Draft Version. Final version published in European Conference on Mobile Robots (ECMR), 2015.

OLT: A Toolkit for Object Labeling Applied to Robotic RGB-D Datasets

J.R. Ruiz-Sarmiento and C. Galindo and J. Gonzalez-Jimenez
Dept. of System Engineering and Automation, University of Málaga, Spain.
{*jotaraul, cgalindo, javiergonzalez*}*@uma.es*

***Abstract*—In this work we present the *Object Labeling Toolkit* (OLT), a set of software components publicly available for helping in the management and labeling of sequential RGB-D observations collected by a mobile robot. Such a robot can be equipped with an arbitrary number of RGB-D devices, possibly integrating other sensors (e.g. odometry, 2D laser scanners, etc.). OLT first merges the robot observations to generate a 3D reconstruction of the scene from which object segmentation and labeling is conveniently accomplished. The annotated labels are automatically propagated by the toolkit to each RGB-D observation in the collected sequence, providing a dense labeling of both intensity and depth images. The resulting objects' labels can be exploited for many robotic oriented applications, including high-level decision making, semantic mapping, or contextual object recognition. Software components within OLT are highly customizable and expandable, facilitating the integration of already-developed algorithms. To illustrate the toolkit suitability, we describe its application to robotic RGB-D sequences taken in a home environment.**

Fig. 1. Example of a kitchen reconstructed from a sequence of RGB-D observations within a robotic dataset. The appearing objects have been labeled (colored boxes). Gray spheres stand for RGB-D sensor poses.

I. Introduction

A comprehensive dataset supposes a valuable benchmark tool for tuning, testing, and comparing robotic algorithms and systems in a convenient and fair way. Public datasets consisting of intensity images [1]–[3] have largely helped researchers to push ahead the state-of-the-art in object recognition or scene interpretation. Nowadays, given the increasing number of capabilities and applications that are demanded to a mobile robot, e.g. semantic mapping [4], high-level decision making [5], or contextual object recognition [6]–[9], new particularly oriented datasets are required.

RGB-D cameras have become a key source of information for such *robotic* datasets. Although the sensory data of these datasets may be conveniently gathered by the mobile robot itself, human supervision is still needed to segment objects and to label them, i.e. to add annotations over portions of the observed data as belonging to a certain object class, e.g. floor, table, lamp, etc. This is the motivation for the software toolkit that we have developed and is described in this paper.

More specifically, we present the *Object Labeling Toolkit* (OLT) to provide the robotic community with a tool to efficiently label datasets compound of sequences of RGB-D observations, gathered from an arbitrary number of RGB-D sensors. For that, the toolkit builds a 3D reconstruction of each RGB-D sequence within a given dataset, and allows the user to graphically label objects within that reconstruction (see Fig. 1). This ground truth annotations are automatically propagated to all the RGB-D observations without requiring human supervision, resulting in a dense labeling of both intensity and depth data.

OLT comprises a number of software components covering the following functionality: i) dataset pre-processing, ii) localization of RGB-D observation poses, iii) 3D scene reconstruction, iv) labeling of the reconstructed scene, and v) automatic propagation of annotated labels (see Fig. 2). Some of these functionalities can exploit additional information coming from sensors usually present in a robotic platform, e.g. the robot pose estimation computed from 2D laser scans. All the components are highly customizable in order to fit the particularities of robotic datasets, and can be easily expandable to integrate other algorithms of interest. OLT is publicly available under a GNU General Public License at (http://mapir.isa.uma.es/work/object-labeling-toolkit), and it resorts to the Mobile Robot Programing Toolkit (MRPT [10]) and the Point Cloud Library (PCL [11]) for point cloud registration and smoothing algorithms, and for data representation and visualization purposes. Aiming to illustrate the toolkit suitability, we show how it is employed for segmenting and labeling a robotic dataset from a home environment, and also describe its impact on the required processing time w.r.t. a typical manual solution.

II. Related Work

In general, RGB-D datasets providing labeled objects information can be grouped into: *object-centric*, *single-view*, and *sequential-view* datasets. *Object-centric* datasets [12]–[15] provide labeled RGB-D observations of isolated objects, a poor

source of information for many robotic applications like scene understanding or contextual object recognition. On the other hand, *single-view* datasets [16]–[20], are compounded of labeled RGB-D observations of particular scenarios (e.g. a room or an office). This information is richer from the point of view of those applications, but data of the whole robot environment is not available. Finally, *sequential-view* datasets [21], [22] provide a sequence of labeled observations covering the whole inspected workspace, which is the best suitable option for testing trending robotic algorithms or systems. Unfortunately, their number is quite limited mainly due to the arduous labor that entails the data processing.

RGB-D datasets carry out the tedious object labeling task in different ways. Some works resort to *Amazon Mechanical Turk* (AMT) to label their intensity images [16], [18], [19], usually through a labeling tool like LabelMe [2], but this merely divides the workload, and the annotated information still needs to be thoroughly checked to fix incoherent labels. Another approach is the manual labeling of *key intensity frames* from a sequence, propagating these labels to the remaining RGB-D observations [21], [22], but this is only suitable for sequences with simple sensor trajectories, and additionally shows the same limitations as the AMT option. There are also works that reconstruct a 3D representation of the inspected scene and annotate the objects appearing on it [17], but there is not a *labeling feedback* to the RGB-D observations' sequence(s). Similar works to our approach are [12] and [15], where the ground truth annotations over a reconstructed scene are also propagated to the individual RGB-D observations employing an ad-hoc software which, to the best of our knowledge, is not publicly available. We contribute in this paper with an open source solution conveniently divided into configurable components, which provides the robotic community with a number of functionalities towards an efficient labeling of arbitrarily large collections of RGB-D data.

III. Dataset Management: OLT Toolkit

The *Object Labeling Toolkit* (OLT) is a set of software components aimed to facilitate the management and processing of robotic *sequential-view* datasets. Concretely, it provides robotic researchers with the needed tools for achieving a dense labeling of the objects appearing in each RGB-D observation within a dataset sequence, aiming to drastically reduce the user participation in the process. It has been designed to be flexible: it handles datasets containing an arbitrary number of sensors providing RGB-D and (optionally) 2D scans information, and its components can be used independently according to the user needs, or even occasionally expanded with the integration of additional algorithms providing the same functionality.

Figure 2 shows an overview of the software components within the toolkit and their interrelations. In a nutshell, the labeling process of RGB-D data within a certain dataset sequence starts with a pre-processing step, which sets the extrinsic and intrinsic parameters of the sensors employed during its gathering (Sec. III-A). Then, the sensor poses in a global frame from where each RGB-D observation was taken are computed through the alignment of their depth information (Sec. III-C). This component can optionally employ a geometric map built upon 2D laser observations from the same dataset sequence (Sec. III-B). This permits the component to perform a rough robot localization within the explored area, hence giving a useful initial guess for such RGB-D sensor poses' computation. The resultant information is used to three-dimensionally reconstruct the scene, and the goodness of such a reconstruction can be visually inspected (Sec. III-D). The reconstructed scene is then manually labeled by an human operator (Sec. III-E), i.e. the objects appearing in the scene are annotated with their belonging classes, e.g. floor, table, book, etc. Finally, those annotated labels are propagated to subsequent RGB-D observations (both intensity and depth images) of the dataset making use of the computed sensor poses (Sec. III-F). This labeling process can be repeated for an arbitrary number of RGB-D sequences within a given dataset. It is worth to mention that each toolkit component resorts to a configuration file to easily fit their behavior to the requirements of a given dataset.

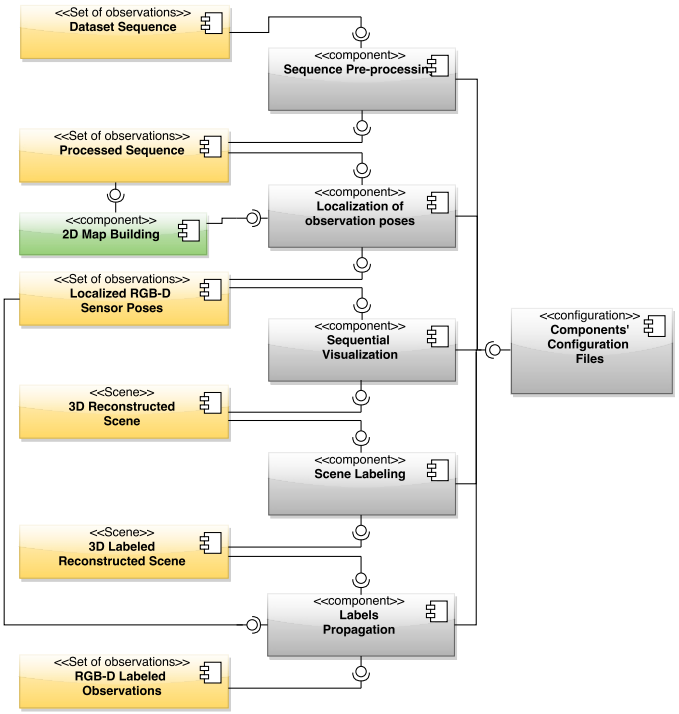

Fig. 2. Components diagram showing their collaboration within the OLT Toolkit. Gray boxes represent components provided by the toolbox, the green box is a component supplied by the MRPT library, and yellow boxes stand for persistent data. The reader is referred to the *online* version of this work for references to colors.

The toolkit components are built upon two widely used libraries: the *Mobile Robot Programming Toolkit* (MRPT [10]), and the *Point Cloud Library* (PCL [11]). We resort to MRPT to manage datasets into the *Rawlog common robotics dataset format*, which are capable of handling any variety of robotic sensor with precise timestamping[1]. This library also provides efficient visualization tools and implementations of point cloud registration algorithms. On the other hand, we rely on PCL to incorporate point cloud smoothing and registering techniques.

[1]There exist a number of tools to convert datasets captured by other popular middlewares to *rawlogs*, e.g. ROS (http://wiki.ros.org/mrpt_rawlog).

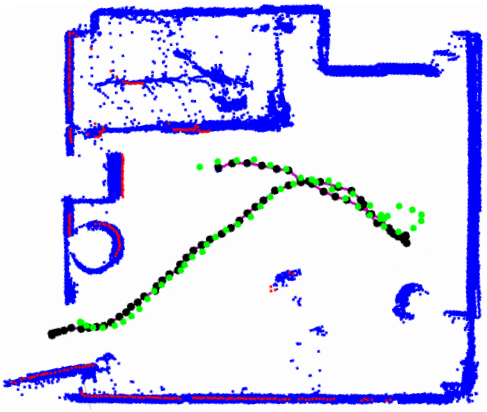

Fig. 3. In blue, geometric map of a bedroom built employing the 2D laser scans within a sequence. In red, example of 2D laser scan aligned within the geometric map. Black circles represent the robot localizations at the time instants when the 2D laser scans where gathered, and green ones the computed poses of the RGB-D sensor. The dataset sequence started at the room's door.

Fig. 4. Left, point clouds representing a bed-set and a pair of shoes reconstructed employing the sensor poses yielded by the robot localization. Right, the same objects reconstructed with the sensor poses refined with GICP.

A. Dataset pre-processing

The first toolkit component sets the extrinsic and intrinsic parameters of the sensors used to gather the dataset sequence being processed. The extrinsic parameters refer to the position of the sensors with respect to the robot centroid, and can be retrieved in different ways [23], [24]. The intrinsic parameters describe geometric and distortion properties of the sensors. RGB-D devices show a different set of intrinsic parameters for their intensity and depth cameras, including: focal length, principal point coordinates, and radial and tangential distortions. Also needed are extrinsic parameters to relate the position of both cameras. The intrinsic parameters largely differ among RGB-D devices, so it is recommended to calibrate them through algorithms like [25]. Those extrinsic and intrinsic parameters can be conveniently introduced into a configuration file, and this component will set them throughout all the contained observations within the dataset sequence.

This pre-processing step permits the user to effortlessly change the sensor(s) calibration parameters within arbitrarily large dataset sequences, enabling in this way the comparison of the results yielded by the following toolkit components when employing different calibration techniques/parameters.

B. 2D map building

The utilization of this component is optional, but it has shown to improve the results obtained during the computation of the RGB-D sensor poses (Sec. III-C). To employ it, the dataset sequence must provide 2D laser observations from, at least, one laser range scanner. These observations are then processed by an ICP-based (Iterative Closest Point) technique [26] within the *icp-slam* MRPT application in order to generate a geometric map. Figure 3 shows an example of a map from a bedroom built in our experiments.

C. Observation poses

This component aims to find the sensor poses from where each RGB-D observation was taken within a 6D global frame (3D position: x,y, and z, plus three attitude angles: yaw, pitch, and roll). This sensors' localization can be performed following any of these two approaches:

i) For each RGB-D observation o_i gathered by a sensor d, it is carried out an alignment process with the observation o_{i-1} previously taken by the same device. For that, it is employed a registration algorithm that exploits their depth information in the form of point clouds. This registration yields the rigid transformation $T_{o_i,o_{i-1}}$ between the two sensor poses, from which we can compute the sensor location where the observation o_i was taken:

$$S_{o_i} = T_{o_i,o_{i-1}} \oplus L_d \tag{1}$$

where L_d stands for the pose of the sensor d on the robot frame (i.e. its extrinsic parameters). The first observation o_1 from such a sensor is considered to be taken with the robot in its initial position, i.e. at the origin of the global frame.

ii) The second approach employs the 2D geometric map from the previous component and the 2D laser observations to localize the robot within the global frame by means of ICP. Then, the sensor poses for each RGB-D observation o_i are computed through the interpolation of those robot localizations employing their timestamps:

$$R_{o_i} = R_1 \oplus ((R_2 \ominus R_1) \cdot t_{elapsed}) \tag{2}$$

$$S_{o_i} = R_{o_i} \oplus L_d \tag{3}$$

where R_1 and R_2 are the robot locations with timestamps just before $(t-1)$ and after $(t+1)$ the o_i one (t), and $t_{elapsed} = (t-(t-1))/((t+1)-(t-1))$ is a scalar value. In this case, the global coordinate frame is specified by the geometric map. Optionally, these locations can be refined through the approach described in i) in a post-processing step (see Fig. 4). Fig. 3 shows an example of robot locations and RGB-D sensor poses from a *bedroom sequence*.

The toolkit user can choose between two different point clouds registration algorithms: the ICP-3D method within the

MRPT library, and the implementation of the Generalized-ICP algorithm [27] from PCL. In addition to the sensor localization and point clouds registration algorithms to be used, a number of options can be selected from the component's configuration file:

- *Point clouds smoothing*. Depth observations from a RGB-D device are prone to provide noisy measurements over surfaces, effect that notoriously increases with distance. This option permits to apply a smoothing method to such depth information before operating with them. Concretely, we rely on the implementation of the *Fast Bilateral Filter* algorithm [28] within PCL.
- *Memory utilization*. This option enables an incremental registration through the use of a *memory of observations*, i.e. when a RGB-D observation taken at time t from a certain sensor is being registered, all the previous registered observations from all RGB-D sensors are considered. This increases the quality of the alignment results at the expense of a higher computational/time cost.
- *Key poses*. When enabled, only observations taken from *considerably* different robot poses are processed. This is useful in cases where the robot speed during the dataset gathering was too slow, so quite similar observations are collected. In the current implementation two poses are considered different according to two user defined parameters: minimum euclidean distance, and minimum rotation angle difference. This option relies on the robot locations yielded by the second localization approach.

The output of this component is a dataset sequence with the poses of the RGB-D observations set according to the their yielded localization into the global frame.

D. Sequential visualization

The goal of the *sequential visualization* component is twofold: first, it permits the user to visually inspect the results of the RGB-D sensor poses localization, and second, it creates a 3D reconstruction of the scene. Concretely, the colored point clouds from the RGB-D observations are projected from its local sensor frame to the global one. For that, given an observation o_i and its sensor pose S_{o_i}, each point P_j in its point cloud is projected as follows:

$$P_{j,G} = S_{o_i} \oplus P_{j,L} \tag{4}$$

being $P_{j,L}$ the point 3D coordinates in the sensor local frame. Once the point clouds have been projected, they are sequentially prompted to the user employing visualization tools from MRPT, which in turn resorts to *octrees* and *OpenGL*. The user can opt for a step by step visualization that adds a new registered point cloud when any *key* is pushed, if s/he needs to inspect the scene reconstruction in detail. Once the reconstruction has been shown, it is created a *scene* file containing the resultant colored point cloud map of the whole scene (see first column in Fig. 6).

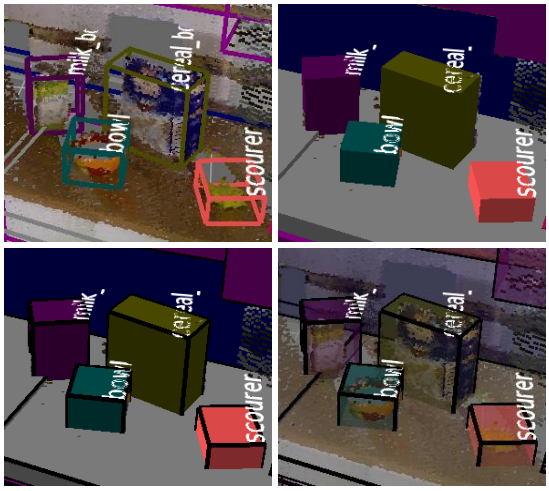

Fig. 5. Example of the four box visualization modes during the labeling of the reconstructed scene.

E. Label reconstructed scene

The labeling of the reconstructed scene is performed by manually fitting *boxes* to the objects appearing in it. We have chosen boxes as the geometric primitives given their easy operation and intuitive fit to objects showing different shapes. Thus, for each object to be labeled in the scene, the user creates and edits a box B_i by setting its position, scale and rotation so the object is fully contained in it. When such an editing is completed, each box can be annotated with its ground truth class, e.g. table, chair, wall, book, etc, conforming a *box-label* pair $(B_i, label_i)$. It is also possible to label the scene for *instance object recognition* purposes, i.e. algorithms trying to recognize particular instances of objects instead of their general category, by adding an identifier to these annotations, e.g. *book_1* or *bed_red*. Complex objects can be labeled employing multiple boxes. The box editing operations, as well as the functionality described below, can be conveniently performed by means of keyboard shortcuts.

In order to facilitate the labeling process, the user has available a number of options: check at any moment a list of the already inserted boxes, add an arbitrary number of boxes, and edit/remove an existing box. Additionally, there are four different box visualization modes: *wireframe*, *solid*, *solid with borders*, and *transparent solid with borders*, which have resulted extremely useful during our tests to visually check the inner points for each box (see Fig. 5). When the labeling is finished, the work done can be saved to a *scene* file containing the initial reconstructed scene and the set $B = ((B_0, label_0), ..., (B_N, label_N))$ of inserted boxes along with their labels, being N the number of objects appearing in the scene (see second column in Fig. 6).

F. Labels propagation

The last component in the toolkit is in charge of propagating the labels into the reconstructed scene to each RGB-D observation within the dataset sequence. For that, given an observation o_i, for each point P_j in its point cloud representation it is checked in which boxes $B_0, ..., B_N$ the point lies inside. It is recalled that the point cloud of both the observation

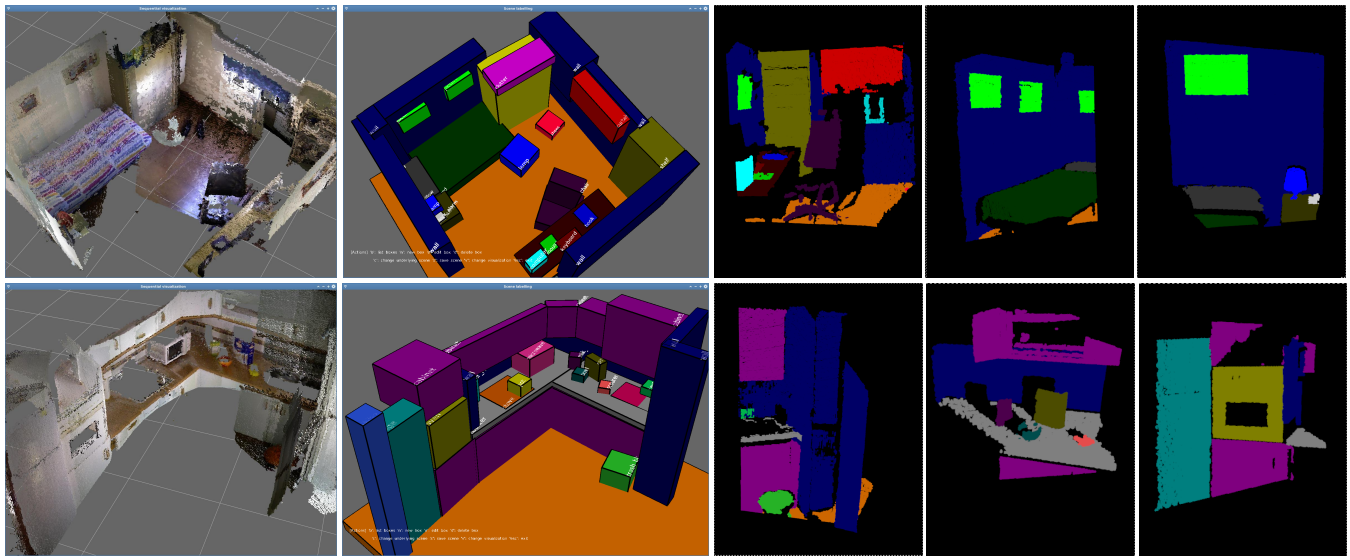

Fig. 6. First column, reconstructed scenes from the sequences within the dataset. Second column, labeled reconstructed scenes. Third-fifth columns, examples of individual point clouds from RGB-D observations labeled by the propagation of the annotations within the reconstructed scenes.

and the labeled scene are in the same coordinate frame thanks to the previous sensor pose localizations, so no additional transformations are needed. Then, if a certain point P_j lies inside a box B_k, for what we employ ray casting with the boxes' boundaries, it is annotated with the box associated label ($label_k$). Since the correspondence between such a point and the (x, y) coordinates of its associated pixel into the intensity and depth images from o_i are known, such images are also annotated. Thus, they are labeled at once the point cloud, the intensity image, and the depth image. The repetition of this process for each RGB-D observation within the dataset sequence completes the labeling pipeline.

IV. Toolkit Usage

This section aims to illustrate the usability of OLT, showing its virtues for an effortless labeling of RGB-D sequences. For that we have collected a home environment dataset employing a Giraff commercial robot [29] enhanced with an RGB-D device (Asus XTion Pro Live [30]), and a 2D laser scanner (Hokuyo model URG-04LX-UG01 [31]). The robot was tele-operated in two different sessions, fully inspecting a kitchen in the first session, and a bedroom in the second one. Each session produced a sequence of observations compound of data from the two added sensors, summing up a total of 77 RGB-D observations and 142 laser scans.

According to the functionality provided by the toolkit, the dataset sequences were preprocessed to set the sensors' calibration parameters (recall Sec. III-A), and the 2D laser scans were used to build a geometric map for both, the kitchen and the bedroom (see Sec. III-B). These maps were used to localize within them the sensor poses from where the RGB-D observations were taken. As it was explained in Sec. III-C, those geometric maps are not an indispensable requirement for such a localization, but they have shown to provide useful cues for improving the registration of RGB-D observations. Once localized, the RGB-D observations are registered, forming a 3D reconstruction of both scenes as it is shown in the first column of Fig. 6 (recall Sec. III-D). These reconstructions are then manually labeled by a human operator that disposes of an intuitive list of options to fit boxes to the scene objects, and annotates them with their respective belonging classes. Notice that this is the unique point in the toolkit where human intervention is needed. They were labeled in total 59 objects, belonging to 39 different classes. The second column in Fig. 6 shows both labeled scenes. Finally, the annotated information is automatically propagated to all the RGB-D observations within the kitchen and bedroom sequences, resulting in an efficient labeling of their intensity and depth images[2]. Fig. 6 shows a number of labeled point clouds.

Regarding the time spent in labeling, the human operator needed 2 hours to annotate both the kitchen and the bedroom scenes, spending on average 2 minutes per object. To compare this with the labeling of all the RGB-D observations individually, we followed the typical intensity image labeling approach and annotated 5 non-consecutive observations from each sequence, extrapolating the results to the whole dataset. This yields a total of $\sim$3 hours needed for the labeling of the kitchen sequence, and $\sim$7 hours for the bedroom, which clearly illustrates the benefits of the toolkit utilization. When following such a typical approach we found problems to accurately label the objects' boundaries, and with objects partially occluded and with an unclear belonging class, drawbacks that are mitigated with the utilization of the proposed toolkit.

It is worth to mention an advantage of the utilization of a geometric map to localize sensor poses when sequences to be labeled are gathered from the same places captured at different times. In this case, the labeling performed for a sequence can be loaded into the reconstructed scene of other sequence, so only the boxes associated to moved/appearing/disappearing objects have to be modified/added, resulting in an additional time saving.

[2]Recall that each point in the point cloud is associated with a pixel from the depth image, and given that this image and the intensity one are registered, the labeling of both images from the point cloud is straightforward.

V. Conclusion and Future Work

In this work we have presented the *Object Labeling Toolkit* (OLT), a publicly available software solution for the management of arbitrary large robotic datasets (http://mapir.isa.uma.es/work/object-labeling-toolkit). The major goal of OLT is to provide the robotic community with a tool to efficiently label objects appearing in a sequence of RGB-D observations. It has been also presented the flexible, highly customizable software components aiming to fit the needs of particular robotic datasets. The toolkit can handle different platform setups, i.e. datasets gathered by an arbitrary number of RGB-D sensors, and even can profit from 2D laser scanners, devices that are usually present in a mobile robot. We have illustrated how OLT is applied to the labeling of a home environment dataset, and show that it considerably decreases the time needed by an human to complete such a task.

The toolkit is in constant development with the inclusion of new features and functionalities. For example, we are studying the incorporation of algorithms for a globally consistent alignment of the RGB-D observations used to reconstruct a scene. We also plan to integrate, in addition to boxes, different geometric primitives to be used during the labeling of the reconstructed scenes, e.g. spheres. OLT welcomes any contribution from the robotics community.

Acknowledgment

This work has been supported by the Spanish grant program *FPU-MICINN 2010* and the Spanish projects *TAROTH: New developments toward a Robot at Home* (DPI2011-25483) and *PROMOVE: Advances in mobile robotics for promoting independent life of elders* (DPI2014-55826-R).

References

[1] M. Everingham, L. van Gool, C. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[2] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "Labelme: A database and web-based tool for image annotation," *Int. J. Comput. Vision*, vol. 77, no. 1-3, pp. 157–173, May 2008.

[3] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *CoRR*, vol. abs/1409.0575, 2014.

[4] A. Pronobis, P. Jensfelt, K. Sjöö, H. Zender, G.-J. M. Kruijff, O. M. Mozos, and W. Burgard, "Semantic modelling of space," in *Cognitive Systems*, ser. Cognitive Systems Monographs, H. I. Christensen, G.-J. M. Kruijff, and J. L. Wyatt, Eds., 2010, vol. 8, pp. 165–221.

[5] C. Galindo and A. Saffiotti, "Inferring robot goals from violations of semantic knowledge," *Robotics and Autonomous Systems*, vol. 61, no. 10, pp. 1131–1143, 2013.

[6] Ruiz-Sarmiento, J.R., C. Galindo, and J. González-Jiménez, "Exploiting semantic knowledge for robot object recognition," in *Knowledge-Based Systems*, 2015.

[7] Ruiz-Sarmiento, J. R., C. Galindo, and J. González-Jiménez, "Mobile robot object recognition through the synergy of probabilistic graphical models and semantic knowledge," in *European Conf. on Artificial Intelligence. Workshop on Cognitive Robotics*, 2014.

[8] Ruiz-Sarmiento, J.R., C. Galindo, and J. González-Jiménez, "UPGMpp: a Software Library for Contextual Object Recognition," in *3rd. Workshop on Recog. and Action for Scene Understanding*, 2015.

[9] Ruiz-Sarmiento, J. R., C. Galindo, and J. González-Jiménez, "Scene Object Recognition for Mobile Robots through Semantic Knowledge and Probabilistic Graphical Models," 2015, submitted.

[10] J.L. Blanco Claraco, "Mobile Robot Programming Toolkit (MRPT)," http://www.mrpt.org, 2015, [Online; accessed 28-April-2015].

[11] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.

[12] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view rgb-d object dataset," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, May 2011, pp. 1817–1824.

[13] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab, "Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes," in *Proceedings of the 11th Asian Conference on Computer Vision - Volume Part I*, ser. ACCV'12. Berlin, Heidelberg: Springer-Verlag, 2013, pp. 548–562.

[14] A. Singh, J. Sha, K. Narayan, T. Achim, and P. Abbeel, "Bigbird: A large-scale 3d database of object instances," in *IEEE International Conference on Robotics and Automation*, May 2014.

[15] K. Lai, L. Bo, and D. Fox, "Unsupervised feature learning for 3d scene labeling," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, May 2014, pp. 3050–3057.

[16] A. Janoch, S. Karayev, Y. Jia, J. T. Barron, M. Fritz, K. Saenko, and T. Darrell, "A category-level 3-d object dataset: Putting the kinect to work," in *1st Workshop on Consumer Depth Cameras for Computer Vision (ICCV workshop)*, November 2011.

[17] A. Anand, H. S. Koppula, T. Joachims, and A. Saxena, "Contextually guided semantic labeling and search for three-dimensional point clouds," *In The International Journal of Robotics Research*, vol. 32, no. 1, pp. 19–34, Jan. 2013.

[18] N. Silberman and R. Fergus, "Indoor scene segmentation using a structured light sensor," in *Proceedings of the International Conf. on Computer Vision - Workshop on 3D Representation and Recog.*, 2011.

[19] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor Segmentation and Support Inference from RGBD Images," in *Proc. of the 12th European Conference on Computer Vision (ECCV 2012)*, 2012.

[20] A. Aldoma, T. Faulhammer, and M. Vincze, "Automation of "ground truth" annotation for multi-view rgb-d object instance recognition datasets," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, Sept 2014, pp. 5016–5023.

[21] D. Meger and J. J. Little, "The UBC visual robot survey: A benchmark for robot category recognition," in *Experimental Robotics - The 13th International Symposium on Experimental Robotics, ISER 2012, June 18-21, 2012, Québec City, Canada*, 2012, pp. 979–991.

[22] J. Xiao, A. Owens, and A. Torralba, "Sun3d: A database of big spaces reconstructed using sfm and object labels," in *Computer Vision (ICCV), 2013 IEEE International Conference on*, Dec 2013, pp. 1625–1632.

[23] E. Fernandez-Moral, J. González-Jiménez, P. Rives, and V. Arévalo, "Extrinsic calibration of a set of range cameras in 5 seconds without pattern," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014)*, Chicago, USA, September 2014.

[24] R. Gómez-Ojeda, J. Briales, E. Fernández-Moral, and J. González-Jiménez, "Extrinsic calibration of a 2d laser-rangefinder and a camera based on scene corners," in *IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, USA, 2015.

[25] A. Teichman, S. Miller, and S. Thrun, "Unsupervised intrinsic calibration of depth sensors via slam," in *Proceedings of Robotics: Science and Systems*, Berlin, Germany, June 2013.

[26] P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, Feb. 1992.

[27] A. Segal, D. Haehnel, and S. Thrun, "Generalized-icp," in *Proceedings of Robotics: Science and Systems*, Seattle, USA, June 2009.

[28] S. Paris and F. Durand, "A fast approximation of the bilateral filter using a signal processing approach," *Int. J. Comput. Vision*, vol. 81, no. 1, pp. 24–52, Jan. 2009.

[29] Giraff Technologies AB, "Giraff robot," http://www.giraff.org/, 2015, [Online; accessed 06-April-2015].

[30] ASUS, "Xtion PRO LIVE," http://www.asus.com/Multimedia/Xtion_PRO_LIVE/, 2015, [Online; accessed 06-April-2015].

[31] Hokuyo Automatic Co., "Hokuyo URG-04LX-UG01," http://www.hokuyo-aut.jp, 2015, [Online; accessed 06-April-2015].