Robot-Based Creation of Complete 3D Workpiece Models

David Singer, Dorian Rohner and Dominik Henrich

Abstract

A complete object database containing a model (representing geometric and texture information) of every possible workpiece is a common necessity e.g. for different object recognition or task planning approaches. The generation of these models is often a tedious process. In this paper we present a fully automated approach to tackle this problem by generating complete workpiece models using a robotic manipulator. A workpiece is recorded by a depth sensor from multiple views for one side, then turned, and captured from the other side. The resulting point clouds are merged into one complete model. Additionally, we represent the information provided by the object's texture using keypoints. We present a proof of concept and evaluate the precision of the final models. In the end we conclude the usefulness of our approach showing a precision of around 1 mm for the resulting models.

Keywords

3D models • Camera vision • Perception for grasping and manipulation • Robot-object interaction • Object recognition

D. Singer (✉) · D. Rohner · D. Henrich
Lehrstuhl für Angewandte Informatik III (Robotik und Eingebettete Systeme), Universität Bayreuth, 95440 Bayreuth, Germany
E-mail: david.harrer@uni-bayreuth.de
URL: <https://robotics.uni-bayreuth.de>

© The Author(s) 2022

T. Schüppstuhl et al. (eds.), *Annals of Scientific Society for Assembly, Handling and Industrial Robotics 2021*,
https://doi.org/10.1007/978-3-030-74032-0_24

1 Introduction

In the last years, the usage of robotic manipulators expanded to small and medium sized enterprises (SME) as well as service environments. The tasks for the robot in these cases often depend on their surroundings (e.g. position of workpieces). Therefore, it is necessary for the robot to recognize its environment. Several object recognition methods use an object database, which stores models for all potential workpieces [1–3]. Depending on the underlying recognition method, different types of models are stored, e.g. coloured point clouds or CAD data. The necessary models are not always available in household or SME settings, due to the specialized or rare nature of the objects. Generating these models manually is often a tedious process or requires expert knowledge. Therefore, our goal is to generate these workpiece models in an automated process.

In order to utilize models for an object database, they usually have to fulfil the requirement of completeness, meaning every geometric feature (e.g. faces, edges, vertices or point based representations) and colour feature should be represented. Additionally, our goal is to develop and evaluate a fully automated concept, meaning no human interaction should be necessary in generating the models. As an additional assumption, we only consider rigid objects. To enable easy handling, we assume that the objects have at least one side, on which they can be placed for a stable stand on a horizontal planar surface.

In the following sections, we discuss the current state of the art and identify possible improvements. Based on this, we develop our overall approach and select a suitable model type for the object database. For a single object, we will discuss how the object is recorded from multiple views in different poses and how to generate the final model. We validate our approach and measure the quality of the models regarding their size compared to ground truth data. Finally, we discuss the contribution of this paper and give an outlook.

2 State of the Art

Existing work on this topic can be classified into two groups: On the one hand there are techniques that use additional and specialized hardware to create object models similar to a 3D-scanner, while on the other hand there are robot-based methods that do not use additional hardware except for the robot and sensors.

In the first group, there are approaches that are semi-autonomic but require dedicated hardware. The object is commonly placed on a rotary plate allowing the camera to record all sides of the object. Depending on the concept, multiple cameras on a rigid frame [4] or a single camera on a slide [5] are used to capture the object. Some systems use a robot mounted camera to capture images of the object on a rotary plate [6]. Techniques from this group do not meet our requirements. Firstly, the approaches are not fully automated, since the objects must be presented to the system one by one. Secondly, the generated models are not complete, since the bottom side of the object is not included in the models.

The second group of approaches for the generation of object models make use of robot-based systems. Unlike the techniques of the first group, no additional hardware beside a robot and a camera is necessary. To capture images from multiple views, the object is moved by the robot itself instead of a rotary plate. The object can also be moved indirectly, by placing it on a robot-mounted plate, which is then moved in front of the camera [7]. In another approach, the robot may grasp the object directly and show it to the camera in different poses [8, 9]. Finally, the object can be rotated in place by pushing it with the gripper of the robot, while a stationary sensor generates the model [10].

Some of the techniques based on the robot manipulating the object have strategies to pick up the workpiece by themselves and solve the first requirement regarding the full automation. This can be done if the robot is able to pick objects up by itself [8] and extract them from cluttered scenes [10]. However, approaches from this group still share the drawback, that the object models are not complete since either the bottom side of the object is missing [7, 10] or some parts of the workpiece are occluded by the gripper of the robot [8]. Approaches exist to handle this occlusion by detecting the gripper, meshing the models and performing hole closing algorithms on the meshes [9].

Overall we can conclude, that no approach exists, that generates complete workpiece models only using a robotic manipulator with a mounted depth camera. In this work, we propose a approach to generate complete 3D object models completely automated and without the usage of specialized hardware and occlusions based on the manipulator grasping the object. This allows us to keep the original point cloud based representation of the model.

3 Our Approach

The problem treated in this work can be described as follows: Our input is an unknown number of objects with unknown pose and shape, which we assume to stand on a known table surface. We use a lightweight robot with an eye-in-hand RGBD-camera to create an object database for object recognition and for retrieval of additional information about those objects.

3.1 Basic Approach

Our approach to the aforementioned problem is structured in several parts (see Fig. 1): In a first step, the camera is moved to a predefined position and captures an initial view on the scene. The captured point cloud is then segmented into pixels, representing the background and objects respectively groups of objects by utilizing the assumption of a known table surface. Individual objects respectively groups of objects can then be identified by applying connected components labeling [11] on the segmented point cloud. Following this, one of the objects is extracted from the cluttered scene and transferred to an examination area. To

Fig. 1 Flowchart of our complete approach for one isolated object. The boxes depict the relevant parts of the software framework

extract the objects we use a technique loosely based on the one described in [12]. The basic idea is to try a grasp calculated on the segmented point cloud. If the grasp is successful, we transfer the object to the examination area. Otherwise, the manipulator is used to push the group of objects towards the centroid of the corresponding pile. Afterwards, we repeat the process of grasping and pushing, until at least one object is isolated from the pile. At the examination area, RGB-D images are taken from multiple fixed views to include all sides of the object. Those images are then combined into a so called *half-model* of the object. Afterwards, the object is picked up and turned around to allow for inspection of the previous, not visible bottom side. A second half-model is then generated from images of the turned object. The two half-models are then merged to a complete model of the object, including all of the objects sides.

As a model representation we chose a point cloud representation because with that the input data of the depth camera, which provides a point cloud representation of the scene, doesn't have to be converted. Additionally, by using the coloured point cloud as representation we do not loose any data from the measurement. This allow later creation of more abstract representations, e.g. a polygon-model via meshing or a CAD-model [13] with higher precision. The geometric information of the point cloud is extended by texture information. To reduce data we do not keep the images itself but use well known methods [1] to extract keypoints and their corresponding feature vectors.

3.2 Sensing

The main goal of this work is to generate complete object models in order to be able to recognize the modelled object from every possible direction. To generate such models we must record point clouds from different views to include all sides of the object. These point clouds have to be merged into one single object model. Due to insufficient calibration precision, noise, and imprecise pose estimation of the robot-mounted depth sensor there is an error in the point cloud poses that prevents us from simply merging the different point clouds based on the pose information acquired through the robots kinematic. A common solution to tackle this problem is the usage of the ICP-algorithm [14]. This algorithm geometrically fits the point clouds together to balance out those errors. The main advantage of this algorithm compared to e.g. keypoint based approaches is, that it uses all points from the overlap area and not just some significant ones, which benefits the accuracy since the data basis is much larger.

Fig. 2 An object with many symmetries and few edges and corners is placed on the calibration object (left). Multiple point clouds taken from different views can still be merged with the ICP algorithm due to the additional imposed edges and corners by the calibration object (right)

The drawback of the algorithm is that besides a starting guess also sufficient unambiguous edges and corners are needed, to function properly. To ensure the availability of such edges and corners even with round or symmetrical objects like the salt shaker in Fig. 2, we place the object on a special asymmetrical calibration body. This calibration body is designed to have many clearly identifiable edges and corners but no symmetry axes. Together with a starting guess from the camera pose the ICP algorithm seems suitable for this task. After the transformation is applied to all point clouds, we remove the known calibration object from the half model as it is not needed anymore.

3.3 Grasping and Turning

To be complete, the object model must include all sides of the object. The half-model generated in the previous section however is missing at least the underside of the object. To include the missing side the object has to be turned around. The idea is to grasp the object sideways, lift it up, turn it 180° and place it back on the calibration body to include the underside in the model. The problem with this approach is, that with relatively flat objects, grasping exactly horizontally is not possible due to the gripper geometry as seen in Fig. 3 (left side). As seen in Fig. 3 (right side) the object can not be placed evenly if it was not grasped horizontally. The solution here is to place the object at the edge of the table to allow for horizontal grasping. The resulting procedure is as follows: In a first step the object is transferred to the edge of the table surface. At this position a grasp is planned and executed under the constraint to grasp the object at vertical edges. The object is then lifted, turned by 180 degrees around the A-axis of the NSA-coordinate-system and placed back on the table. After successfully turning, the object is transferred back to the calibration body.

Fig. 3 The geometry of the robots hand (gripper and wrist) does not allow for horizontal grasping of objects lying flat on the tables surface (left). As a consequence the object can not be placed correctly after turning it around (right)

3.4 Merging of the Half-Clouds

After creation of the first half model and turning the object, we repeat the sensing process for the other half. We now have two partial models of the object which need to be fitted together. The geometrical approach used to create the half model from the different point clouds has two major drawbacks here: Firstly, there is no good starting guess available in this case. The object is turned by 180 degrees but we cannot be sure that it stays rotated by this angle. On the one hand, the object will move a bit while grasping and turning. On the other hand, the object may topple based on its shape. While a box-shaped object will likely stay as placed, a pyramid-shaped object will almost certainly topple in one direction. While the toppling of the object is not a problem for the whole process since the previously occluded side is still visible, we can not estimate a meaningful starting guess for the geometrical approach. Secondly, the two half models need not have common corners or edges to allow the ICP-algorithm to run. While this problem could be fixed in the aforementioned case, here it is not possible to use such a calibration object, since this would have to be rotated exactly as the object and would be required to topple in the same way.

An alternative to the geometrical approach are texture based methods such as e.g. SIFT [1], or SURF [2]. These methods are specifically suitable, since we have relatively big overlapping areas at the sides of most objects. The drawbacks with these methods is that they are vulnerable to symmetries and that they depend on overlapping sides. Thus it is not possible to process largely symmetrical and very flat objects like a ruler. In practice we used the SIFT keypoint detection algorithm to extract keypoints and to find correspondences and then used geometric consistency grouping [15] to calculate the correct transformation based on these keypoint correspondences. An example of the merging process can be seen in Fig. 4. Finally, we can apply the calculated transformation to obtain the resulting workpiece model. In the complete process no human interaction is necessary. The models should be complete, due to multiple views for each side, as well as the turning of the object to capture every part.

Fig. 4 Merging of two half-models of a pen box (top) and a tea box (bottom). One half-model is shown in original colours, while the other one as well as the resulting transformed half-model are shown in yellow. The keypoints are depicted in blue and correspondences as green lines, with their respective keypoints in red

4 Experimental Results

The following section describes the validation, where we present a proof-of-concept. Afterwards, we will evaluate the precision of the generated models in regard to the deviation from ground truth data. For our evaluation we used a Franka Emika Panda with a mounted Intel® Realsense™ D435.

4.1 Validation

To validate our approach we first generate a complete model as described above. The results of this process for one workpiece model can be seen in Fig. 5. The workpiece (a) is recorded in two different poses. For each pose multiple point clouds are merged using an initial pose estimation together with the ICP algorithm to provide reasonable half-clouds ((b) and (c)). These are merged afterwards into one complete model (d) using the keypoints to determine the transformation. In an optional step, this point cloud is transformed into a planar Boundary Representation model (BRep) [16] (e). This last step can also be utilized to eliminate noise (black voxels) from the used depth sensor (b–d).

We utilize these models in two scenarios as a proof-of-concept: On the one hand, we record a new point cloud and utilize SIFT to recognize the model in the point cloud. The SIFT keypoints are used to determine correspondences and with the underlying 3D information the 6 DOF transformation from the model to the point cloud is calculated. On the other hand,

Fig. 5 The complete process of one recording: starting with a workpiece (a), we generate two point clouds from multiple views (b) + (c) and merge them into the complete model (d). Additionally, a BRep (e) can be generated

Fig. 6 The results of the proof of concept: the learned model is recognized correctly within a cluttered scene (left). Also, the generated models were transformed into a BRep, added to an object database, and recognized in a reconstructed scene (right)

we apply a surfaced based object recognition method [3] based on boundary representation models (BReps). Our new model is transformed into a BRep by utilizing [13] and added to the object database. Then we capture a scene as a BRep [13] with a Kuka LWR 4 and an Ensensio N10 and use the method from [3] for object recognition. In Fig. 6 the results regarding the object recognition are visible. In both scenarios the objects are classified correctly, which indicates, that the used colour and surface-based features are extracted correctly during the generation of the model. Furthermore, it shows the independence of the models from different depth sensors.

4.2 Evaluation

For our evaluation we generate models from multiple objects and transform them into planar BReps and measure their characteristic lengths (all edges to uniquely describe an objects geometry). For non-planar objects we determine these lengths by segmenting the recorded point cloud into single areas and measuring the length of them. Additionally, we determine the corresponding ground truth lengths, calculate the difference and determine the mean error. To generate these models, we use 20 views overall and turn the object once. This is

Table 1 The results of the evaluation regarding the geometry. Row (a) shows the test objects with their relevant edges in green, row (b) shows the **measured size of the model** and the ground truth. The following row (c) is the mean error. All values are given in [mm]

a)
	112.7 - 115.7	73.3 - 75.2	120.7 - 121.0	120.2 - 120.3
b)	48.8 - 48.8	159.7 - 157.0	48.0 - 46.6	64.1 - 64.0
	118.0 - 116.6	38.7 - 37.1	64.0 - 64.8	76.0 - 76.6
	116.2 - 113.4	37.0 - 36.4		
c)	1.8	1.7	0.8	0.3
a)
	102.7 - 105.3	100.0 - 101.0	62.8 - 63.8	54.5 - 55.1
b)	101.5 - 102.7	154.0 - 156.2	143.2 - 143.5	76.5 - 77.4
	65.0 - 65.3			
c)	1.4	1.6	0.7	0.8

used to evaluate the geometric correctness of the generated models. This measure is relevant due to the fact, that multiple objects have similar form and size (see Table 1). For a correct classification it is therefore necessary to have a precise workpiece model.

The results regarding the size can be seen in Table 1. For each object (first and fourth row) we measured the length of multiple edges (first five objects) or the height and the diameter (last three objects), as shown in the second and fifth row, left values for each object. The other value (right) is the ground truth length. The last row is the mean error. These values indicate multiple results: Overall, the error is around 1 mm with some variance depending on the object and the concrete length inspected. The maximum difference is 3 mm. The minimum is 0 mm, due to imprecision caused by rounding. Nevertheless, a difference lower than 1 mm occurs several times, which shows the precision of our approach. This result may be improved furthermore by using a depth-camera more suitable for close up views or with a higher resolution. Furthermore, the workpiece models are complete with the exception of minor holes in some of the models, which may be closed automatically by our work [17].

Overall we conclude the usefulness of our approach: The final models are precise enough to distinguish similar sized objects. The necessary colour and surface information is extracted and can be used to recognize the objects successfully in two different scenarios. The con-

cept of the generation works autonomously without human interaction and the models are generally complete.

5 Conclusion

In this paper we presented an automated approach to generate complete workpiece models. We developed an overall process with the goal of creating complete models without any human interaction. To do this, a given pile of objects is separated into isolated workpieces by a robot manipulator. One half of each of these objects is recorded while the object is placed on a calibration object. The generated point clouds are merged by using the ICP algorithm with an initial pose estimation. Afterwards, the object is grasped and turned around. The other half is recorded as well. Both half-models are merged using SIFT keypoints. In a validation we showed the functionality of our approach and evaluated its precision.

Future work may include more sophisticated methods regarding the separation of objects, another feature representation or the consistent merging of keypoints. To reduce the amount of required memory it is possible to downsample the resulting models. Furthermore, the amount of necessary views for each half-model to generate a complete model should be evaluated. Additionally the view directions could be calculated by active vision methods and the trajectories should be optimized.

Acknowledgements This work has partly been supported by the Deutsche Forschungsgemeinschaft (DFG) under grant agreement He2696/21 SeLaVi.

References

1. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis* (2004)
2. Bay, H. et al.: SURF: speeded up robust features. *Comput. Vis. – ECCV* (2006)
3. Rohner, D., Henrich, D.: Object recognition for robotics based on planar reconstructed B-rep models. In: *IEEE International Conference on Robotic Computing* (2019)
4. Singh, A., et al.: Bigbird: a large-scale 3d database of object instances. In: *IEEE International Conference on Robotics and Automation* (2014)
5. Kasper, A., et al.: The KIT object models database: an object model database for object recognition, localization and manipulation in service robotics. *Int. J. Robot. Res.* (2012)
6. Banerjee, D., et al.: Robotic arm based 3D reconstruction test automation. *IEEE Access* (2018)
7. Wang, W., et al.: Textured/textureless object recognition and pose estimation using RGB-D image. *J. R.-Time Image Process.* (2013)
8. Venkataraman, A., et al.: Kinematically-informed interactive perception: robot-generated 3d models for classification (2019). [arXiv:1901.05580](https://arxiv.org/abs/1901.05580)
9. Krainin, M., et al.: Manipulator and object tracking for in-hand 3D object modeling. *Int. J. Robot. Res.* (2011)
10. Bevec, R., Ude, A.: Pushing and grasping for autonomous learning of object models with foveated vision. In: *IEEE International Conference on Advanced Robotics* (2015)

11. Wu, K., et al.: Optimizing two-pass connected-component labeling algorithms. *Pattern Anal. Appl.* (2009)
12. Katz, D., et al.: Clearing a pile of unknown objects using interactive perception. In: *IEEE International Conference on Robotics and Automation* (2013)
13. Sand, M., Henrich, D.: Incremental reconstruction of planar B-rep models from multiple point clouds. *Vis. Comput.* (2016)
14. Arun, K., et al.: Least-square fitting of two 3-D point sets. *IEEE Trans. Pattern Anal. Mach. Intell.* (1987)
15. Chen, H., Bhanu, B.: 3D free-form object recognition in range images using local surface patches. *Pattern Recognit. Lett.* (2004)
16. Sand, M., Henrich, D.: Matching and pose estimation of noisy. Partial and planar b-rep models. In: *Proceedings of the Computer Graphics International* (2017)
17. Rohner, D., et al.: User guidance and automatic completion for generating planar b-rep models. *Annals of Scientific Society for Assembly, Handling and Industrial Robotics* (2020)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.