2M1-5

Fully Convolutional Object Depth Prediction for 3D Segmentation from 2.5D Input

Kentaro Wada, Kei Okada, Masayuki Inaba

University of Tokyo, JSK Laboratory

3D object segmentation is a crucial ability of machine to percept the real world environment, and previous works on this problem used 2D segmentation using rgb-d sensors. In environments with heavy occlusions, however, there are fragments in segmentation results even with mapping in multiple views. We tackle this problem with object depth prediction by convolutional networks. In our method, the occluded surface depth of objects is predicted from input rgb images, and 3d points are generated from prediction and input depth. We use datasets with 3D annotations for training, and show the performance and real-time efficiency our method.

1. Introduction

3D object segmentation, capability of localizing and segmenting objects in real world, is a central problem of vision that has wide range of applications for robotics: navigation [Elfes 89, Furuta 16] and manipulation [Rusu 10, Eppner 16]. One of the main difficulties on three-dimensional segmentation is the handling of occlusions, and previous works use multi-view methods to handle this problem [Zeng 16, Wada 16]. However, multi-view capturing takes time, furthermore, there are situations where multi-view strategy does not contribute because of heavy occlusions and tightly packed objects.

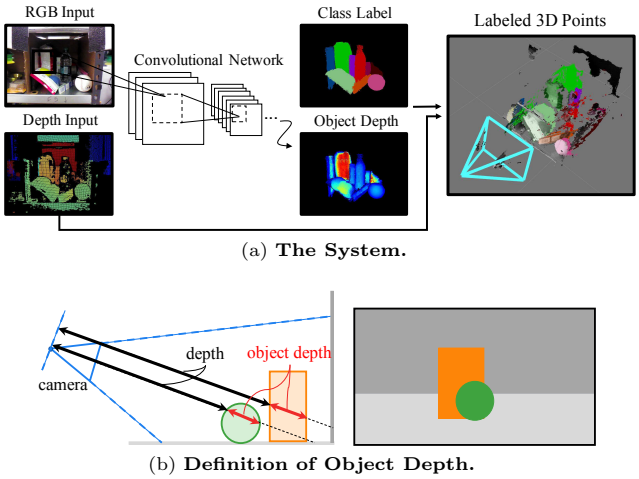
In order to solve these problems, we propose prediction of "object depth", distance from seen object surface to that occluded from camera angle (Fig.1b). The object depth is naturally dependant to object class, therefore in our method the pixel-wise object class and depth are simultaneously predicted.

The overall system for 3d object segmentation is shown in Fig.1a, in which inputs comes from rgb-d sensor, and labeled 3d points are generated using the sensed depth and network outputs. The predicted object depth is converted to the occluded surface depth by adding it with the seeable surface depth. The both seenable and occluded depth are converted to 3d points (point cloud), and labeled using the result of pixel-wise label prediction. Our proposed system successfully predicts occluded object surface, and generates 3d object model with a single view. In the experiment, we evaluate our method using a dataset with 6d pose annotations of mesh models, and show its efficiency.

2. Related works

Depth prediction

Previous works on depth prediction/estimation use learning-based method [Eigen 15, Liu 15] using large scale training sets of rgb and depth image [Saxena 09, Silberman 12]. In these previous works, depth estimation is mainly tackled without object class information, because surface depth is not defined with object classes and relatively unrelated. However, object depth is the dimension of each object as defined in Fig.1b, and use of object class information is necessary.



(a) **The System.**

(b) **Definition of Object Depth.**

Fig. 1: **3D Segmentation System with Object Depth Prediction.**

3D object segmentation

Three-dimensional multi-class object segmentation is tackled with combination of 2D segmentation and depth input [Eppner 16], multi-view frame mapping [Wada 16, Zeng 16], and voxel reconstruction [Song 16]. Our proposed method is relatively closed to previous voxel reconstruction method [Song 16] in terms that both methods predicts unseen three-dimensional property of object and class. Ours has two advantages: first, the higher speed of computation for 3D reconstruction by predicting only object surface compared to prior work that predicts for all voxels, second, the denser resolution of prediction by predicting pixel-wise compared to voxel-wise.

Contact: Kentaro Wada. Graduation School of Information Science and Technology, The University of Tokyo. 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan. +81-3-5841-7416, wada@jsk.imi.i.u-tokyo.ac.jp.

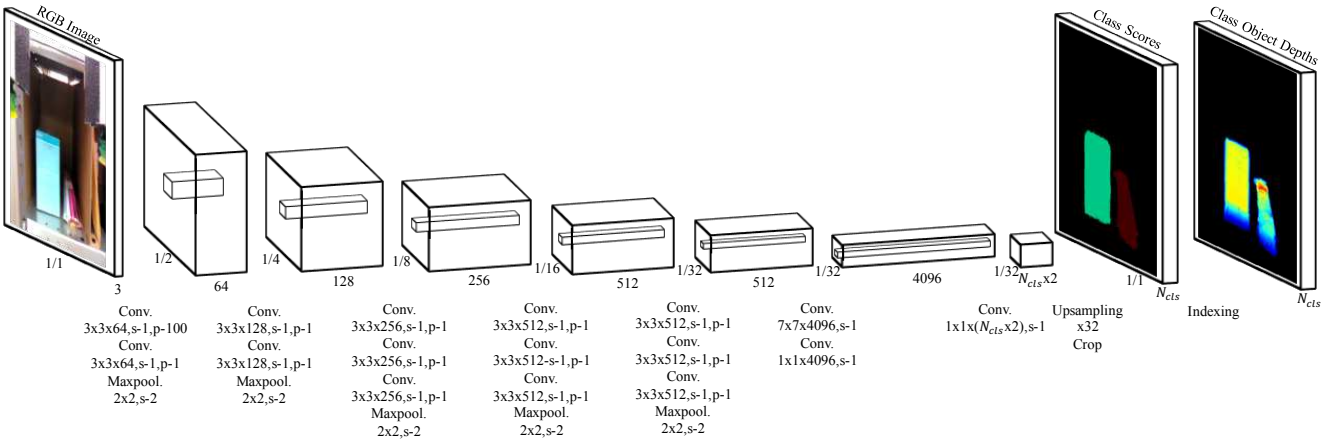


Fig. 2: **Network Model.** Filter size of convolutional and pooling layer is represented as (kernel height) × (kernel width) × (output channels), s-(stride size), p-(padding size).

3. Fully convolutional network for object depth prediction

In this section, we describe the method to predict pixel-wise class label C and object depth O from input image I.

Network input and output

Input of the network f is RGB image I, and outputs are pixel-wise class scores C^{score} and class object depth O^{cls}: $C^{score},\ O^{cls} = f(I)$. The tensor size of I is $(H, W, 3)$, where H and W represents image height and width 3 does RGB channels, that of C^{score} and O^{cls} is (H, W, N_{cls}) where N_{cls} is number of object classes. Considering the test time of object depth prediction, depth image also can be sensed by rgb-d sensor, but we did not used it as the network input because of its noisiness.

From outputs of the network, the class label and object depth are given as follows:

$$c = C_{ij} = \underset{k}{\text{argmax}}(C^{score}_{ijk}), \quad O_{ij} = O^{cls}_{ijc}. \tag{1}$$

where i and j is the index of image height and width, and note that C and O has tensor size $(H, W, 1)$. In this output design, the network estimates class dependent object depths O^{cls} whose number of channels is N_{cls}, but it is also considerable of use of single channel for object depth prediction. In this case, network directly predicts O and there will be no dependency between class prediction and depth prediction. However, it is reasonable of these dependency because "object depth" is defined with the classes. We show its efficiency of predicting class dependent object depth in the experiment.

For object depth prediction, we apply sigmoid to the output of convolution layer to make the range in $[0, 1]$, and multiplied 1e3 to make unit meter to milimeter. In order to estimate the depth of object larger than 1m, there should be a scaling factor, but in the experiment, we handle object smaller than 1m and set scaling factor to 1.

Network architecture

We design the network architecture shown in Fig.2 based on the previous work on 2D object segmentation [Shelhamer 16], in which they use a fully convolutional network to estimate class score map C^{score} from input image I. In addition to the class scores, the object depth is also predicted in our network, so we double the number of filters in the last convolutional layer. The deconvolutional layer of network in previous work is replaced with bilinear upsampling for smaller parameters and faster training. We use ReLU function for activations after the convolutions except for the last layer aside from the last layer.

Loss function

The network loss comes from multi-tasks, class segmentation L^{cls} and object depth prediction L^{objd}:

$$L = L^{cls} + \lambda \cdot L^{objd} \tag{2}$$

where λ is the weight parameter of the two tasks, which is set to 1e4 in the experiment by comparing values in evaluation with no training. For class segmentation, we use pixel-wise softmax cross entropy loss:

$$\mathbb{1} = \begin{cases} 1 & (C^{gt}_{ij} = k) \\ 0 & (otherwise) \end{cases},\ L^{cls} = -\sum_{i,j} \mathbb{1} \cdot log(\sigma(C^{score}_{ij})_k) \tag{3}$$

where i and j is the index of image height and width, and k is that of channels and has range $[0, N_{cls}]$. And for object depth prediction, we use pixel-wise smooth l1 loss:

$$L^{objd} = \sum_{i,j} smooth_{L_1}(O^{gt}_{ij}, O_{ij}) \tag{4}$$

$$smooth_{L_1} = \begin{cases} 0.5x^2 & (|x| < 1) \\ |x| - 0.5 & (otherwise) \end{cases} \tag{5}$$

The mean squared error loss is also used in the experiments, but we find using smooth l1 loss makes it easy to converge the training while the line search of learning rate for the optimizer.

4. Dataset of object depth

In this section, we describe the aquisition of training dataset for object depth prediction.

6D pose annotated datasets

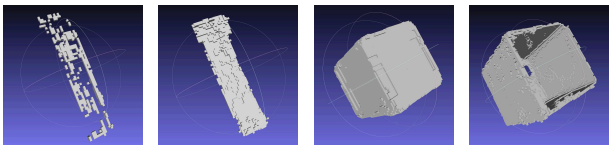
In order to create object depth dataset, we consider generating from that with 6D pose annotation of mesh models with raytracing using camera parameters. There are several datasets with pose annotations which are widely used in computer vision field [Xiang 14, Xiang 16], however, the dimensions of mesh models in the datasets are not correct comparing with that of real world. Because of this, we select the dataset previously used as a benchmark of 6D pose estimation in a robotic challenge [Zeng 16], in which pose of mesh model is annotated on RGB-D image for 39 object classes shown in Fig.3. We use 798 views (train: 598, validation: 200) for training and evaluation, which is a successfully pose annotated part of whole data.

(a) **Coodinates.** (b) **Image and object pose.** (c) **Mesh model.**

Fig. 3: **Dataset with 6D Pose Annotation.** (a): red point stands for world, blue for camera, and green for objects. (b): green points stand for projected poses of objects.

Object depth extraction from pose annotation

The object depth is acquired with raytracing using camera parameters on the 6D pose annotated mesh models. Though mesh models are given as shown in Fig.3c, it is not set of polygons but that of points in actual. To make models raytracable, we have created two polygon models from the points: one is generated with surface estimation and voxelization Fig.4a,4c, and the other is generated with converting 3d points to voxels Fig.4b,4d. As shown in the figures, both voxels generated two methods has pros and cons depending on the original object model, so we used the union set of the two voxels at raytracing.



(a) (b) (c) (d)

Fig. 4: **Voxelizations.**

5. Experiments

Object depth is class dependant

In this experiment, we verify the efficiency of predicting class dependant object depth. As described in Section 3, two types of network outputs are considerable for object depth: conditional object depth for class O^{cls} and class independant one O. For former output, the network has $2 \times N_{cls}$ convolution filters (N-Cls-objd), and $N_{cls} + 1$ (N-1ch-Objd)filters for latter. Model construction is the same in other layers.

For training, we initialized the weight of layers by copying from VGG16 network trained for an image classification task [Simonyan 14], aside from the filters for predicting object depth initialized with random values. We use Adam as the optimizer and 1e-5 for the learning rate after the line search, 3 for batch size and 300000 for iteration.

Accuracy metric for class segmentation is intersect-over-union averaged about class (meanIU) previously used [Shelhamer 16], and that for object depth prediction is accuracy (Ac_X) with error range X: 1mm, 10mm and 100mm.

Table 1 shows the comparision of class dependant/independant object depth, and it shows predicting class dependant object depth (N-Cls-objd) shows better performance than predicting independant depth (N-1ch-Objd) regarding of the metrics of object depth. In terms of class segmentation, however, N-Cls-objd shows lower accuracy than N-1ch-Objd. It is considerable that this is caused by the difficulty of learning multi-tasks, class segmentation and object depth prediction, simultaniously. We show the result using staged learning in the next section: firstly we trains about class segmentation, and about object depth prediction secondly.

Table 1: **Results of Object Depth Prediction.** Validation results extracted according to the best result of Ac_{1mm}.

Model	meanIU	Ac_{1mm}	Ac_{1cm}	Ac_{10cm}
N-1ch-objd	66.4	6.3	51.2	99.3
N-Cls-objd	61.9	6.8	53.5	99.4
N-Cls-objd-Staged	**79.3**	**6.9**	**54.7**	**99.5**

Staged learning for the multi-tasks

For the staged learning about the two tasks, we firstly trained the network only about class segmentation with initialized weight from pre-trained VGG16, and the result shows 84.7 meanIU. In the next stage, we trained all parameters as well as the last N_{cls} convolution filters with object depth dataset. The best result is shown in Table 1 at row of model N-Cls-objd-Staged, and it shows the efficiency of staged learning compared to the initialization from VGG (N-Cls-objd).

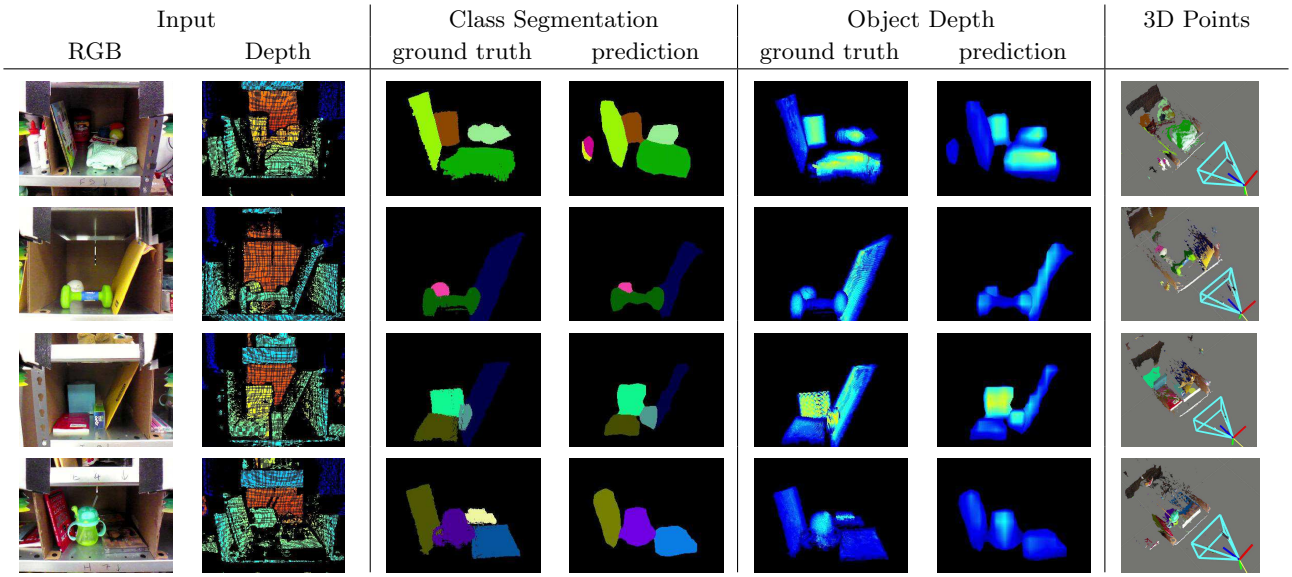
Real-time 3d object segmentation

The real-time efficiency is evaluated with emulating camera on validation dataset. The input rgb-d images has 30Hz, and N-Cls-objd-Staged predicts class and object depth in 7-8Hz. The qualitative results are shown in Table 2, and the class-awared occluded surface are successfully predicted.

6. Conclusions

We proposed a novel approach of 3d object segmentation with prediction of occluded surface of objects. Our proposed method is real-time and efficient to generate dense object model by predicting pixel-wise object depth.

Table 2: **3D Segmentation Results with Object Depth Prediction.**



References

[Eigen 15] Eigen, D. and Fergus, R.: Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture, in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2650–2658 (2015)

[Elfes 89] Elfes, A.: Using occupancy grids for mobile robot perception and navigation, *Computer*, Vol. 22, No. 6, pp. 46–57 (1989)

[Eppner 16] Eppner, C., Höfer, S., Jonschkowski, R., Martın-Martın, R., Sieverling, A., Wall, V., and Brock, O.: Lessons from the amazon picking challenge: Four aspects of building robotic systems, *Proceedings of Robotics: Science and Systems* (2016)

[Furuta 16] Furuta, Y., Wada, K., Murooka, M., Nozawa, S., Kakiuchi, Y., Okada, K., and Inaba, M.: Transformable semantic map based navigation using autonomous deep learning object segmentation, in *Humanoid Robots, IEEE-RAS 16th International Conference on*, pp. 614–620IEEE (2016)

[Liu 15] Liu, F., Shen, C., and Lin, G.: Deep convolutional neural fields for depth estimation from a single image, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5162–5170 (2015)

[Rusu 10] Rusu, R. B.: Semantic 3d object maps for everyday manipulation in human living environments, *KI-Künstliche Intelligenz*, Vol. 24, No. 4, pp. 345–348 (2010)

[Saxena 09] Saxena, A., Sun, M., and Ng, A. Y.: Make3d: Learning 3d scene structure from a single still image, *IEEE transactions on pattern analysis and machine intelligence*, Vol. 31, No. 5, pp. 824–840 (2009)

[Shelhamer 16] Shelhamer, E., Long, J., and Darrell, T.: Fully Convolutional Networks for Semantic Segmentation, *PAMI* (2016)

[Silberman 12] Silberman, N., Hoiem, D., Kohli, P., and Fergus, R.: Indoor segmentation and support inference from rgbd images, in *European Conference on Computer Vision*, pp. 746–760 (2012)

[Simonyan 14] Simonyan, K. and Zisserman, A.: Very deep convolutional networks for large-scale image recognition, *arXiv preprint arXiv:1409.1556* (2014)

[Song 16] Song, S., Yu, F., Zeng, A., Chang, A. X., Savva, M., and Funkhouser, T.: Semantic Scene Completion from a Single Depth Image, *arXiv preprint arXiv:1611.08974* (2016)

[Wada 16] Wada, K., Murooka, M., Okada, K., and Inaba, M.: 3D object segmentation for shelf bin picking by humanoid with deep learning and occupancy voxel grid map, in *Humanoid Robots, IEEE-RAS 16th International Conference on*, pp. 1149–1154IEEE (2016)

[Xiang 14] Xiang, Y., Mottaghi, R., and Savarese, S.: Beyond pascal: A benchmark for 3d object detection in the wild, in *Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on*, pp. 75–82IEEE (2014)

[Xiang 16] Xiang, Y., Kim, W., Chen, W., Ji, J., Choy, C., Su, H., Mottaghi, R., Guibas, L., and Savarese, S.: Objectnet3d: A large scale database for 3d object recognition, in *European Conference on Computer Vision*, pp. 160–176Springer (2016)

[Zeng 16] Zeng, A., Yu, K.-T., Song, S., Suo, D., Walker Jr, E., Rodriguez, A., and Xiao, J.: Multi-view Self-supervised Deep Learning for 6D Pose Estimation in the Amazon Picking Challenge, *arXiv preprint arXiv:1609.09475* (2016)