Project IP-2014-09-3155

Advanced 3D Perception for Mobile Robot Manipulators

SEMI-AUTOMATIC GENRATION OF GROUND TRUTH DATA FOR POINT CLOUD SEGMENTATION

Technical Report

ARP3D.TR4

version 1.0

Emmanuel Karlo Nyarko, Robert Cupec, Ivan Vidović, Ratko Grbić, Michael Coene

Josip Juraj Strossmayer University of Osijek

Faculty of Electrical Engineering Osijek

Osijek, 2016.

Ovaj rad je financirala/sufinancirala Hrvatska zaklada za znanost projektom IP-2014-09-3155.

Mišljenja, nalazi i zaključci ili preporuke navedene u ovom materijalu odnose se na autora i ne odražavaju nužno stajališta Hrvatske zaklade za znanost.

This work has been fully supported by/supported in part by the Croatian Science Foundation under the project number IP-2014-09-3155.

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of Croatian Science Foundation.

1. Motivation

One of the main goals of the project *Advanced 3D Perception for Mobile Robot Manipulators (ARP3D)* is improvement of the existing methods or development of new methods for realization of general mobile manipulation tasks such as: (i) picking of objects, their transport to a desired location and storage in an adequate container; (ii) positioning of a tool in a suitable position relative to an object of interest for the purpose of performing a particular operation. Interpretation of a perceived scene in the form of separate physical objects is a basic ability required by an intelligent agent which manipulates with physical objects. One of the specific objectives of project ARP3D is improvement of the existing methods or development of new approaches for segmentation of point clouds obtained by a RGB-D camera into physical objects. This problem can be more precisely formulated in the following way. Given a point cloud, each point in this cloud should be assigned a label of an object it belongs to.

A reliable and efficient solution to this problem is a precondition for realization of several operations included in mobile manipulation tasks. First, it would enable a user of a mobile robot manipulator to select an object of interest in a given scene by picking a point in the RGB-D image of this scene, e.g. by a mouse click, or by a single touch on a touch screen. Assuming that a segmentation algorithm has successfully assigned a physical object to each point in the RGB-D image, selecting a particular point in this image results in selecting the object this point belongs to. Second benefit of segmenting point clouds into objects is to provide useful information for object recognition. Global recognition approaches (Aldoma et al., 2012; Narayanany and Likhachev, 2016) require points in the observed scene to be segmented into different clusters representing separate objects, so that descriptors can be computed on each object cluster separately. Local recognition method can also benefit from segmentation. Many of these methods are based on feature detection and generation of object hypotheses by matching of scene features to features in a model database. In some cases, matching of a single scene feature to a model feature doesn't provide sufficiently accurate information about the relative pose of an object in the scene. Using two or more feature pairs results in a large number of combinations which must be processed. This number of combinations can be reduced by limiting the selection of feature pairs to separate clusters of points representing separate objects.

A survey of state-of-the-art approaches for segmentation of point clouds into objects is given in (Cupec, Nyarko and Grbić, 2016). The core of a segmentation algorithm is a criterion which decides if two clusters of a given point cloud belong to the same object or to different objects. Some of the existing solutions to this problem use machine learning techniques to learn the parameters defining the segmentation criterion (Richtsfeld et al., 2012; Richtsfeld et al., 2014; Gupta, Arbeláez and Malik, 2013; Karpathy, Miller and Fei-Fei, 2013; Finman et al., 2013). A prerequisite for learning is availability of ground truth data which is used as the reference that guides the learning process. In (Cupec, Nyarko and Grbić, 2016), a survey of existing databases is provided.

Generating ground truth data for point cloud segmentation is a tedious and time consuming task, but it is necessary for application of machine learning techniques. In order to facilitate generation of ground truth data, we developed software for semi-automatic generation of ground truth data, which allows fast creation of large reference datasets.

One approach to segmentation of images into objects is to program a robot to move objects and to segment the image into background and the moved region (Beale, Iravani and Hall, 2011). A similar approach is presented in (van Hoof, Kroemer and Peters, 2014), where a robot autonomously interacts with its environment to segment a scene into objects. Another approach based on a similar idea is to detect changes in the environment and segment the scene according to the difference of point clouds representing the same location in the environment captured in different time instants. The difference between such two point clouds represents an object which moved between the two time instants (Finman et al., 2013). The resulting changes represent discovered objects, which are then used to train multiple segmentation algorithms.

We used the idea of segmentation based on differential depth image to develop software for semi-automatic generation of ground truth data for teaching and testing of RGB-D image segmentation algorithms, which is presented in this report.

2. Semi-Automatic Generation of Ground Truth Data

The approach for generation of ground truth data proposed in this paper is based on differential depth image. Let's consider a scenario where a 3D sensor is mounted in a fixed pose and acquires images of the scene captured in its field of view on user request. After capturing the first image, the user places an object in the sensor's field of view and triggers the next image acquisition. Assuming that the scene captured in the second image differs from the scene captured in the first image only in this single object placed on the scene between the two image acquisitions. The depths of the points of the inserted object in the second image are smaller than the depths of the corresponding points in the first image. The depths of all other points remain the same. This depth difference can be used to segment the inserted object from the rest of the scene. This procedure can be repeated by placing other objects in the scene one by one, acquiring an image after each object insertion, computing the depth difference between two consecutive images and annotating the inserted object in the latest image according to the depth difference. The proposed procedure is represented by Algorithm 1.

The result of Algorithm 1 is matrix **S** representing the segmented scene. Each element s_{uv} of this matrix corresponds to the pixel (u, v) of the last depth image. The value s_{uv} is the number of the object placed in the scene, i.e. $s_{uv} = k$ if pixel (u, v) represents the k-th object in the last image. In most cases, some points of a depth image acquired by a 3D sensor have undefined depth value due to reflection or absorption of the sensor beam as well as to a limited sensor range. If pixel (u, v) has undefined depth value for any of the acquired depth images, $s_{uv} = 0$.

Threshold τ_{GT} used in the condition for assigning a pixel to the newly inserted object (line 16) is determined based on the measurement noise of the used sensor. If τ_{GT} is set to a high value, many points of the newly inserted object are not assigned to this object because the depth difference doesn't exceed τ_{GT}. Consequently, some relatively small objects can be completely blended with the background. On the other hand, if τ_{GT} is below the noise level, many scene points which don't belong to the new objects are annotated as belonging to this object. In order to reduce this problem, the ground truth generating software developed in ARP3D project performs filtering of the acquired depth images by averaging over a sequence of 15 images. Further reduction of the noise effects is achieved by segmenting the depth image into clusters of pixels and performing the assignment to

the new object on the cluster level. If a cluster contains at least α_{GT} of pixels (u, v) which satisfy condition

$$d_{uv} < \tau_{GT},$$

then the entire cluster is assigned to the new object. Otherwise, none of the points belonging to this cluster is assigned to the new object.

Algorithm 1 *Differential Image Segmentation*

Input: τ_{GT}
Output: **S**
$k \leftarrow 1$
D(k) ← depth image acquired by a 3D sensor
S ← matrix of same dimensions as **D**(k) with all elements equal to 0
For all pixels (u, v) of **D**(k)
    If $d_{uv}(k)$ is defined **then**
        $s_{uv} \leftarrow k$
    end if
end for
Repeat
    $k \leftarrow k + 1$
    Place a new object in the scene.
    D(k) ← depth image acquired by the 3D sensor
    For all pixels (u, v) of **D**(k)
        If $d_{uv}(k)$ is defined **and** $d_{uv}(k-1)$ is defined **then**
            $\Delta d_{uv}(k) \leftarrow d_{uv}(k) - d_{uv}(k-1)$
            If $\Delta d_{uv}(k) < \tau_{GT}$ **then**
                $s_{uv} \leftarrow k$
            else
                $s_{uv} \leftarrow 0$
            end if
        else
            $s_{uv} \leftarrow 0$
        end if
    end for
until stopped by the user

Two methods for grouping pixles into clusters are used as options in the current implementation of our ground truth generation software: Voxel Cloud Connectivity Segmentation proposed in (Papon et al., 2013) and segmentation to planar surface segments described in (Cupec and Vidović, 2016). The latter is used to create the dataset presented in Section 3.

In order to constrain the ground truth generation process to a particular volume inside the camera field of view, which can be easily controlled, the ground truth generation software is designed to consider only the points whose depth is smaller or equal to a user specified value d_{max}.

The software for semi-automatic generation of ground truth data developed in the project ARP3D has two modes of operation:

1. Image acquisition and
2. Ground truth generation and editing.

Image acquisition. In this mode, the user takes snapshots of a given scene using a RGB-D camera connected to a personal computer. After the program is started, the depth image and RGB image

stream are displayed in the computer screen. Image acquisition is triggered by pressing the space key. The current depth image, RGB image and filtered image, i.e. average depth image over a sequence of 15 images, are stored in separate files. An example of RGB image, depth image and filtered image are shown in Fig. 1. The depth image is transformed in a point cloud represented in the format of the Point Cloud Library (Rusu and Cousins, 2011) and stored in a file. Furthermore, a triangular mesh is created from this point cloud and stored in a file in standard PLY format. The triangular mesh created from the RGB and depth image in Fig. 1 is shown in Fig. 2. After an image acquisition is completed, the user places a new object in the scene and triggers another image acquisition.

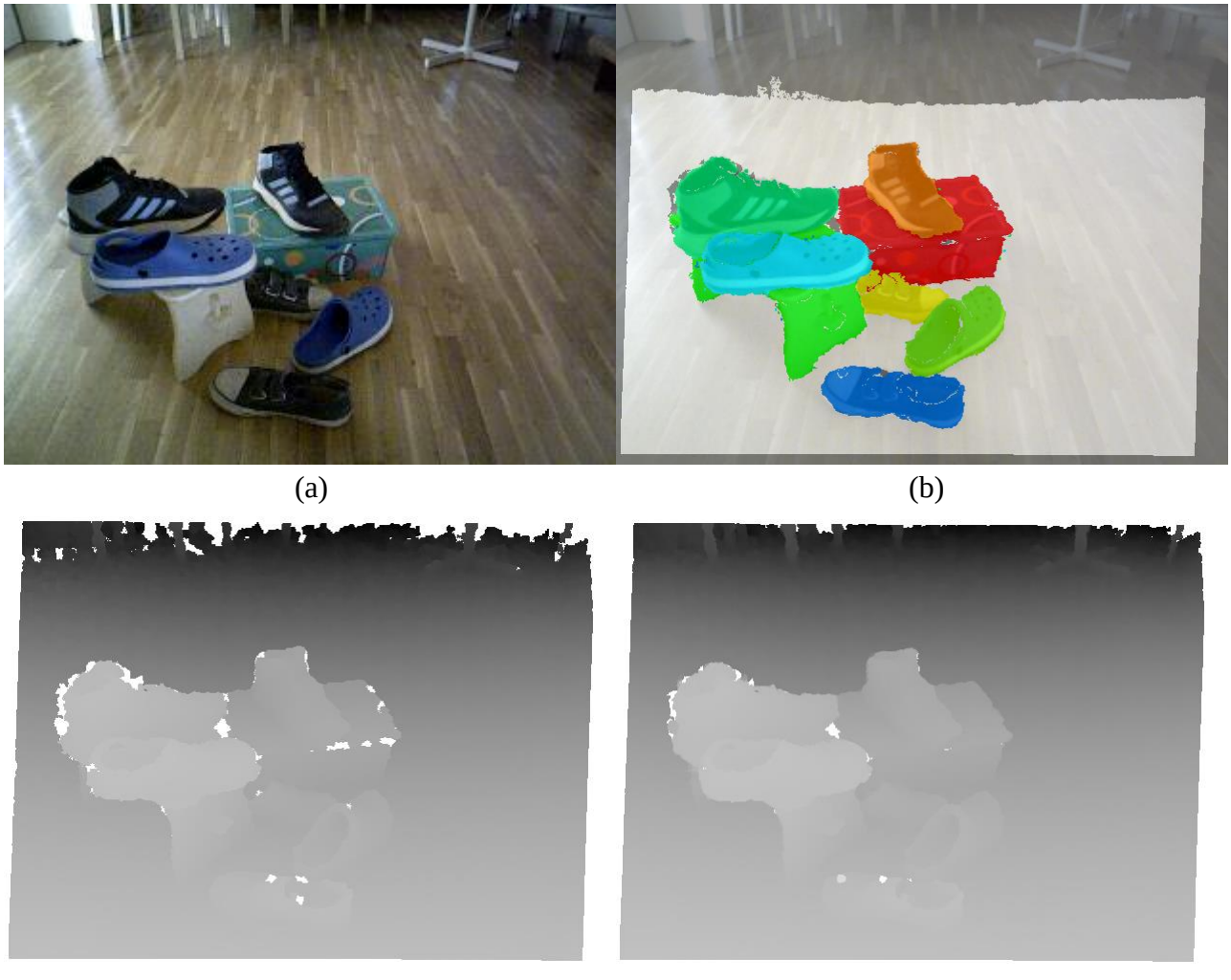

Fig. 1. (a) RGB image of a scene; (b) the scene segmented into objects, where each object is denoted by different color; (c) depth image; (d) filtered depth image.

Ground truth generation and editing. When using the discussed software in this mode, the user specifies the location of files containing the first image in a sequence obtained by the described image acquisition process. After the program is started, the first RGB image shown in grayscale is displayed on the computer screen, where points with defined depth smaller or equal to d_{max} are shown with brighter intensity, while the other points, which are not considered for ground truth, are darker. These bright points represent object 0 or background. After pressing N-key, the next recorded image is displayed, where, in addition to the background, the first object placed in the scene is denoted by a distinguishing color. After each pressing of N-key, the image with the next object placed in the scene is displayed. Each object is represented with different color. Hence, each

image represents a segmentation of the scene to separate physical objects. An example of the image segmented into objects is shown in Fig. 1(b).

Fig. 2. Triangular mesh created from RGB and depth image shown in Fig. 1 viewed from four different viewing angles.

The software allows the user to correct the segmentation results if needed. If a cluster, which should belong to the new object, is assigned to a different object or the background, the user can reassign this cluster by a right-mouse-click on the new object followed by a left-mouse-click on the falsely assigned cluster. Since the a new object placed on the scene in front of other objects projects to the camera image as a connected set of pixels, it is reasonable to reject all small connected sets in the image which are falsely recognized as parts of the new object. By pressing C-key the connected component analysis is performed on the set of points which are assigned to the new object and only the connected components containing at least n_{cc} number of points are preserved, while smaller connected components are rejected.

The parameters of the ground truth generation algorithm are specified in a configuration file. The relevant parameters, their corresponding items in the configuration file, their mathematical symbol and values used in the experiments are listed in Table 1.

Table 1. Parameters of the ground truth generation software.

Description	Item in the configuration file	Mathematical symbol	Values used in experiments
threshold which defines significant change in depth image	GT.DifferenceThreshold	τ_{GT}	5, 10, 20 mm
minimum percentage of pixels with changed depth within a new object cluster	GT.PercThreshold	α_{GT}	80 %
maximum depth of scene points considered for ground truth data	GT.MaxDist	d_{max}	1.4, 2.0, 3.0 m
minimum allowed size (number of points) of the connected component which is considered to represent an object	GT.minConnectedComponentSize	n_{cc}	1000

3. Dataset

There are number of benchmark datasets for training and testing of segmentation algorithms for RGB-D images and 3D point clouds. Most of them contain scenes with household objects, such as bottles, cups, glasses, bowls, boxes, cans, toys etc. In order to make an original contribution to the existing volume of segmentation datasets, we created a dataset which, in addition to the aforementioned 'standard' object types, contains scenes which significantly differ from the currently available datasets. Our dataset includes scenes with fruits, vegetables, pieces of wood, stones, construction blocks and gardening tools. The dataset, created using the software described in Section 2, consists of 424 images representing indoor and outdoor scenes. The scenes contain between 1 and 10 objects. Many of these images represent scenes where objects are occluded by other objects. Some examples of the scenes from the database and the corresponding ground truth segmentations are shown in Fig. 3. The dataset is made publically available on the web page https://www.etfos.hr/projects/hrzz-projects/advanced-3d-perception-for-mobile-robot-manipulators/research.

References

Aldoma A, Tombari F, Rusu RB, and Vincze M (2012) OUR-CVFH – Oriented, Unique and Repeatable Clustered Viewpoint Feature Histogram for Object Recognition and 6DOF Pose Estimation, *Joint DAGM-OAGM Pattern Recognition Symposium*.

Beale D, Iravani P and Hall P (2011) Probabilistic Models for Robot-Based Object Segmentation, *Robotics and Autonomous Systems*, vol. 59, issue 12, pp. 1080–1089.

Cupec R and Vidović I (2016) Features for Object Recognition in 3D Point Clouds Based on Planar Patches, *Technical Report ARP3D.TR5*

Cupec R, Nyarko EK and Grbić R (2016) Survey of State-of-the-Art Point Cloud Segmentation Methods, *Technical Report ARP3D.TR1*

Finman R, Whelan T, Kaess M and Leonard JJ (2013) Toward lifelong object segmentation from change detection in dense rgb-d maps, *IEEE European Conference on Mobile Robots (ECMR)*, pp. 178–185.

Gupta S, Arbeláez P, and Malik J (2013) Perceptual Organization and Recognition of Indoor Scenes from RGB-D Images, *Computer Vision and Pattern Recognition (CVPR)*

van Hoof H, Kroemer O and Peters J (2014) Probabilistic Segmentation and Targeted Exploration of Objects in Cluttered Environments, *IEEE Transactions on Robotics*, vol. 30, no. 5, pp. 1198–1209.

Karpathy A, Miller S and Fei-Fei L (2013) Object discovery in 3d scenes via shape analysis, *IEEE International Conference on Robotics and Automation (ICRA)*

Narayanany V and Likhachev M (2016) PERCH: Perception via Search for Multi-Object Recognition and Localization, IEEE International Conference on Robotics and Automation (ICRA)

Papon J, Abramov A, Schoeler M and Wörgötter F (2013) Voxel Cloud Connectivity Segmentation - Supervoxels for Point Clouds, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2027–2034.

Richtsfeld A, Mörwald T, Prankl J, Zillich M, and Vincze M. (2014) Learning of perceptual grouping for object segmentation on rgb-d data. *Journal of Visual Communication and Image Representation*. vol. 25, pp. 64–73.

Richtsfeld A, Mörwald T, Prankl J, Zillich M, and Vincze M (2012) Segmentation of unknown objects in indoor environments, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4791–4796.

Rusu RB and Cousins S (2011) 3D is here: Point Cloud Library (PCL), IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China

Fig. 3. Sample RGB images (first and third column) and the corresponding ground truth segmentations (second and fourth column).