all, of the objects in the scene. Regarding robotic tasks such as grasping or manipulating objects, at least the objects that should be interacted with need to be known a priori. Other objects that may be occluded or are not in the field of view (FOV), typically remain unrecognized by an autonomous system. Thus, in order to increase possibilities of interaction with the current and future scenes, additional actions are required. For example, occlusions can be resolved by multiple view points or the recognition database can be extended by acquiring object models.

In robotic perception, the recognition and localization of objects is usually kept separated from environment exploration, modeling for path planning, self localization, and object model extraction. However, for tackling the analysis of partially known scenes in an autonomous way, recognition and exploration have to cooperate as a single scene exploration system. Thereby, exploration can provide useful views from the global model for multi-view recognition, and, vice versa, recognition can refine the global model with object information. Furthermore, the detection of unmatchable data clusters during recognition has to trigger autonomous object modeling and a database update.

In this paper, we present an active scene exploration system, tightly integrating exploration, view planning, object recognition and modeling. This system enables novel functionalities for autonomous operation in real world scenarios such as:

- considering recognized objects during exploration
- combining knowledge from multiple views for detection, improving both exploration and recognition
- use of a probabilistic representation of the explored and unexplored space regions for Next-Best-View (NBV) planning, occlusion avoidance, collision-free path planning and object recognition validation
- dynamic update of an open-ended object model database
- support of multiple sensors for optimal performance
- automatic identification of the number of views that are necessary for obtaining the 3D model quality required for accurate object recognition.

In summary, this work improves and combines existing autonomous 3D modeling and object recognition methods into a unified system and shows its applicability to tabletop scenes with industrial and household objects (see Fig. 1).

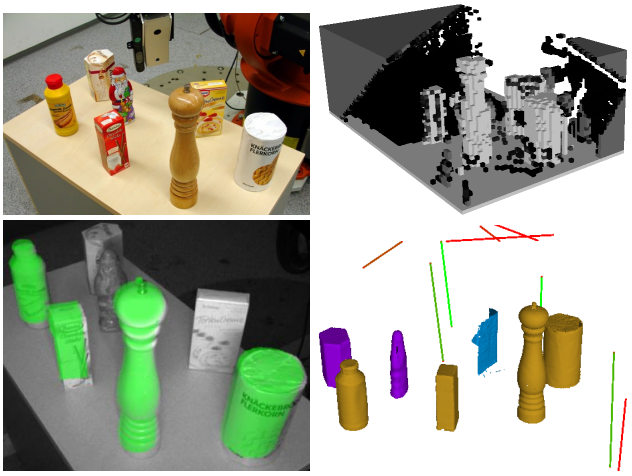

Fig. 1. Scene exploration for an example tabletop scene. *Top left:* scene with 7 household objects. *Top right:* Probabilistic Voxel Space from multiple measurements. The probabilities are color coded from black (almost free), through gray (unknown) to white (occupied). Free space is transparent. *Bottom left:* intermediate scene with recognized objects. *Bottom right:* NBV planning and modeling. The two previously occluded objects (purple) are successfully detected from this view. The flat box remains unknown and is autonomously modeled. The lines show scan path candidates generated from its partial mesh (blue) and their rating (red: low, green: high).

II. RELATED WORK

Scene exploration comprises different topics such as object recognition, robotic exploration, and 3D environment modeling. In the context of this work however, exploration and

This work has partly been supported by the European Commission under contract number FP7-ICT-260026-TAPAS.

The authors are with the Institute of Robotics and Mechatronics, German Aerospace Center (DLR), 82234 Oberpfaffenhofen, Germany simon.kriegel@dlr.de

modeling can be summarized as autonomous 3D modeling. In the following sections, the most recent work in each topic and in scene exploration are summarized.

A. Autonomous 3D Modeling

The affordable and thus widely available RGB-D sensors catalyzed a multitude of efforts for 3D modeling and recognition. One of the most well known applications is KinectFusion [1], with a freely available implementation in the Point Cloud Library[1]. Despite its indisputable uses, the authors show that for reconstruction of objects with KinectFusion, curved and concave details in the scale of around $10\mathrm{mm}$ are lost and simply smoothed out [2]. This is not sufficient if accurate 3D modeling is required.

Several methods have been developed in order to automatically generate 3D models of objects or scenes with minor or without human interaction by NBV planning, as reviewed in [3]. In [4], NBVs are planned for a humanoid robot in order to generate 3D models of single objects placed on a table. However, the modeling was only tested in simulation and not on a real robot. In [5], new objects are learned by grasping them, moving them in front of a 3D camera and planning a NBV regrasp for covering the previously occluded parts. In the case of large or heavy objects, if the scene should not be changed or if a robot lacks manipulation capability, such an approach is not applicable. In [6], a humanoid robot explores uncluttered scenes containing several objects. The work proves that a lot more of the space can be explored with an eye-in-hand camera in contrast to a head camera and it can be explored faster with a Bayesian approach.

B. Object Recognition

For object recognition and pose estimation, geometry based detection methods have the advantage (over purely texture based methods [7]) of being able to handle a wider range of different objects. This is especially important in industrial scenes as, in such contexts, the objects of interest do not usually exhibit plenty of texture information.

Geometric model based object recognition methods can be roughly categorized in two classes: local correspondence and global methods, with an overview presented in [8]. Global methods try to capture either multiple views or the complete geometry of an object. A promising new feature of this type is presented in [9]. However, all of these global features require a pre-segmentation of objects for recognition and usually require special extensions for pose estimation. In contrast, for local correspondence based recognition this is not the case, as points matching the template are found in a complete scene. Due to the increased availability of RGB-D sensors recent work proposes hybrid point features combining geometry and intensity information [10]. The matching between the template and the scene point features can be done using multiple methods, in [8] a RANSAC-based approach was used.

[1]see KinFu on `http://www.pointclouds.org`

Another approach is voting based object detection, prominent examples of which are the generalized Hough transform [11] and Geometric Hashing [12], [13]. The general approach in this class of methods is to establish lots of simple correspondences describing (multiple) possible poses of an object in a scene. These represent votes in 6DOF pose space and dominant clusters of votes are considered probable poses. In this work the method from [12] is extended and adapted to recognition in low-quality depth images.

C. Scene Exploration

Scene exploration is in some sense the extension of multi-view recognition, with the advantage that the object model database gets initialized and expanded by autonomous modeling. In contrast, recognition of a fixed set of objects from multiple viewpoints is presented in [14]. While the work is very promising, the viewpoints are defined beforehand, and the planning step only chooses between them. A more flexible multi-view recognition system is presented in [15], however also lacking the modeling of unknown parts. There, the sensor placement is based on known locations of good features on the object (OCR and bar codes), but these locations need to be manually predefined.

In [16], [17], reconstruction was integrated with detection, but without view planning. Also, modeling was performed using approximation with symmetric shape primitives. As in these works, we also assume that we can individuate all the objects, as the problem of segmentation in clutter is a separate research topic, outside of the scope of this work. It is, however, actively explored for example in [18], for relatively simple (mostly convex) geometric shapes, or in [19], dealing with more complicated shapes, but requiring human guidance.

This work presents a full autonomous scene exploration system with multi-view recognition, modeling of unknown parts, and autonomous sensor placement. Therefore. previous work on autonomous 3D modeling [20] and object recognition [21] is extended and combined. The autonomous object modeling is extended to multiple objects, and sped up by the recognition of objects that are already known. Object recognition is performed by geometric matching of the autonomously acquired 3D models and verifying the candidates through their conformity to the global knowledge of the explored workspace. NBV planning is applied to both modules, the object modeling and recognition, which benefit from each other.

III. SCENE EXPLORATION SYSTEM

The main idea of the scene exploration system is presented in Fig. 2. Two complementary sources of 3D information are used, a 3D camera and a laser striper. This setup is chosen since better model quality can be obtained with a laser striper than with a 3D camera [22]. However, the 3D camera provides a fast overview of the complete scene at once and therefore an initial depth image is obtained from a random position. The dominant plane of the tabletop is detected using RANSAC. Based on the table's extents, a Probabilistic Voxel Space (PVS) is initialized, encompassing the workspace.

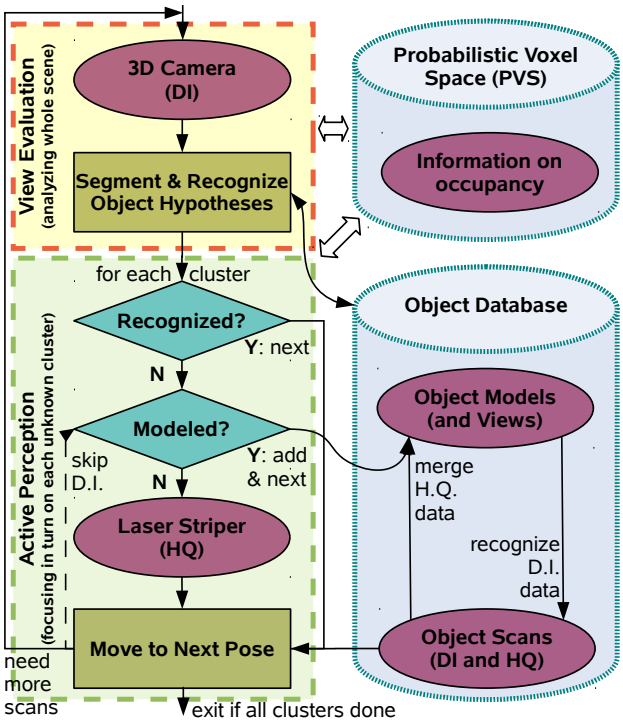

Fig. 2. Scene exploration system overview (DI stands for depth image and HQ for high quality laser scans).

Then, both in the *View Evaluation* and *Active Perception* steps, the PVS is updated for each depth measurement from the 3D camera and laser striper (see III-A), and utilized in order to select a NBV (see III-B).

Since segmented objects are assumed, in the *View Evaluation* step clusters are extracted from the depth image by plane subtraction and euclidean clustering. These are stored in the *Object Database*, which, for each cluster, holds its currently estimated location as a bounding box computed from the depth image, and a merged triangulated model from the high quality laser scans, Furthermore, the database contains a set of a priori known object models.

The *Active Perception* step is performed for each cluster individually. The clusters are iteratively tested against the known objects (see III-C). If for one of them no correct match is found in the database, NBVs are planned in order to obtain more information. In this process objects are re-scanned from different views, potentially discovering new clusters. The *View Evaluation* and *Active Perception* steps are repeated until every cluster is either recognized or modeled.

Since the laser scans are slower and have a limited FOV, recognition is attempted using the 3D camera before each scan. To this end, a single frame is captured each time the robot moves to a new position. This can shortcut the tedious modeling step if known objects are recognized. After N failed attempts to recognize the cluster, the depth image based recognition is skipped (signaled by the dashed upwards arrow in Fig. 2). In order to create complete object models, which are fit for object detection, several laser scans of each cluster are accumulated (see III-D). Finally, the interaction between the modules is discussed in III-E.

A. Exploration

Based on the initial 3D camera depth image, a Probabilistic Voxel Space (PVS) is initialized for the space above the table. It will be used for exploration of the initially unknown environment. For estimating the relevant workspace, the dominant plane is detected and a rectangle is fitted to encompass the table. The rectangle is then projected upwards to create a cuboid, analogous to [23].

The PVS is represented by an octree, where the probability distribution of occupied/free locations is modeled. All measurements from the 3D camera and the laser striper update the probability p_v (0.0 free, 0.5 unknown, 1.0 occupied) of each intersected voxel with Bayes' Rule as in [20].

The PVS is used to verify recognized objects (III-E) and to select NBVs (III-B). In contrast to [20], in this work we do not only consider the entropy of the PVS e_p as NBV selection measure but also the surface quality q_s of regions which will be rescanned (see III-B). If the space around the object, is already completely explored, but the required mesh quality has not been reached yet, the NBV selection exclusively based on entropy reduction is more random. Therefore, not only the probability p_v, but also the relative point density d over all mesh vertices within a voxel is saved in the PVS. Here, point density is defined as percentage of neighbors within a sphere in relation to the maximum possible number of neighbors according to the reduction [24].

Finally, the PVS is also used by a path planner, in order to be able to avoid collisions within the scene when moving the sensor between the objects. Here a probabilistic path planning approach is necessary, since accurately modeling of the objects all around requires to move very close to the objects and therefore to move into the workspace. This was already described in [20], but now, in the case of multiple objects in the scene, is of higher importance.

B. Next-Best-View Planning

In order to find a NBV for the 3D camera or a Next-Best-Scan (NBS) for the laser striper, the method presented in [20] is used and adapted for scenes with several objects. As shown in Fig. 2, object recognition is performed using the current list of object models. All clusters that cannot be matched, require further exploration, since an unknown object is expected. Therefore, these unknown clusters are iteratively processed. Starting with the largest one, a 3D triangle mesh is generated for all depth measurements within a bounding box of this cluster. Then, the *Boundary Search* [25] is applied to this mesh, in order to generate possible viewpoints or scan paths. Thereby, boundaries are detected, the curvature of the object shape is estimated and candidates, which view the estimated surface with an optimal configuration, depending on the sensor type, are determined.

Since the examined scenes can contain several objects, other objects can occlude the target cluster or be in collision with the determined sensor pose. Therefore, similar to [26], the sensor pose that is collision free and produces minimal occlusion is selected from the ones on the sphere around

the object's surface to be scanned. Additionally, candidate views are also removed if the incidence angle is too high for reasonable scan quality. Then for each sensor pose candidate generated by the *Boundary Search*, a depth measurement is simulated within the PVS in order to select a NBV or NBS. Therefore, for each candidate, the surface quality q_s is determined by weighting the average border edge percentage b_i (see [20]) and the average relative point density d_i per voxel i over all k voxels, which are intersected by a beam:

$$q_s = \frac{1}{k}\sum_{i=1}^{k}\left[\lambda \cdot b_i + (1-\lambda)\cdot d_i\right] \qquad \lambda, b_i, d_i \in [0,1] \quad (1)$$

Then, for the view selection measure, we suggest a utility function which does not only consider exploration (entropy reduction e_p as in [20]), but also 3D modeling (mesh quality improvement). The weighting between the two can be adjusted depending on the task:

$$f_{utility} = \underbrace{(1-\omega)\cdot e_p}_{\text{Exploration}} + \underbrace{\omega\cdot(1-q_s)}_{\text{3D Modeling}} \quad (2)$$

Since we want to rescan surface areas with low quality, $1-q_s$ is used for the 3D modeling part. For the first few scans, the exploration part needs to be weighted higher in order to get a rough mesh of the unknown object. Later, the 3D modeling part needs to be considered more, since now the mesh quality should be addressed. Therefore the weight ω is selected such that it is dependent on the scan number n_s:

$$\omega = \frac{n_s}{n_q} \Big/ \left(\frac{n_s}{n_q} + 1\right) \quad (3)$$

The value for n_q is usually selected to be around 8, which means that after scan number 8, the 3D modeling part is given higher priority. After determining $f_{utility}$ according to equation 2 for each pose candidate, the candidate with the highest value is selected as NBV or NBS and can be applied for recognition and modeling of the objects.

C. Object Recognition

In [21] the geometric primitives based sequential scene analysis system of [27] was extended to handle generic triangulated models in complex cluttered scenes. In contrast to the original work, that was assuming a static depth camera, here, the camera is moved freely in space. Thus, the system is extended to make use of multiple different views of the scene. The positional sensor readings of the robot are used to transfer recognized object poses back and forth between the camera's current and the robot's global coordinate system. However, since only the relative movement of the camera between detection steps is relevant, any other sufficiently precise method of estimating camera movements can be employed, making the method independent of the robot.

The original local visual feature based object detection algorithm was found to be ill-suited to handle some of the sparsely textured industrial objects targeted in this work. This, as well as the lack of texture in the accurate models generated by the laser striper, led to the development of a geometry based object recognition method based on [12].

In the original method, a global model for each object is built using a feature similar to a surflet pair feature [28]. Specifically, the feature is a four dimensional feature vector $\vec{F}$ describing the geometrical relation between two points (m_1,m_2) with corresponding normals (n_1,n_2):

$$\vec{F}(\vec{m_1},\vec{n_1},\vec{m_2},\vec{n_2}) = \left(||\vec{d}||_2, \angle(\vec{n}_1,\vec{d}), \angle(\vec{n}_2,\vec{d}), \angle(\vec{n}_1,\vec{n}_2)\right), \quad (4)$$

where $\vec{d} = \vec{m}_2 - \vec{m}_1$ and $\angle(\vec{a},\vec{b})$ denotes the angle between two vectors.

For each of the objects to be detected, a model is generated by sampling points on the object's surface and calculating the features between all point pairs. The generated features are discretized and used as an index to a four dimensional hash table storing the point pairs.

For object detection, a random reference point $\vec{s}_r$ is chosen in the scene data and paired with all other points $\vec{s}_i$ (or a sampled subset thereof) in the scene. For each of these point pairs, similar pairs $(\vec{m}_r, \vec{m}_i)$ are retrieved from the model hash tables. Assuming a correspondence between $\vec{s}_r$ and $\vec{m}_r$, and considering their respective normals $\vec{n}_{sr}$ and $\vec{n}_{mr}$, the pose of the object in the scene is defined up to a rotation around $\vec{n}_{sr}$. The angle α of this rotation can be calculated by aligning $\vec{s}_i$ and $\vec{m}_i$.

Consequently, for each matched pair of point pairs a vote for a correspondence between $\vec{s}_r$ and $\vec{m}_r$ as well as an angle α is cast in a two dimensional accumulator array. The most dominant peaks in the array are considered to be hypotheses for poses of the object in the scene. Since there might be multiple instances of an object in a scene, and there is a chance that $\vec{s}_r$ is not selected from the surface of an object, the process is repeated several times. Finally, all the retrieved pose hypotheses are clustered in pose space and rated according to the total amount of votes the hypotheses in a cluster received. The pose hypotheses of the most dominant clusters are averaged and the resulting poses are considered to be an instance of the object in the scene.

While this original method showed promising results in [12], the evaluation was done exclusively on synthetic data and the very high quality laser scan dataset published in [29]. In contrast to the original method, which is applied to the complete depth data of a view of a scene, the object recognition module in [21] only examines pre-segmented clusters of data. When experimenting with a reimplementation of the method, it was discovered that, for the lesser quality depth images produced by a Kinect like sensor, taking only the most dominant pose hypothesis, did not yield satisfying results. Therefore, instead of generating multiple pose hypotheses from one sample point and clustering all hypotheses from a few sample points, only the most dominant peak in the accumulator array is considered per sample point. However, significantly more sample points are considered in order to increase the chance of picking at least one descriptive point. Furthermore, instead of clustering the generated hypotheses and choosing the most dominant clusters, all hypotheses are evaluated in an additional verification step.

To that end, the objects in question are rendered in the hypothesized poses and the resulting depth buffer is pixel

wise compared with the relevant area of the data cluster in the acquired depth image. Each hypothesis is rated according to how many pixels in the projected image are within a distance threshold to the sensor data. Specifically, the quality q_h of a hypothesis is calculated according to:

$$q_h = \frac{N_\mathrm{m}}{N_\mathrm{r}} \left(\frac{N_\mathrm{m}}{N_\mathrm{c}} \right)^2, \tag{5}$$

where N_m is the number of matching pixels, N_r is the number of rendered pixels, and N_c is the number of pixels in the current cluster. Finally, if the quality of the highest rated hypothesis exceeds 0.4, the corresponding object (and pose) is considered to be present in this view. This threshold is selected to account for possibly incomplete object models and invalid areas in the depth images. This verification procedure is chosen since it can be easily and very efficiently implemented on common computer graphics hardware.

D. Object Modeling

If no object from the current object database can be associated to a cluster, then a complete 3D model for this unknown cluster, which is assumed to describe only one object, is autonomously generated with the laser striper. The NBS poses are generated (as described in III-B), and after each scan obtained with the laser striper the iterative closest point (ICP) algorithm is applied, in order to compensate for the robot positioning error.

The steps laser scan, ICP registration, space update, mesh generation, sensor pose candidate generation and NBS selection are iteratively applied to the cluster until the desired mesh coverage is reached. Therefore, we introduce an estimated mesh coverage $\hat{c}_m$, which calculates the surface area A_{filled} of all triangles in the mesh and estimates the surface area A_{holes} of each hole by a simple hole filling triangulation algorithm. The factor of these describes the coverage:

$$\hat{c}_m = \frac{A_{filled}}{A_{filled} + A_{holes}} \qquad \hat{c}_m \in [0, 1] \tag{6}$$

When the desired mesh coverage $\hat{c}_m$ is reached, the 3D mesh is added to the object database and will be used for object recognition. To ensure a quick object recognition, the final mesh is automatically downsampled by re-meshing the original range data. Still, high-quality models are required for other applications, such as grasp planning.

E. Module Interaction

In summary, the integration of the different modules, described in the previous subsections, creates a system that is more than just a collection of its parts. It enables complex interactions that make the task execution more robust. The object recognition results are fused over different views, improving the estimates of the objects' poses and making the system more robust, which will be shown in IV-C.

Typically, NBVs are planned for exploration, maximizing the information gain for modeling the objects. Here, they are also used for object recognition. Intuitively, both for modeling and recognition, views, that give most information about the unseen and low quality parts of the objects, are required.

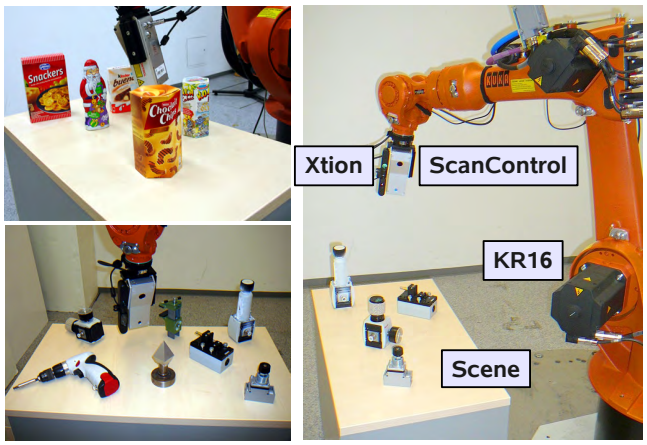

Fig. 3. Overview of experimental setup, household and industrial scenes.

Therefore, NBVs determined for each cluster individually also increase the probability of a good recognition.

Due to the PVS, poses computed from single depth images can be verified against the information coming from all previous measurements. Objects that are matched incorrectly have parts in regions of space that is occluded in the current view but might have been explored in previous steps. Therefore, the PVS assigns low probability to these parts of the objects being located in such regions. If 33% of an object is located in regions of space that are likely free (i.e. cells in the PVS with $p_v < 0.25$), the recognition is considered invalid, and the *Active Perception* step is continued.

IV. EXPERIMENTS AND EVALUATION

During our experiments, a Kuka KR16 industrial robot with a KRC4 controller is used to move the sensor to the determined NBV position within the scene (see Fig. 3). Due to the robot workspace, several but not all poses inside a half sphere around a table can be reached. The pose of the robot, for which the absolute positioning error is in the millimeter range, is synchronized with a laser striper and RGB-D sensor (for details see Tab I). In contrast to [20], we use the ScanControl 2700-100 from Micro-Epsilon with higher accuracy and frequency. Its working area is very limited but the measurements are very accurate. The Asus Xtion Pro is less accurate but also less limited in the FOV.

A. Sensor Comparison

Measurements of both sensors are obtained with two test objects for comparison, and presented in Fig. 4. On the left side, a picture of a whisk and the depth points obtained

TABLE I

COMPARISON OF THE SENSORS UTILIZED FOR DEPTH MEASUREMENTS

Sensor	DIS	FOV	DOF [mm]	FPS
ScanControl 2700	640x1	$14.25°$	$300 - 600$	50Hz
Asus Xtion Pro	640x480	$58°$H $45°$V	> 500	25Hz

DIS (depth image size), DOF (depth of field), FPS (frames per second)

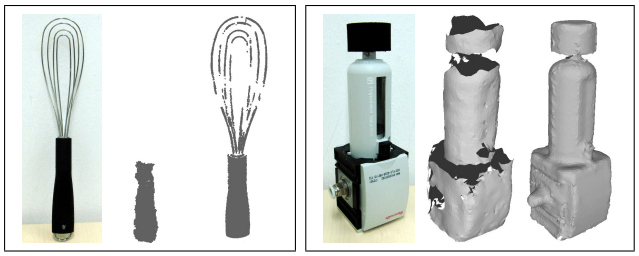

Fig. 4. Comparison for two example objects obtained with the Xtion sensor and the laser striper. From left to right: picture of whisk, depth points from RGB-D and laser sensor, picture of pneumatic filter, 3D mesh using the Xtion and laser striper (both generated autonomously from 12 scans).

from a single shot with the Xtion and a scan with the laser striper can be seen. The Xtion is not able to measure the thin metal wire loops. On the right the mesh of a pneumatic filter generated with the autonomous 3D modeling system from [20] is presented. 12 Depth images were obtained with the Xtion and the laser striper. When comparing the generated 3D models with the CAD data, for the Xtion, the actual mesh completeness is 74.1% and the coordinate root mean square error (CRMS) is 10.6mm. For the laser striper the quality is a lot higher with a completeness of 98.1% and a CRMS of 2.15mm. The Xtion has more difficulties with black or shiny parts, the details are lost in the model and the scale is incorrect due to the higher range measurement error [2]. However for the scene exploration scenario, both sensors complement one another. The RGB-D can be used to explore the workspace and recognize objects and the laser striper is applied to generate accurate 3D models.

B. Object Modeling

First, an evaluation of the modeling step was performed separately, using 2 scenes, one containing 5, the other 7 objects, which are all autonomously modeled. The variations in the proposed quality criteria (see III-A for detail) are presented in Fig. 5. As the number of scans increase, the relative point density d and the estimated mesh coverage $\hat{c}_m$, as suggested in III-D, both increase. The coverage can go down a bit since no ground truth of the model is given. Also the relative point density might slightly decrease when new areas with low point density are scanned. The minimum/maximum values for the coverage are also plotted.

Since the bottom of the objects is not scanable, the estimated coverage starts to stagnate between 75 and 95%.

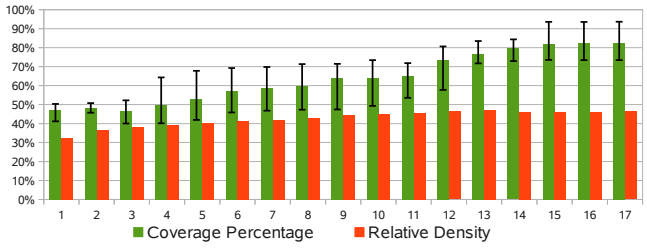

Fig. 5. The average coverage and relative point density as a function of the number of iterations are shown for 12 objects. The error bars represent minimum and maximum estimated coverage.

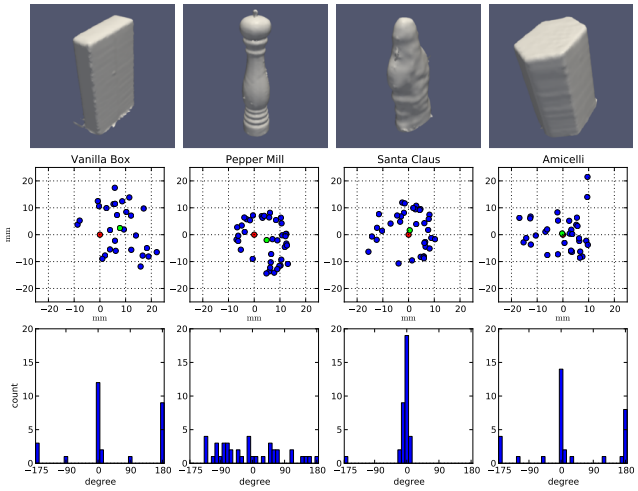

Fig. 6. Detected object positions and angular errors for four selected objects. For better visualization the positions are projected in the xy-plane. Middle row: the estimated translations. The reference translation is shown in red and the average position in green. Bottom row: the corresponding histograms of angular errors.

Therefore 75% was selected as a suitable stopping criterion for modeling in this work. To achieve such coverage, 15 scans are needed in the worst case, and to reach a coverage of 50%, 10 scans are necessary in some cases.

The object modeling with the laser striper takes approximately 17 seconds per scan (scanning: 7s, moving robot in between scans: 6s, NBS planning: 4s), which results in 2-3 minutes per object, depending on the number of scans and object size. However, the robot was not moving at full speed due to safety reasons and the scanning was performed slowly for high point density. Mesh generation and space update are performed in real-time while scanning. Still, for a time efficient scene understanding, the recognition of already modeled objects is clearly necessary. For recognition to work, a coverage of around 75% proved to be sufficient.

C. Object Recognition from Multiple Views

To evaluate the precision of the object recognition module, the scene presented in Fig. 1 was examined. First, as already done in IV-B, all objects in the scene were autonomously modeled to provide both, the object models for recognition, as well as the ground truth pose for pose verification. Then the object detection was run on several randomly sampled views on the scene. In order to get meaningful measures for the precision of the object recognition module, the state of the scene was not tracked in this setup. Therefore, in each view, the object recognition is run from scratch, without any prior knowledge of the objects' poses in the scene.

Specifically, 48 views of the scene consisting of 7 different objects were taken. In total, there were 326 possibly visible object instances in the scenes, not counting object instances that were either out of the FOV of the camera or completely occluded by others. Nonetheless, several heavily occluded and therefore barely visible objects are included. From those, 245 objects were correctly recognized, which means an estimated position was no more than $30\,\text{mm}$ from the actual

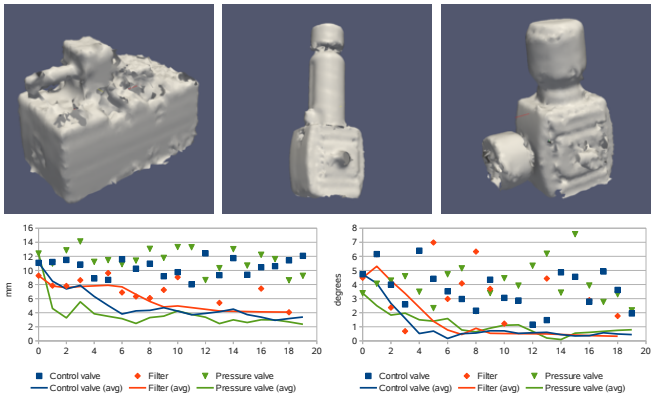

Fig. 7. Top row: The three objects examined in 20 NBV planned views each. Bottom row: Translational (left) and angular (right) errors of the estimates in each view as well as the errors of the running average poses.

known position in the scene. This threshold was chosen large enough to ensure the exclusion of a total of 8 false positives, i.e. object instances that were wrongfully detected in a cluster of data caused by another object.

Fig. 6 shows the distribution of the translational and angular error for four selected objects. For the sake of better visualization only the x and y-component of the translational errors and the angular error around the z-axis is shown. For the translation of the objects it can be observed, that while single estimates exhibit errors of up to 23 mm (in the x/y plane), the average of all poses is considerably closer to the reference poses. The angular error distribution, however, is extremely dependent on the shape symmetries of an object. In case of the rotationally symmetric pepper mill, the distribution is roughly uniform, whereas there are two major peaks for both box-like objects and only one for the asymmetric Santa Clause object.

As suggested by these results, multiple viewpoints can lead to better pose estimation. This was evaluated, first for individual objects in Fig. 7. Three objects were separately placed on a pedestal and recognized in 20 different views generated by the NBV algorithm. The assumption of reasonably well distributed view positions is guaranteed by NBV, even in the case of scenes with partial occlusions, as we will show later.

We used the methods from [30] to compute the extrinsic distance and average of rotations. However, nearest neighbor pose clustering was performed instead of mean-shift, s.t. a separate cutoff value for distance and rotation could be specified. Then the largest pose cluster is selected and its average computed. For displaying the rotation errors, the angle from the axis-angle representation is used, which was computed from the relative transformation between the detected and ground truth rotation.

Due to the simpler setup in this test case the error distributions show smaller overall errors, and all detections formed a single cluster. However, the pose averaging significantly decreased the errors, converging to below 4 mm and 1 degrees respectively. We can observe the steepest decline in the first 5-10 iterations, suggesting that under ideal conditions around 10 views should be enough for high-precision recognition.

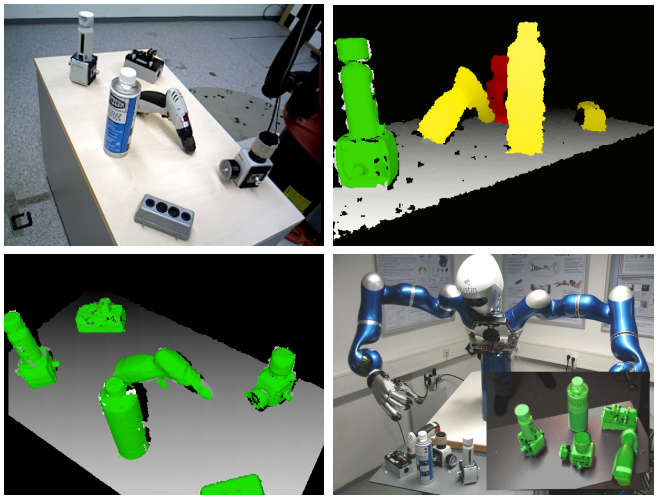

Fig. 8. Industrial scene (top left), initial view (top right), final view after autonomous modeling of unknown clusters (bottom left) and same objects used in a different scene with different sensor and robot (bottom right). The images include detected objects (green) and unrecognized clusters, i.e. occluded (red) and unmodeled objects (yellow).

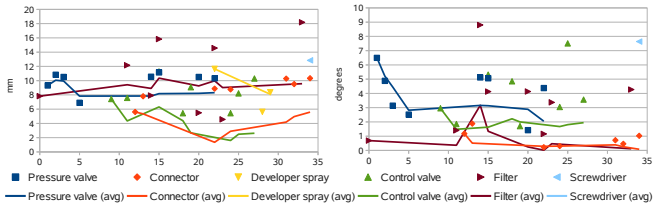

Fig. 9. Translational (left) and angular (right) errors of the estimates in each view as well as the errors of the running average poses of the largest clusters. After objects get modeled (in $\approx$ 12 scans) they start to be recognized. The meaningless rotation errors for the symmetric spray can were left out.

D. Combined Object Recognition and Modeling

Next, we want to demonstrate the performance of the complete scene exploration system on a partially unknown industrial scene (see Fig. 8 top left). Due to surface properties (dark parts, reflectivity) of the industrial objects, this scene proved to be a lot more difficult to handle than the household scene and the generated 3D models are more noisy. In order to generate ground truth for the object poses, each object in the scene, is autonomously modeled. For the pose estimation evaluation, an initial database is created containing a subset of 3 previously modeled objects (see Fig. 7).

As can be seen in Fig. 8 top right, in the initial depth image only 1 of the 3 objects from the database is recognized (the filter) due to occlusion and the limited FOV. Furthermore 4 clusters of unexplainable data are observed, for which NBVs are planned for further observation. Due to the view planning, the remaining 2 known objects, which could not be matched in the initial view, are recognized (pressure valve in view 2 and control valve in view 9). The 3 unknown objects are autonomously modeled and added to the database (connector in view 12, spray can in view 23 and screwdriver in view 34). Finally, the bottom left figure shows a depth image of the last step, where all objects are detected. The bottom

right figure shows that the quality of the generated object models is sufficient to recognize them in a new scene and on a different system with a stereo camera, possibly enabling further application scenarios.

The pose estimation errors are shown in Fig. 9. The improvements of averaging are not as large as for the previous simple case, but still considerable. Additionally, the pose clustering successfully grouped the correct detections, filtering out incorrect ones. Thus, first results of the scene exploration system are promising but require further evaluation concerning e.g. improvement of the autonomous view planning over randomly generated or manual view selection.

V. CONCLUSIONS AND FUTURE WORK

In this paper a scene exploration system is presented, that explores tabletop scenes consisting of household and industrial objects, by joining NBV planning for multi-view recognition of known, and the autonomous modeling of unknown objects. The scene exploration system at a whole and its different aspects have been evaluated with an industrial robot system and a combination of different sensors. This shows that the parts mutually benefit each other in the context of autonomous perception tasks in partially known environments, opening the way for grasping experiments.

Planning NBV poses that can also be inside the analyzed scenes was required for modeling the objects, and it also improved the pose estimation of previously known objects, thanks to the pose clustering and averaging. Concluding, such tight integration of different perception modules is an important step towards completely autonomous systems that can act in real world environments. The use of complementary sensors is beneficial, since RGB-D cameras can speed up object recognition, while laser stripers are best for modeling.

In the future, we want to continue the modeling of a previously modeled object in order to complete the models from the database. This is beneficial in cases when the detected object's pose allows for scanning of previously unmodeled parts (e.g. the bottom). Furthermore, currently only the laser data is used for modeling, so we will target the fusion of the two data sources, i.e. mapping RGB information onto the models, which will enable the use of hybrid intensity/depth object recognition methods.

VI. ACKNOWLEDGMENTS

Our special thanks go to Christian Rink and to Daniel Seth for their support with efficient raycasting, and to Klaus Strobl and Stefan Fuchs for helping with the sensor calibration.

REFERENCES

[1] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon, "Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera," in *ACM UIST*, Santa Barbara, CA, USA, 2011.

[2] S. Meister, S. Izadi, P. Kohli, M. Hmmerle, C. Rother, and D. Kondermann, "When Can We Use KinectFusion for Ground Truth Acquisition?" in *IEEE/RSJ IROS Workshop*, Portugal, Oct. 2012.

[3] W. R. Scott, G. Roth, and J.-F. Rivest, "View Planning for Automated 3D Object Reconstruction Inspection," *ACM Comput. Surv.*, vol. 35, no. 1, 2003.

[4] T. Foissotte, O. Stasse, A. Escande, P.-B. Wieber, and A. Kheddar, "A Two-Steps Next-Best-View Algorithm for Autonomous 3D Object Modeling by a Humanoid Robot," in *IEEE ICRA*, Japan, May 2009.

[5] M. Krainin, B. Curless, and D. Fox, "Autonomous Generation of Complete 3D Object Models Using Next Best View Manipulation Planning," in *IEEE ICRA*, Shanghai, China, May 2011.

[6] C. Potthast and G. S. Sukhatme, "Next Best View Estimation With Eye In Hand Camera," in *IEEE/RSJ IROS Workshop*, San Francisco, CA, USA, Sept. 2011.

[7] A. Collet, M. Martinez, and S. S. Srinivasa, "The MOPED framework: Object Recognition and Pose Estimation for Manipulation," *IJRR*, vol. 30, no. 10, 2011.

[8] A. Aldoma, Z.-C. 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," *RAM*, vol. 19, no. 3, 2012.

[9] M. Madry, C. H. Ek, R. Detry, K. Hang, and D. Kragic, "Improving Generalization for 3D Object Categorization with Global Structure Histograms," in *IEEE IROS*, Villamoura, Portugal, Oct. 2012.

[10] K. Lai, L. Bo, X. Ren, and D. Fox, "Sparse Distance Learning for Object Recognition Combining RGB and Depth Information," in *IEEE ICRA*, Shanghai, China, 2011.

[11] F. Tombari and L. DiStefano, "Hough Voting for 3D Object Recognition under Occlusion and Clutter," *IPSJ*, vol. 4, Mar. 2012.

[12] B. Drost, M. Ulrich, N. Navab, and S. Ilic, "Model globally, match locally: Efficient and robust 3D object recognition," in *IEEE CVPR*, San Francisco, CA, USA, June 2010.

[13] C. Papazov, S. Haddadin, S. Parusel, K. Krieger, and D. Burschka, "Rigid 3D geometry matching for grasping of known objects in cluttered scenes," *IJRR*, vol. 31, no. 4, 2012.

[14] R. Eidenberger and J. Scharinger, "Active perception and scene modeling by planning with probabilistic 6d object poses," in *IEEE/RSJ IROS*, Taipei, Taiwan, Oct. 2010.

[15] D. Stampfer, M. Lutz, and C. Schlegel, "Information driven sensor placement for robust active object recognition based on multiple views," in *IEEE TePRA*, Woburn, MA, USA, Apr. 2012.

[16] G. Taylor and L. Kleeman, *Visual Perception and Robotic Manipulation - 3D Object Recognition, Tracking and Hand-Eye Coordination*, ser. STAR, 2006, vol. 26.

[17] Z. C. Marton, D. Pangercic, N. Blodow, and M. Beetz, "Combined 2D-3D Categorization and Classification for Multimodal Perception Systems," *IJRR*, vol. 30, no. 11, September 2011.

[18] A. Richtsfeld, T. Mörwald, J. Prankl, M. Zillich, and M. Vincze, "Segmentation of Unknown Objects in Indoor Environments," in *IEEE/RSJ IROS*, Vilamoura, Portugal, Oct. 2012.

[19] N. Bergström, M. Björkman, and D. Kragic, "Generating object hypotheses in natural scenes through human-robot interaction," in *IEEE IROS*, San Francisco, CA, USA, Sept. 2011.

[20] S. Kriegel, C. Rink, T. Bodenmüller, A. Narr, M. Suppa, and G. Hirzinger, "Next-Best-Scan Planning for Autonomous 3D Modeling," in *IEEE/RSJ IROS*, Villamoura, Portugal, Oct. 2012.

[21] M. Brucker, S. Léonard, T. Bodenmüller, and G. D. Hager, "Sequential scene parsing using range and intensity information," in *IEEE ICRA*, St. Paul, Minnesota, USA, May 2012.

[22] S. Foix, S. Kriegel, S. Fuchs, G. Alenyà, and C. Torras, "Information-gain view planning for free-form object reconstruction with a 3d tof camera," in *ACIVS*, ser. LNCS, vol. 7517, Czech Republic, Sept. 2012.

[23] R. B. Rusu, Z. C. Marton, N. Blodow, M. E. Dolha, and M. Beetz, "Functional Object Mapping of Kitchen Environments," in *IEEE/RSJ IROS*, Nice, France, Sept. 2008.

[24] T. Bodenmüller, "Streaming Surface Reconstruction from Real Time 3D Measurements," Ph.D. dissertation, Technische Universität München (TUM), 2009.

[25] S. Kriegel, T. Bodenmüller, M. Suppa, and G. Hirzinger, "A Surface-Based Next-Best-View Approach for Automated 3D Model Completion of Unknown Objects," in *IEEE ICRA*, Shanghai, China, May 2011.

[26] F. Prieto, R. Lepage, P. Boulanger, and T. Redarce, "A CAD-based 3D data acquisition strategy for inspection," *MVA*, vol. 15, no. 2, 2003.

[27] G. D. Hager and B. Wegbreit, "Scene parsing using a prior world model," *IJRR*, vol. 30, no. 12, 2011.

[28] E. Wahl, U. Hillenbrand, and G. Hirzinger, "Surflet-pair-relation histograms: A statistical 3D-shape representation for rapid classification," in *3DIM*, Banff, Canada, Oct. 2003.

[29] A. S. Mian, M. Bennamoun, and R. Owens, "Three-dimensional model-based object recognition and segmentation in cluttered scenes," *IEEE PAMI*, vol. 28, no. 10, 2006.

[30] M.-T. Pham, O. J. Woodford, F. Perbet, A. Maki, B. Stenger, and R. Cipolla, "A new distance for scale-invariant 3D shape recognition and registration," in *IEEE ICCV*, Barcelona, Spain, 2011.