

Creating and using RoboEarth object models

Daniel Di Marco*, Andreas Koch*, Oliver Zweigle*, Kai Häussermann*, Björn Schiessle*, Paul Levi*,
Dorian Gálvez-López†, Luis Riazuelo†, Javier Civera†, J.M.M. Montiel†,
Moritz Tenorth◊, Alexander Perzylo◊, Markus Waibel• and René van de Molengraft*

I. INTRODUCTION

This work introduces a way to build up and use an extensive sensor-independent object model database. In a first step, a cost-effective and computationally cheap way to create colored point cloud models from common household objects by using a Microsoft Kinect camera [1] is presented. Those object models are stored in a world-wide accessible, distributed database called RoboEarth [2], [3]. Finally, the models are used for recognizing the corresponding objects with any kind of camera. In the presented implementation the demonstration was done with both a Kinect and common RGB cameras. The implementation is available as a set of ROS [4] packages¹.

A. Related work

There are multiple instances of 3D object databases available on the Internet today. Popular examples include Google 3D Warehouse [5], the KIT object model database [6] or Willow Garage's household objects SQL database [7]. A common property of these databases is that the object models are stored as triangular meshes. They are mostly of high quality, but object creation requires either a lot of manual work or expensive scanning equipment.

In contrast, the focus of the work presented in this video is on providing a simple and cost-effective way to create object models for object recognition and pose estimation. Instead of triangular meshes, object models are stored as 3D colored point clouds.

II. RECORDING ARBITRARY OBJECTS

For the object recording process the Kinect camera is used in conjunction with a marker pattern. Before starting the object recording process, the target object is placed in the center of the predefined marker pattern on a table (see Fig. 1(a)). Subsequently, the marker pattern is rotated slowly by the user so that the Kinect camera records the object from different views.

The ARToolkit library [8] is used to extract the approximate positions of the markers in the camera's RGB image. To further improve the precision, two lines are selected that cross in the center of the marker. Using the depth information



Fig. 1. Object recording setup (a) and merged point cloud (b)

from the Kinect, we sample at least six 3D points from each line and apply a least-square fitting approach to estimate the line parameters and the marker's position. The intersection point of these two lines gives a better estimation for the respective marker's center point. In case the lines do not meet, the marker's center point is defined as the point where the distance of both lines is minimal, as long as this distance is below a given threshold.

To discard implausible marker positions, we compare the detected marker positions with the a-priori known relative distances between the markers on the marker template. If at least three of the detected marker center points are classified as plausible in this manner, a coordinate system is established with its origin in the center of the marker pattern by applying the Gram-Schmidt process (e.g. in [9]).

Finally, the different recordings can be transformed into the marker pattern center coordinate system and merged into the final object model (see Fig. 1(b) for an example). For further details on this we refer to [10].

III. DATABASE

In the next step, the data created in the recording process is compressed and uploaded to the RoboEarth database, a Hadoop-based distributed database accessible over the Internet². The user may annotate the recordings with an object class, name and a free-form, human-readable description. Also, a simple OWL description is generated and uploaded to make the model usable in future knowledge processing. We use KnowRob [11] as knowledge processing framework.

The description of the database can be found in [12].

IV. OBJECT DETECTION AND POSE ESTIMATION

The user may download one or more of the object model files stored in the RoboEarth database and use them for

* Universität Stuttgart, Germany

† I3A.Universidad de Zaragoza, Spain

• ETH Zurich, Switzerland

◊ Technische Universität München, Germany

* Eindhoven University of Technology, Netherlands

¹<http://www.ros.org/wiki/roboearth>

²<http://api.roboearth.org>



Fig. 2. Pose estimation using a Kinect camera and the method presented in section IV-B. The arrows indicate the position of the camera. The marker pattern is visible on the left.

object recognition and object pose estimation. Each object model consists of several recordings from different points of view. For each recording, a 3D point cloud along with the SURF features [13] associated to some of those points are stored.

There are currently two different algorithms implemented that make use of the objects stored in the database, one for using common RGB cameras and another one for use with the Kinect. Though the recognition process could be moved onto the RoboEarth servers, they are currently implemented to run on the client side. In principle the models can be used with any kind of camera.

A. RGB camera

When an image is acquired with an RGB camera, SURF features are extracted. Then correspondences are calculated between each view of the object model and the camera image. Using the RANSAC algorithm, at least five correspondences that describe a valid transformation from the model image to the camera image are searched, and the object pose is estimated by solving the Perspective-N-Point problem. This approach is described in detail in [14].

B. Kinect camera

The detection method employed for use with a Kinect sensor is similar to the method presented in the previous subsection. In this case, the pose is estimated with a rigid transformation between the model and the camera point clouds. Additionally, the depth information for the feature points estimated in the camera image is used to compare the distances between given feature points with the distances in the respective object model feature points. This check is used to discard more implausible correspondences. An example for successful pose estimation in a point cloud acquired from a Kinect camera is shown in Fig. 2.

V. CONCLUSION

In this work we presented an approach to create 3D object models for robotic and vision applications in a fast

and inexpensive way compared to established approaches. By using the RoboEarth system for storing the created object models users have world-wide access to the data and can immediately reuse a model as soon as it was created and uploaded. The approach shows general applicability for different kinds of cameras. In this work this was shown by two example implementations for the recognition process of objects. The quality of the recognition can be verified in the video. Combined with the knowledge saved in the RoboEarth database the objects can also be properly classified.

The complete software in conjunction with the RoboEarth platform is already available for download. For further details please see [2].

VI. ACKNOWLEDGMENTS

The research leading to these results has received funding from the European Union Seventh Framework Programme FP7/2007-2013 under grant agreement no 248942 RoboEarth.

The authors gratefully acknowledge the contributions by the members of the RoboEarth project and the contributors of the ROS project.

REFERENCES

- [1] Microsoft. Kinect camera. <http://www.xbox.com/kinect>. Accessed September 13, 2011.
- [2] RoboEarth Consortium. Roboearth. <http://www.robearth.org/>. Accessed September 13, 2011.
- [3] M. Waibel, M. Beetz, J. Civera, R. D'Andrea, J. Elfring, D. Galvez-Lopez, K. Häussermann, R. Janssen, J.M.M. Montiel, A. Perzylo, et al. Roboearth. *Robotics & Automation Magazine, IEEE*, 18(2):69–82, 2011.
- [4] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng. Ros: an open-source robot operating system. In *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA) Workshop on Open Source Robotics*, Kobe, Japan, May 2009.
- [5] Google. 3D warehouse. <http://sketchup.google.com/3dwarehouse/>. Accessed September 13, 2011.
- [6] Karlsruhe Institute of Technology (KIT). KIT object models web database. <http://i61p109.ira.uka.de/ObjectModelsWebUI/>. Accessed September 13, 2011.
- [7] Willow Garage. Household objects database. http://www.ros.org/wiki/household_objects. Accessed September 13, 2011.
- [8] H. Kato and M. Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *Augmented Reality, 1999 (IWAR'99) Proceedings. 2nd IEEE and ACM International Workshop on*, pages 85–94. IEEE, 1999.
- [9] G. B. Arfken, H. J. Weber, and F. Harris. *Mathematical Methods for Physicists, Sixth Edition: A Comprehensive Guide*. Academic Press, 6 edition, 7 2005.
- [10] A. Koch, D. Di Marco, J. Winkler, and K. Häussermann. 3D Object Recording and Storing in the ROBOEARTH Database. Preprint, http://www.ipvs.uni-stuttgart.de/abteilungen/bv/abteilung/mitarbeiter/Daniel.DiMarco_infos/object_recording.pdf. Accessed September 16th 2011.
- [11] M. Tenorth and M. Beetz. Knowrob – knowledge processing for autonomous personal robots. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 4261–4266. IEEE, 2009.
- [12] B. Schießle, K. Häussermann, and O. Zweigle. Deliverable 6.1: Complete specification of the RoboEarth platform. Technical report, December 1, 2010. <http://www.robearth.org/wp-content/uploads/2011/03/D61.pdf>.
- [13] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. *Computer Vision–ECCV 2006*, pages 404–417, 2006.
- [14] J. Civera, D. Galvez-Lopez, L. Riazuelo, J.D. Tardos, and J. M. M. Montiel. Towards semantic slam using a monocular camera. To appear. *International Conference on Intelligent Robots and Systems*, 2011.