Interactive 3D Point Cloud Segmentation with HoloLens 2

Changan Chen, Matias Turkulainen, Nora Eirich
ETH Zürich
Rämistrasse 101, Zürich, Switzerland
chencha, mturkulainen, eirichn@student.ethz.ch

Abstract

3D semantic segmentation is important for various applications in general scene understanding tasks. However, annotating ground truth datasets is a time-consuming and costly process. We propose an auto-labeling tool using the Microsoft HoloLens 2 for interactive egocentric object segmentation. Recent deep learning techniques for volumetric 3D segmentation of point clouds have shown the effectiveness of using machine learning to guide and simplify human annotation. However, these have been confined to using 2D graphical interfaces and point-and-click methods. We propose an interactive 3D segmentation method where the user directly interacts with the environment point cloud in a mixed-reality setting and annotates objects with the help of a state-of-the-art volumetric segmentation network. We implement an intuitive user interface on the HoloLens 2 and show how user-guided segmentation can be achieved to greatly reduce the time required for volumetric segmentation.

1. Introduction

Deep learning has been effectively applied to problems in 2D and 3D semantic segmentation [6, 7]. Semantic segmentation of scenes has been considered a critical problem for computer vision systems to achieve a human-like understanding of their environments. While the principle behind general human scene understanding is still unsolved [4], great leaps have been made to mimic human abilities with artificial intelligence. However, most systems are trained on supervised datasets that rely on well-labeled datasets. Annotation for 2D and especially 3D segmentation is a challenging and time-consuming task. If done manually, each pixel or voxel in a scene has to be labeled; therefore, there is a demand for algorithms capable of auto-labeling. Several methods exist for 2D segmentation [5, 10] and recently methods for 3D segmentation have also been proposed [8, 9]. Despite the significant progress that has been made, there is still no intuitive method to capture and simultaneously label 3D point clouds. This is the motivation for this paper.

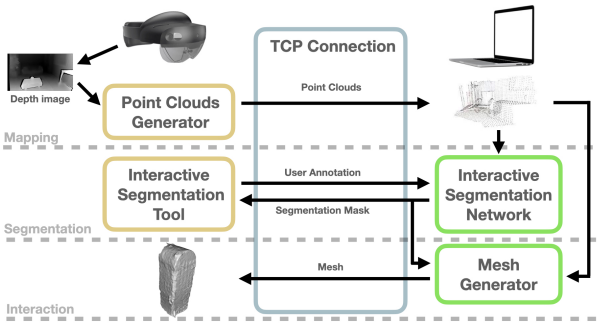


Figure 1. System overview.

The work by Kontogianni et al. [8] introduced an auto-labeling framework for 3D point clouds guided by human annotation as well as by a semantic segmentation network based on the Minkowski engine [3] in a 3D graphical user interface. In this work, we expand the graphical user interface to the mixed reality setting using Hololens 2. The HoloLens 2 headset is used to label 3D point clouds with sparse user annotations from egocentric views. This interactive first-person semantic segmentation tool significantly improves the time and effort required for labeling tasks in custom environments and enables faster workflows for other research endeavors. Furthermore, it allows for real-time verification and correction of segmentations with minimal user annotation. We implement a tool capable of visualizing point clouds, segmentation results, and iterative refinement of segmentations, as well as a 3D mesh visualizer of the final segmentation. Our contributions can be summarized as follows:

- An interactive 3D point cloud segmentation and refinement tool from egocentric views on the HoloLens 2.

2. Method

2.1. System Overview

The system architecture is divided into the following modules as shown in Fig. 1

1. A point cloud generator Sec. 2.2 generating a point cloud representation of the environment;

2. An interactive segmentation tool Sec. 2.3 on the HoloLens 2 used for iterative labeling and correction of segmented objects;

3. A pre-trained segmentation network Sec. 2.4 running on a remote PC used to create segmentation masks from user labels;

4. A mesh generator Sec. 2.5 producing a mesh for an object from the point clouds that belong to the object;

5. A TCP client-server connection Sec. 2.6 connecting the HoloLens 2 and the remote PC for data transfer of point clouds, user corrections, and meshes to-and-from the user.

The application pipeline is divided into three stages according to their purpose as shown in Fig. 1:

1. Mapping;

2. Segmentation;

3. Interaction.

In the mapping stage, the user builds a point cloud representation of the environment wearing the HoloLens2. The environment point cloud is then transferred to a remote computer via a TCP connection before the user enters the segmentation stage. In the segmentation stage, the user provides annotation labels by pointing to the point clouds with the hand ray and using customized gestures. The user annotation is sent to the computer. An interactive segmentation network takes the user corrections and the environment point clouds as inputs, and outputs a refined segmentation mask, which is transferred back to the HoloLens 2. The HoloLens 2 displays the updated segmentation results to the user so the user can provide further corrections to refine the segmentation. This correction process loops until the user is satisfied with the results and terminates the segmentation stage with another customized gesture. Finally, in the interaction stage, we generate a mesh of the segmented object using the segmented pointcloud on the remote computer and send it to the HoloLens2, allowing the user to interact with it.

(a) A backpack on a table (b) Point clouds representation

Figure 2. Point clouds of the environment built with HoloLens 2. (a) An environment of a backpack on the table. (b) The point cloud representation of the environment overlayed on the real scene and displayed to the user on the HoloLens 2.

2.2. Point Clouds Generation

To generate the point cloud, we use an open-source HoloLens 2 research mode Unity plugin [1]. The plugin builds the point cloud from the long-throw or short-throw depth camera. The short-throw depth camera is mainly used to recognize the user's hands. Since we only desire to map the environment and do not want artifacts produced by the movement of the hands of the user, we opt to build the point cloud from only the long-throw depth camera. Using one frame of point cloud is not enough to map a large scene due to the limited field of view of the long-throw depth camera of HoloLens 2. Therefore we sparsify the point cloud and accumulate them every n_f frames to map a larger scene while preventing the point clouds from becoming too dense. We simply concatenate the point clouds generated from different frames without performing any alignment since we aim assume a static environment, while the point clouds are represented in world coordinates, and the localization of HoloLens 2 itself is already accurate enough. The user is then able to build a complete point cloud representation of the object, as shown in Fig. 2, by moving around the object to provide different perspectives.

2.3. Interactive Segmentation Tool

Once the point clouds of the environment are built and sent to the remote computer, the user starts to provide the segmentation network (Sec. 2.4) with human annotations as input. The human annotations include pointing out the part that is wrongly segmented, i.e., the point clouds should be a part of the object but segmented as background, or the point clouds belong to the background but are segmented as a part of the object. We refer to the former as a positive correction and the latter as a negative correction.

To realize the correction, we need a 3D cursor that can be easily placed on any part of the point clouds. Using the spatial-awareness of the HoloLens2 and the hand ray API, we can obtain the 3D point on the surface of the environment that the user is pointing to, as shown in Fig. 3. Since

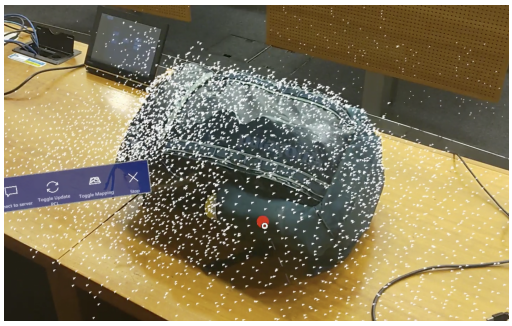


Figure 3. 3D cursor. The 3D cursor, marked as a red sphere, is the hit point of the hand ray and the environment mesh built by the spatial-awareness of the HoloLens 2. It can be seen that the cursor can be placed on the environment surface.

the point clouds are immediately on the surface, this cursor can be used to point at any part of the point clouds, provided that the environment mesh is built accurately enough by the spatial-awareness functionality.

To differentiate the different types of user annotations and actions, we define the following three simple gestures.

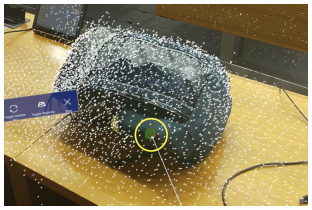
- **Positive correction**: a pinch with the thumb and the index finger;
- **Negative correction**: a pinch with the index finger and the middle finger;
- **End of segmentation**: a pinch with the thumb and the middle finger.

With the gestures defined, the process of interactive segmentation is summarized in the following algorithm.

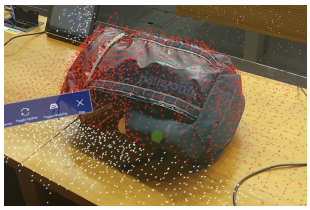
while *!segmentation finished* **do**
  if *positive correction detected* **then**
    Get the hit point of the hand ray;
    Send it to the PC as a positive correction;
  else if *negative correction detected* **then**
    Get the hit point of the hand ray;
    Send it to the PC as a negative correction;
  else if *segmentation end detected* **then**
    Send finish signal to the PC;
    segmentation finished ← True
  end
end

Algorithm 1: *Interactive Segmentation*

The algorithm is an iterative process of refining the segmentation result. A single iteration for both giving a positive correction and a negative correction is illustrated in Figs. 4a and 4b, and Figs. 5a and 5b, respectively.

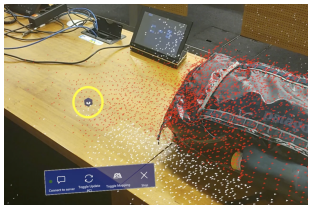


(a) The positive correction.

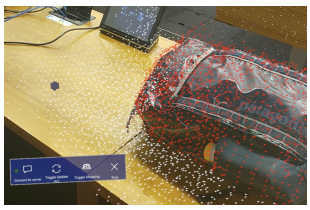


(b) After the positive correction.

Figure 4. User giving a positive correction at the beginning of the process. The point clouds segmented as the object are colored in red, whereas the ones segmented as the background are in white. (a)At first, all points are labeled as background. The user places the cursor on the bag and gives a positive correction, which is displayed as a green cube on the HoloLens 2 and highlighted with a yellow circle. (b) After the correction, the updated segmentation mask correctly includes the part the user indicates to the object, and corresponding point clouds are shown in red.



(a) The negative correction.



(b) After the negative correction.

Figure 5. User giving a negative correction to refine the segmentation mask. (a) As it can be seen that some point clouds on the table are in red, the segmentation mask wrongly labels the table, i.e. the background, as the object. The user provides a negative correction, marked with a blue cube on the HoloLens 2 and highlighted in a yellow circle. (b) After the correction, the segmentation mask successfully adapts to the user annotation and labels the previously incorrectly labeled points as background. Those points are now in white.

2.4. Interactive Segmentation Network

The input to the segmentation network consists of the 3D scene $P \in R^{N \times C}$ where N is the number of points in the point cloud and C the number of input features. For our segmentation task, C is five-dimensional corresponding to xyz coordinates for each point and two additional channels for binary labels T_p and T_n for the positive and negative clicks, respectively. The positive clicks are considered on the object, and the negative clicks are on the background. For each positive and negative click, a small bounding box volume is created, expanding the region of influence for each user annotation. The bounding cube edge length is set to 0.05m after ablation studies.

The network architecture is an adaptation of the Minkowski Engine [3] as described in the implementation paper [8]. We use the pre-trained network from [8] trained on a dataset of indoor scenes, which was deemed successful

in indoor environments obtained with the HoloLens 2.

2.5. Mesh Generation

After the user is satisfied with the object mask from the network, the points making up the object are used to create a 3D object. The python library Open3D [11] was used to create the mesh for the 3D reconstruction of the point cloud. For the ball pivoting algorithm [2], the radii used are based on the average distance between the nearest neighbored points. We then transfer the created mesh via the TCP connection from the server to the HoloLens 2. Since not the whole object file can be sent at once, two lists are extracted from it. One is the list of vertices, previously the points of the point cloud, and the other is the list of triangle vertices. Once the two lists have been transferred to the HoloLens 2, they are transformed into the proper format and assigned to an empty mesh object. That object can then be seen and used within the HoloLens 2 environment.

2.6. TCP connection

The TCP client-server connection is built with the Socket API. The TCP connection is used to transfer the environment point cloud, user-labeled coordinates, segmentation masks, as well as the final mesh to-and-from the Hololens 2. Data is transferred in chunks. Point clouds and user annotation xyz coordinates are transferred and distinguished with unique data buffer headers corresponding to the environment point cloud, positive label, and negative label respectively. The final mesh is transferred as a list of xyz coordinates of the vertices and correspondences of the vertices to form triangles.

3. Results

We test the proposed interactive segmentation tool on different objects and present the qualitative analysis of the results in terms of the quality of the generated segmentation mask and the produced mesh.

3.1. Segmentation

We use the proposed tool to segment six different objects. A table, a sign, a trash can, a desk, a backpack, and a kettle. The segmentation results are shown in Figs. 6g to 6l. The average number of user corrections required to generate the shown results is 4 to 5. It can be seen that the segmentation of the table, the sign, the trash can, the desk, and the backpack are successful using only a few corrections. The segmentation masks are nearly the ground truth, which would otherwise require a few hundred of annotations to generate when manually labeling every point or many iterations of troublesome resizing and placing a labeling bounding box. The effectiveness of the proposed segmentation tool is shown.

The only problematic result is the kettle, as shown in Fig. 6l. However, this is due to the incompleteness of the point clouds. Since the point clouds are generated from depth images, there is certain noise, and for the textureless patches, no points are generated. As shown in Fig. 6l, we fail to map the black upper part of the kettle with point clouds due to its texturelessness, which results in an unrecognizable segmentation mask. Nevertheless, the semantic segmentation itself works fine, given that most points are labeled correctly as either the object or the background.

3.2. Mesh Generation

While the generation and transfer of the mesh are fast, there are some aspects limiting the quality of the generated mesh. We noticed that because the segmented point cloud can be irregularly spaced, the Ball Pivoting Algorithm generally generates meshes with some holes. Another factor influencing the mesh is the sparsity of the segmented points. It is noticeable that the smaller the object, the fewer points it is made of, and hence the mesh becomes less accurate. The sparsity also limits the detail in the generated mesh.

3.3. Interaction

The interaction with the mesh as well as its visualization on HoloLens 2 is not properly achieved due to some technical problems. The issue can be tracked down and tackled given more time for debugging. However, it should be noted that the emphasis of this work is on the segmentation tool rather than the interaction with a mesh using HoloLens 2.

4. Limitation

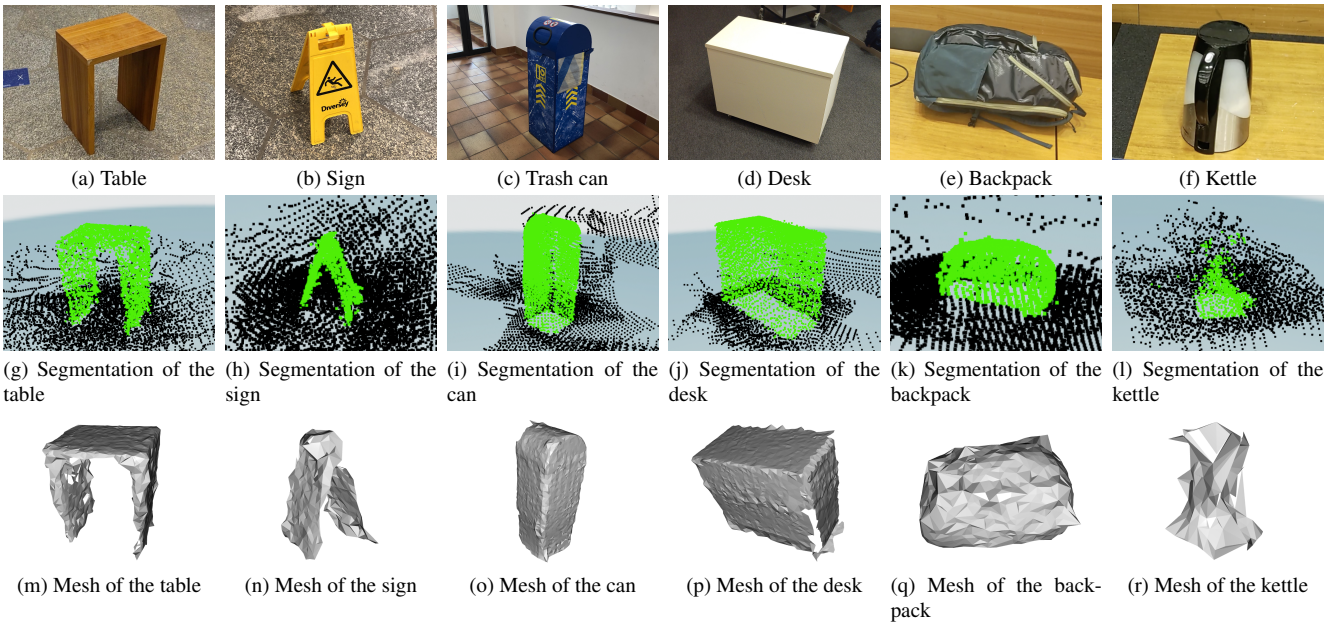
The limitation is mainly the quality of the point clouds, which are reconstructed from the depth image. In the textureless region, no points can be constructed and the reconstructed point clouds do not conform to the actual object. Furthermore, the so obtained point clouds are noisy, which results in the failure of mapping and segmenting the object as well as mesh generation.

5. Conclusion

In this work, we propose an interactive 3D point cloud labeling tool with HoloLens 2 with the help of a state-of-the-art point cloud segmentation neural network. We show the effectiveness of the proposed method with experiments on six different objects. A close-to-ground-truth segmentation mask can be generated in 4-5 clicks for each object.

There are several avenues to further improve the proposed HoloLens 2 interactive segmentation tool.

The current architecture only uses the xyz position in the world coordinates frame as an input to the segmentation network; however, support for color RGB information for each point is also possible. The pre-trained network [8] was



(a) Table (b) Sign (c) Trash can (d) Desk (e) Backpack (f) Kettle

(g) Segmentation of the table (h) Segmentation of the sign (i) Segmentation of the can (j) Segmentation of the desk (k) Segmentation of the backpack (l) Segmentation of the kettle

(m) Mesh of the table (n) Mesh of the sign (o) Mesh of the can (p) Mesh of the desk (q) Mesh of the backpack (r) Mesh of the kettle

Figure 6. Segmentation results and generated meshes

trained on a collection of datasets with xyz only and xyz + RGB information. We hypothesize that including color information could improve the segmentation results. This could be achieved by extracting the color information of the pixels of the projected points on the camera of the HoloLens 2 image.

Furthermore, the network architecture currently returns a mask for the binary classification of a single object. Expanding the output of the network for multi-object classification will make the system more appealing for automatic semantic segmentation dataset generation. In addition, if the tool is intended for commercial or more general use, additional support with offline refinement, storing, and visualization tools would be required.

References

[1] https://github.com/petergu684/HoloLens2-ResearchMode-Unity.git. 2

[2] Fausto Bernardini, Joshua Mittleman, Holly Rushmeier, Claudio T. Silva, and Gabriel Taubin. The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 5(4):349–359, 1999. 4

[3] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page 3070–3079, 2019. 1, 3

[4] Venkata Satya Sai Ajay Daliparthi. The ikshana hypothesis of human scene understanding, 2021. 1

[5] Hai Gao, Wan-Chi Siu, and Chao-Huan Hou. Improved techniques for automatic image segmentation. *Circuits and Systems for Video Technology, IEEE Transactions on*, 11:1273 – 1280, 2002. 1

[6] Yanming Guo, Yu Liu, Theodoros Georgiou, and Michael S. Lew. A review of semantic segmentation using deep neural networks. *International Journal of Multimedia Information Retrieval*, 7(2):87–93, 2018. 1

[7] Yong He, Hongshan Yu, Xiaoyan Liu, Zhengen Yang, Wei Sun, Yaonan Wang, Qiang Fu, Yanmei Zou, and Ajmal Mian. Deep learning based 3d segmentation: A survey, 2021. 1

[8] Theodora Kontogianni, Ekin Celikkan, Siyu Tang, and Konrad Schindler. Interactive object segmentation in 3d point clouds. 2022. 1, 3, 4

[9] Yuxing Xie, Jiaojiao Tian, and Xiao Xiang Zhu. Linking points with labels in 3d: A review of point cloud semantic segmentation. *IEEE Geoscience and Remote Sensing Magazine*, 8(4):38–59, 2020. 1

[10] Xiang Zhang, Wei Zhang, Jinye Peng, and Jianping Fan. Automatic image labelling at pixel level, 2020. 1

[11] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018. 4