3D Reconstruction and Object Detection for HoloLens

Zequn Wu[1,2], Tianhao Zhao[1,2]
[1]*College of Engineering and Computer Science*
Australian National University
Canberra, Australia
{Zequn.Wu, Tianhao.Zhao}@anu.edu.au

Chuong Nguyen[2]
[2]*Cyber Physical Systems - Imaging and Computer Vision*
CSIRO Data61
Canberra, Australia
chuong.nguyen@csiro.au

***Abstract*—Current smart glasses such as HoloLens excel at positioning within the physical environment, however object and task recognition are still relatively primitive. We aim to expand the available benefits of MR/AR systems by using semantic object recognition and 3D reconstruction. Particularly in this preliminary study, we successfully use a HoloLens to build 3D maps, recognise and count objects in a working environment. This is achieved by offloading these computationally expensive tasks to a remote GPU server. To further achieve realtime feedback and parallelise tasks, object detection is performed on 2D images and mapped to 3D reconstructed space. Fusion of multiple views of 2D detection is additionally performed to refine 3D object bounding boxes and separate nearby objects.**

***Index Terms*—HoloLens, Mixed Reality, 3D Reconstruction, Object Detection**

I. Introduction

Mixed and Augmented Reality can greatly extend a user capabilities and experiences by bringing digital data directly into the physical world where and when it is most needed. Current systems excel at positioning within the physical environment, however object and task recognition is still relatively primitive. With an additional semantic understanding of the wearer's physical context, intelligent digital agents can assist workers in warehouses, factories, greenhouses, etc. or guide consumers through completion of physical tasks.

Several studies have been done in this area. YOLOv2 deep network [1], [2] was applied to HoloLens in client-server configuration to perform 2D object detection from color video stream. Realtime 3D object detection and pose estimation [3] were applied to a RealSense camera attached to a HoloLens and provides pose to the Hololens for 3D visualisation. Hololens was also used in [5] to label 3D point cloud to train a deep network to locate robot in 3D environment.

In this work, we aim to build a real-time application using a HoloLens to scan the indoor environments and then building up 3D interactive scans. Particularly, our work focuses on 3D scene reconstruction of a small room from the recorded HoloLens v1's depth and color information. We also conduct 3D object detection and tracking for objects in an indoor environment.

II. 3D Scene Reconstruction and Object Detection for HoloLens

A. 3D Scene Reconstruction

A HoloLens v1 captures depth and infrared reflectivity by its time-of-flight (ToF) camera and color by its RGB camera. For ToF depth capture, HoloLens provides long-throw depth and short throw depth. Long throw depth captures objects between 1 and 4 meters at 1 FPS while short throw captures between 0.02 and 3 meters at 15 FPS. In the following section, we use the long throw depth unless otherwise stated.

1) Point Cloud Reconstruction: Apart from the depth estimation of each frame, HoloLens also provides transformation matrices to calculate camera poses while recording. The provided transformation matrices for each camera include frame-to-origin (of HoloLens' world coordinate system) matrix M_{f2o}, camera-view matrix M_{cvt} (extrinsics) and camera-projection matrix M_{cpt} (intrinsics & extrinsics). The point cloud in world coordinate system P_w is obtained from 3D point transformation from depth coordinate to world coordinate:

$$P^w = M^d_{f2o}(M^d_{cvt})^{-1}P^d \tag{1}$$

where P^w and P^d is the homogeneous coordinates of 3D points in the world and depth-camera coordinate systems, M^d_{f2o} and M^d_{cvt} are the matrices for depth camera.

2) Point Cloud Colorization: Due to the limitation of gray scale images from HoloLens reflectivity data, the reconstructed point cloud lacks color information. HoloLens also provides RGB images of smaller field of view, therefore we can colorize the point cloud with relative poses between depth camera and RGB camera. The transformation from depth coordinate to color frame coordinate is:

$$P^c = ((M^d_{cvt})^{\mathbf{T}})^{\mathbf{-1}}(M^d_{f2o})^{\mathbf{T}}((M^c_{f2o})^{\mathbf{T}})^{\mathbf{-1}}(M^c_{cvt})^{\mathbf{T}}(M^c_{cpt})^{\mathbf{T}}P^d \tag{2}$$

where P_c is the homogeneous coordinates in the color image to obtain the color for P^d; M^c_{f2o} and M^c_{cvt} are the matrices for color camera.

Considering that the appearance in different photos slightly changes from different angles, we average RGB value of the 3D points over multiple color views. Due to smaller field of

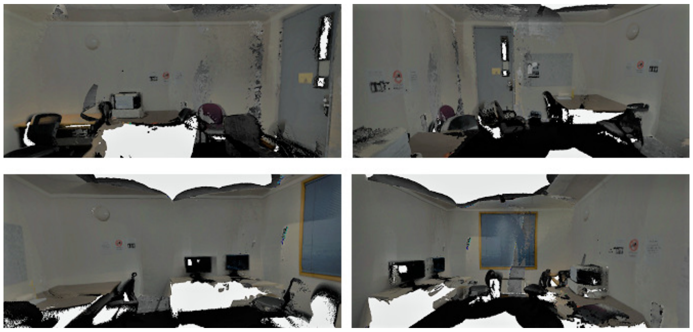

Fig. 1. Different perspectives of the point cloud of TV room. Blank areas are due to lack of color views and white areas due to incomplete coverage.

view of color frames, not all 3D points have color information as shown in Figure 1.

Algorithm 1 Update 3D bounding boxes with IoU

```
Input: Point clouds $\{P_i | i = 1, 2, \cdots, N\}$
Output: Updated 3D bounding boxes $\{B_j | j = 1, 2, \cdots, K\}$
Initialise first Point cloud $P = P_1$
for i = $2, 3, \cdots, N$ do
    Compute $IoU_{3D}$ between $P$ and $P_i$
    if $IoU_{3D} > threshold$ then
        Fuse two boxes $P \leftarrow P + P_i$
        Remove outliers too far from the median of $P$
    else
        Record 3D bounding box $B_j$ from $P$
        Initialise $P \leftarrow P_i$
    end if
end for
```

B. Object Detection and Tracking

In parallel to the 3D point cloud reconstruction, object detection is performed on reflectivity and color frames. 2D bounding boxes can be obtained from detected objects and matched with their responding point cloud coordinates in 3D environment. However, the 2D bounding boxes contain pixels of the object, and some background pixels as well. As a result, we use segmented masks to reduce point clouds from background and fuse the 3D bounding boxes of the point clouds from different views of the same object. Here we apply Mask R-CNN [4] on individual color frames to obtain object masks, and then find their corresponding 3D point cloud.

Another challenge is how to define the border of different objects of the same category and locate closely. Our solution is to fuse 3D bounding boxes through computing the IoU of two neighbouring boxes. As an extension of 2D IoU, here we calculate 3D IoU as $IoU_{3D} = \frac{V_1 \bigcap V_2}{V_1 + V2 - V_1 \bigcap V_2}$. If the 3D IoU is larger than a threshold, the two detected point clouds should belong to the same object and be concatenated. We update the median 3D point and its l_0 distances to other 3D point. To remove points in the background, we select and drop a percentage number of points of the largest distance. The procedure is demonstrated in Algorithm 1.

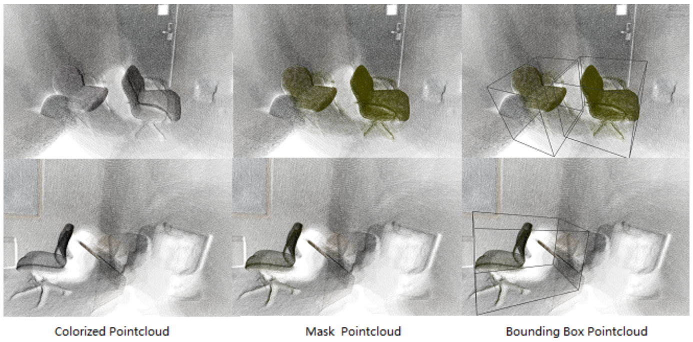

Fig. 2. Two corners of our room model, each contains some masked chairs with refined bounding boxes.

III. Experiments and Results

We choose a TV room and record a 2-minute video using HoloLens v1. Point clouds are reconstructed and colorized using laptop with 2.2 GHz 6-Core Intel Core i7 CPU. We record the video using a HoloLensForCV [6]. Object detection and tracking are performed using Mask RCNN [4] on the Tesla P100 GPU. Figure 1 shows the 3D reconstruction of TV room from four perspectives. Figure 2 demonstrated object detection and bounding box refinement. It shows that the proposed approach successfully tracks individual chairs and extract good 3D bounding boxes from multiple 2D segmentations.

IV. Conclusion

We have presented a workflow from recording data to 3D reconstruction to object detection using the HoloLens. We have demonstrated that 2D object segmentation can be used to obtain 3D segmentation and bounding boxes to allow for realtime object detection and counting. Fusion of 3D bounding boxes improves the separation between nearby objects and refine the bounding box sizes. Future works include feedback loop for realtime object detection visualisation, and question and answer capability to query information of the environment.

References

[1] Haythem Bahri, David Krčmařík, and Jan Kočí. Accurate object detection system on hololens using yolo algorithm. In *2019 International Conference on Control, Artificial Intelligence, Robotics & Optimization (ICCAIRO)*, pages 219–224. IEEE, 2019.

[2] A Corneli, B Naticchia, A Carbonari, and F Bosché. Augmented reality and deep learning towards the management of secondary building assets. In *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*, volume 36, pages 332–339. IAARC Publications, 2019.

[3] Mathieu Garon, Pierre-Olivier Boulet, Jean-Philippe Doiron, Luc Beaulieu, and Jean-François Lalonde. Real-time high resolution 3d data on the hololens. In *2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct)*, pages 189–191. IEEE, 2016.

[4] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[5] Linh Kästner, Vlad Catalin Frasineanu, and Jens Lambrecht. A 3d-deep-learning-based augmented reality calibration method for robotic environments using depth sensor data. *arXiv preprint arXiv:1912.12101*, 2019.

[6] Microsoft. Hololensforcv. https://github.com/microsoft/HoloLensForCV, August 2020.