Two Stream 3D Semantic Scene Completion

Martin Garbade, Johann Sawatzky, Alexander Richard, Juergen Gall

Computer Science Institute III,
University of Bonn
{garbade, sawatzky, richard, gall}@iai.uni-bonn.de

Abstract. We propose a new paradigm for advancing 3D semantic scene completion based on RGBD images. We introduce a two stream approach that uses RGB and depth as input channels to a 3D convolutional neural network. Our approach boosts the performance of semantic scene completion by a significant margin. We further provide a study on several input encoding schemes and set a new state of art in 3D semantic scene completion on the NYU dataset.

Keywords: scene completion, semantic segmentation, 3D, RGBD

1 Introduction

Humans quickly infer the 3D semantics of a scene, i.e., an estimate of the 3D geometry and the semantic meaning of the surfaces. The perception, however, is not limited to the visible part of the scene. When looking at a mug on a table, a human can estimate the full geometry of both objects including parts which are invisible, since they are occluded by the objects themselves. This information is obtained from semantic understanding of the scene which allows to estimate the spatial extent of the objects from experience. Such an ability is highly desirable for autonomous agents, e.g., to navigate or interact with objects. A robot for example could plan ahead given a single view instead of exhaustively explore the occluded parts of a scene first. Instead he should have an intuition about the geometry behind the surface he sees.

RGB-D sensors provide color and depth information of a scene. Although the depth information allows to directly infer the geometry of a scene, the resulting representation is very sparse since large parts of the 3D scene are occluded and not visible. In this work, we aim to estimate the semantics not only of the visible part, but of the entire scene including the occluded space. To this end, we build on the work of Song et al. [1]. They show that semantic scene understanding and 3D scene completion benefit from each other. On one hand, recognizing a part of the object helps to estimate its location in the 3D space and the voxels it occupies. On the other hand, knowing the occupancy in the 3D space gives information on form and size of the object and thus facilitates semantic recognition. For estimating for each voxel in the scene the occupancy and semantic label, they proposed an end-to-end trainable 3D convolutional neural network (3D CNN) which incorporates context from a large field of view

via dilated convolutions. The approach, however only uses depth as input and neglects the RGB image. This means that the semantic label has to be inferred from the geometry alone and properties such as color, texture, or reflectance are not taken into account.

We therefore extend the approach [1] by keeping its beneficial context incorporation and end-to-end trainability while modifying it to leverage semantic information inferred from the RGB image at the input stage as well as at the loss. Given a single RGB-D image, we first use a 2D CNN to infer the semantic labels from the RGB data and construct an incomplete 3D semantic tensor. To this end, we map the inferred semantic labels to the 3D space and label each visible surface voxel by the inferred class label. The 3D semantic tensor is incomplete since it only contains the labels of the visible voxels but not of the occluded voxels. From the depth image, we construct an incomplete 3D occupancy tensor using a flipped truncated signed distance function. Both tensors are then used as input for a 3D CNN that infers a complete 3D semantic tensor, which includes the occupancy and semantic labels for all voxels.

Using the RGB images as input leads to significant performance advantage in scene completion and semantic scene completion as our experiments section shows. We outperform [1] by a substantial margin of 6.5 % on NYU. This implies that RGB images provide a rich discriminative signal.

2 Related Work

Several works address the problem of semantic segmentation of RGBD images [2], [3–5]. However, they only predict semantic labels for the visible pixels of the image, which means that occluded voxels are not reconstructed.

A possible strategy for semantic scene completion is 3D object proposal generation and subsequent 3D shape completion of the respective object. There is a variety of methods for 3D shape completion [6–13]. To predict a voxel wise semantic, holes between objects have to be filled. As long as these missing parts are small, they can be filled using plane fitting [14] or object symmetry [15, 16]. However non detected objects heavily disturb the 3D scene completion. Completing the scene geometry without predicting the semantics has been addressed by [17]. Their model assumes that objects of semantically dissimilar classes can still be represented by similar 3D shapes, i.e. it is possible to predict the unobserved voxels from the frontal geometry. However, this approach fails for complex scenes where these geometric constraints are not true.

An alternative is to combine instance level 3D mesh models to fit the scene geometry [15, 18–26]. However, the variance within objects of the same category cannot be modeled and the number of models is limited to the available amount of CAD models. Another tradoff of these models is that a high expressibility allows the combination of individual mesh grids to potentially better fit to the geometry of the scene but makes the model retrieval more difficult. One can neglect the fine grained details and simply fit 3D primitives to to the scene as done in [27–29].

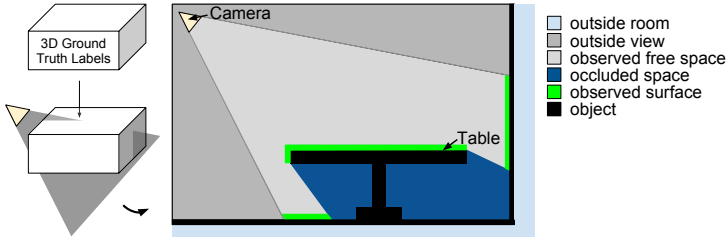

Fig. 1. Using the protocol of [1], ground-truth labels are provided for all voxels of a 3D volume. Voxels that are outside the intersection of the camera frustum and ground-truth volume are outside the room or outside the view and not taken into account. Within the intersection, there are observed surface voxels (green) and observed non-occupied voxels (light gray), but other voxels are not observed by the camera. These voxels are either non-occupied (blue) or belong to an object (black).

Various contextual cues proved to be helpful for semantic scene completion: Physical reasoning is employed in [30], while [31] predict voxel labels with a CRF whose unary potentials are determined by floor plans. However, the CRF only proves useful to model short distance contextual information. One can optimize for semantic scene completion jointly with multi-view reconstruction as done by [32] and [33]. In these works, the models are not trained end to end but use predefined features and fuse the context and features with hand crafted methods.

To facilitate learning of scene completion in an end to end manner, [34–36] collected large scale datasets with real world data. Earlier, synthetic datasets were employed for rendering 2D ground truth for semantic segmentation [37,38], object completion [9,39]. A first synthetic scene dataset was proposed by [37], a large scale dataset was used in [1] to train the first end to end semantic scene completion network.

Deep learning based semantic scene reconstruction was done from single views [1] as well as more recently from multiple views [40]. Another similar line of research is point cloud classification here deep learning also showed promising results, first in [41] and later in [42] where it was combined with reinforcement learning and RNNs.

3 Two Stream Semantic Scene Completion

3.1 Semantic Scene Completion

The goal of 3D semantic scene completion is to classify every voxel in the view frustum into one of $K+1$ labels $c = c_0, ..., c_K$ where c_0 represents an empty voxel and $c_1, ..., c_K$ represents one of K class labels like ceiling, floor, wall, window, chair, bed, sofa, table, tv, furniture and object. As illustrated in Figure 1, the camera observes only a part of the scene while other voxels are occluded. The occluded voxels can either be empty (c_0) or belong to one of the K classes.

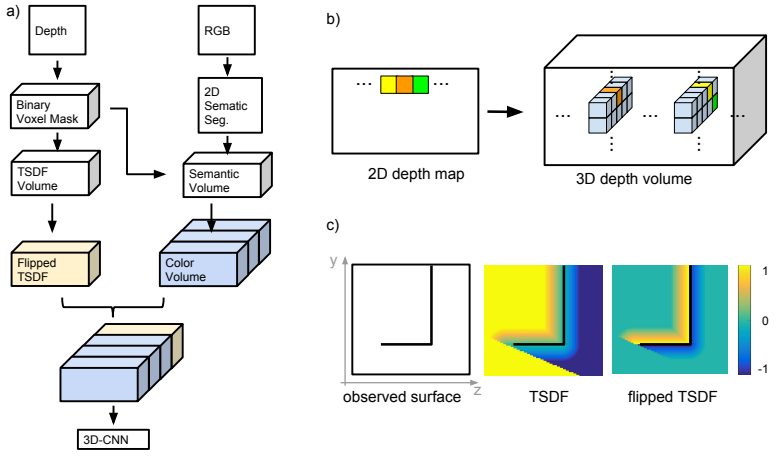

Fig. 2. a) The proposed two stream approach for semantic scene completion transforms first the depth data and RGB image into a volumetric representation, which represent the geometry and semantic of the visible scene and then uses a 3D-CNN to infer a 3D semantic tensor for the entire scene. b) Given 2D depth map and camera pose, a binary voxel mask is created by setting each voxel that belongs to a depth pixel to one and all other voxels to zero (blue). c) Comparison of TSDF vs flipped TSDF. One can see the long 'shadow' caused by the observed surface which produces high gradients at the occlusion boundary (between -1 and 1). In flipped TSDF this effect is suppressed. The gradient is highest at the surface.

To address the task of 3D semantic scene completion, we propose an approach that leverages two input streams, namely RGB and depth. An overview of the approach is given in Figure 2 a). While the depth data is converted into volumetric representation using a flipped truncated signed distance function (TSDF), which will be described in Section 3.2, the RGB image is first processed in a separate branch to infer 2D semantic segmentation maps and then transformed into a volumetric representation referred to as color-volume (Section 3.3). The volumetric representations of both streams are then concatenated and fed to a 3D convolutional neural network (3D-CNN). The 3D-CNN infers a 3D semantic tensor where every voxel is classified as either being empty or to belong to one of the K semantic classes. In the following, each step will be discussed in detail.

3.2 Depth Input Stream

To obtain the volumetric input encoding, the depth map is projected into a regular voxel grid using the camera pose which is provided along with each image. The voxel grid is of size 240 x 144 x 240 voxels and encodes a scene of 4.80m horizontally, 2.88m vertically, and 4.80m in depth with a resolution of 0.02m. For every pixel in the depth map, its corresponding voxel in the 3D input volume is computed using the camera pose. The obtained binary voxel

mask encodes the location of surface points that are visible to the camera, see Figure 2 b).

This binary mask is used to first compute a truncated signed distance function (TSDF) encoding as illustrated in Figure 2 c). In the TSDF, every voxel contains as value the distance d to the next surface point. The sign of the distance value indicates whether a voxel lies in empty (1) or occluded space (-1) as shown in Figure 3. The TSDF has the disadvantage of having high gradients at the occlusion boundary, i.e., the boundary between observed and unobserved space behind a surface. Therefore in the TSDF encoding every surface yields a shadow into the unobserved space [1].

To provide a more meaningful input signal, the TSDF is transformed into a flipped TSDF, where every distance value d is converted into a distance d_f which is 1 or -1 at a surface and linearly falls to 0 at a distance d_{max} from the surface:

$$d_f = \mathcal{H}(d_{max} - d)sign(d)(d_{max} - d) \tag{1}$$

where d_{max} is the maximum distance of 24 cm and $\mathcal{H}$ is the Heaviside function:

$$\mathcal{H}(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0. \end{cases} \tag{2}$$

As pre-processing, all 3D scenes are rotated such that the room orientations are aligned. For indoor room scenes, one can assume that most of the observed surface normals are oriented either like the normals of the walls, floor or ceiling, which are usually planar. Therefore a principal component analysis of the surface normals is used to infer the room orientation, which is used to align the scene.

3.3 Color Input Stream

The input RGB image is first processed by a 2D-CNN [43] for semantic segmentation. The network is an adaptation of the Resnet101 architecture [44] for semantic segmentation. While all but one pooling layer are omitted, dilated convolutions are used to keep the output resolution high while simultaneously increasing the receptive field. The output is downsampled by a factor of 4 with respect to the input. The output is then upsampled using bilinear interpolation. The 2D-CNN predicts the softmax probabilities for every class and pixel. A densely connected CRF [45] is then used in combination with the inferred class probabilities and the RGB image to refine the semantic segmentation map. For training, we use the same setting as in [43]. As initialization, we use a model that is pre-trained on MSCOCO [46] and fine-tune it on the dataset for 3D semantic scene completion.

As in Section 3.2, we convert the 2D segmentation map into a volumetric representation. Since each pixel in the depth map corresponds to a pixel in the 2D semantic segmentation map, every class pixel can be projected into the 3D volume at the location of its corresponding depth value. This yields an incomplete 3D semantic tensor that assigns to every surface voxel its corresponding

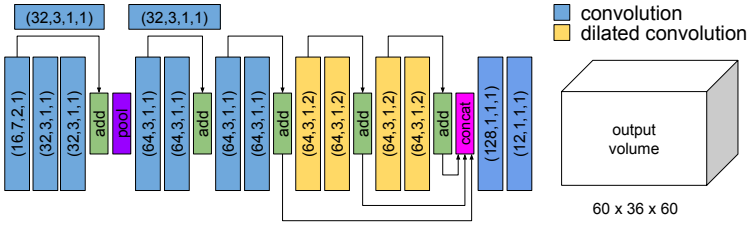

Fig. 3. Architecture of the 3D-CNN. The parameters of the convolution kernels are denoted as (number of filters, kernel size, stride, dilation). All but the last convolution layer have a ReLu activation function assigned to it. Arrays indicate skip connections [44] where the output of one convolution layer is added to another output at a later stage. Pool signifies max pooling. The output is a volume that is 4 fold downsampled with respect to the input of the 3D CNN and encodes for every voxel the probability of it being empty (label 0) or to belong to one of 11 semantic classes.

class label. Encoding semantical classes with a 1 channel only implies semantical proximity of classes given the numerical proximity of their class values. Therefore a numerical equidistance would be desirable. Ideally, the class labels are encoded by one-hot encoding. This is, however, impractical due to memory constraints since it requires to store a K dimensional vector per voxel.

We therefore represent the semantic information by a lower dimensional vector. We use a three-dimensional vector and encode the classes linearly from $(0,0,1)$ over $(0,1,1)$, $(0,1,0)$, $(1,1,0)$ to $(1,0,0)$.

3.4 3D-CNN

For the 3D-CNN, we adapt the architecture of [1] by increasing the number of input channels of the first convolutional layer such that it fits to our input. The architecture is inspired by the 2D-CNN for semantic segmentation. The major difference apart from using 3D instead of 2D convolutions is that the network only has a depth of 14 convolutional layers. Therefore the network has significantly less parameters than its two dimensional counter part. Moreover, batch-normalization layers are omitted due to the small size of the batches.

We adapt the training protocol of [1] as follows. We increase the batch size from 1 to 4 which is the maximum to fit into the memory of a 11GB GTX 1080 Ti GPU. Since the gradients are also accumulated over 4 steps before backpropagation, this leads to an increase of the effective batch size from 4 to 16. We train for 40,000 steps with a learning rate of 0.01 that is reduced by a factor of 0.1 after 20,000 iterations. As optimizer SGD with momentum is applied. As initialization we chose a random intialization with a gaussian distribution with mean $\mu = 0$ and a standard deviation of $\sigma = 0.01$.

The output of the 3D-CNN is a semantic tensor of size 60 x 36 x 60 x $(K+1)$, where K is the number of object classes and an additional class is added for empty voxels. We compute a softmax cross entropy loss on the unnormalized

network outputs y:

$$\mathcal{L} = -\sum_{i,c} w_{ic}\hat{y}_{ic} \log\left(\frac{e^{y_{ic}}}{\sum_{c' \in \mathcal{C}} e^{y_{ic'}}}\right) \tag{3}$$

where $\hat{y}_{ic}$ are the binary ground truth vectors, i.e., $\hat{y}_{ic} = 1$ if voxel i is labeled by class c, and w_{ic} are the loss weights. Since the ratio of empty vs occupied voxels is 9:1, the empty space is randomly subsampled. Therefore w_{ic} is chosen as binary mask such that only 2N empty voxels are selected for loss calculation where N is the number of occupied voxels in the scene.

4 Experimental Evaluation

4.1 Evaluation Metric

During evaluation only voxels that are part of the occluded space and within both the room and the field-of-view are considered (see Figure 1). According to the evaluation protocol [1] only a subset of voxels is used to evaluate the performance. While generating the 3D semantic labels from the annotated CAD models, every voxel in the input volume is marked as being on surface, free space, occluded space, outside field of view, outside room or outside ceiling. For semantic scene completion, a binary evaluation mask is computed such that the evaluation metric is only computed for voxels which are either occluded, on surface or close to the surface (within the range of the TSDF function). For scene completion another mask is computed which comprises all voxels in the occluded space. To assess the quality of 3D scene completion, several metrics are computed. First we compute the Jaccard index, which measures the intersection over union (IoU) between ground truth and predicted voxel for every object category $c_1, ..., c_{11}$. As an overall segmentation performance we compute the average across all classes. For scene completion all voxels are considered to belong to one of the two classes empty vs non-empty. All object categories $c_1, ..., c_{11}$ are counted as "non-empty". For completion, IoU as well as precision and recall are computed.

4.2 Datasets

We evaluate our method on the NYUv2 dataset (in the following denoted as NYU). NYU consists of 1449 indoor scenes that are captured via a kinect sensor. For 3D semantic scene completion labels, we use the 3D annotated labels provided by Hoiem et al. [47]. These annotations consist of CAD models that are fitted into the scene. Since the CAD models do not exactly fit the shape of the annotated objects and neglect small objects such as clutter, there is a significant mismatch between the kinect input data ad the output labels. To address this problem (following [47]) depth maps generated from the backprojections of the 3D annotations are used for training and evaluation as well denoted as CAD.

Table 1. Comparison to the state-of-the-art.

NYU Kinect		Scene Completion			Semantic Scene Completion										
method	trained on	prec.	recall	IoU	ceil.	floor	wall	win.	chair	bed	sofa	table	tv	furn.	objs.	avg.
Lin et al. [28]	NYU	58.5	49.9	36.4	0.0	11.7	13.3	14.1	9.4	29.0	24.0	6.0	7.0	16.2	1.1	12.0
Geiger et al. [18]	NYU	65.7	58.0	44.4	10.2	62.5	19.1	5.8	8.5	40.6	27.7	7.0	6.0	22.6	5.9	19.6
SSCNet [1]	NYU	57.0	94.5	55.1	15.1	94.7	24.4	0.0	12.6	32.1	35.0	13.0	7.8	27.1	10.1	24.7
SSCNet [1]	SUNCG+NYU	59.3	92.9	56.6	15.1	94.6	24.7	10.8	17.3	53.2	45.9	15.9	13.9	31.1	12.6	30.5
Ours:	NYU	69.5	82.7	60.7	12.9	92.5	25.3	20.1	16.1	56.3	43.4	17.2	10.4	33.0	14.3	31.0
NYU CAD																
Zheng et al. [30]	NYU	60.1	46.7	34.6												
Firman et al. [17]	NYU	66.5	69.7	50.8												
SSCNet [1]	NYU	75.0	92.3	70.3												
SSCNet [1]	SUNCG+NYU	75.4	96.3	73.2	32.5	92.6	49.2	8.9	33.9	57.0	59.5	28.3	8.1	44.8	25.1	40.0
Ours:	NYU	80.2	91.0	74.2	33.8	92.9	46.8	27.0	27.9	61.6	51.6	27.6	26.9	44.5	22.0	42.1

For 2D semantic segmentation besides NYU, we use SUN-RGBD [48] for data augmentation. SUN-RGBD comprises NYUv2 as well as Berkeley B3DO [49], and SUN3D [50]. We use all images except for NYUv2 test set as training images. We manually map the 37 class scheme to the 11 classes used by NYU.

4.3 Comparison to State-of-the-Art

We evaluate our approach on two test sets NYU CAD and NYU Kinect in the following denoted as 'CAD' and 'Kinect'. The quantitative results can be found in Table 1.

We set a new state of art in scene completion with 60.7 % (Kinect) and 74.2 % IoU (CAD). Thereby we surpass the latest approach by Song et al. [1] by 4.1 % on Kinect and 1 % on CAD. However in contrast to our approach Song et al. use SUNCG for data augmentation. Compared to their performance when training on the same dataset (NYU) we outperform them by 5.6 % (NYU Kinect) and 3.9 %.

On the metrics precision and recall we see a slight inversion compared to Song et al. While our precision is higher, the recall performs worse than their approach.

On semantic scene completion we outperform Song by 6.3 % (Kinect) when using the same training dataset (NYU). Since Song et al. do not report numbers on CAD we compute them using the models provided by the authors. This model is pretrained on the synthetic SUNCG dataset. Nevertheless we outperform their approach by 2.1 %. Zheng [30] and Firman [17] only address scene completion and therefore don't provide numbers on semantic scene completion. With respect to the individual class accuracies we consistently (on both datasets) outperform Song et al. on the classes window and bed.

All non deep learning approaches lack significantly behind our approach. In the case of completion, the closest to our approach with -16.3 % is th approach by Geiger et al. [18] on Kinect and with -23.4 % the approach by Firman et al. [17] on CAD. For semantic scene completion the strongest non deep learning approach is with -11.4 % the approach by Geiger et al. on Kinect.

Table 2. Results of the ablation study. We evaluate our model against varying input encodings and training parameters. Evaluation is done using NYU CAD.

	segmentation input		num input channels		batch size		
	gt 2D	pred 2D	1ch	3ch	1	3	4
Scene Completion	75.8	74.2	74.7	75.8	76.9	76.0	75.8
Semantic Scene Completion	52.9	42.0	39.0	52.9	53.1	53.8	52.9

4.4 Ablation Study

In the following we conduct an ablation study to analyze the design choices of our model.

Ground Truth vs Predicted Segmentation Input In order to find an upper bound for our prediction quality using 2D semantic segmentation as input, we use ground truth semantic segmentation masks for the RGB images as input to the 3D-CNN. As one can see from Table 2, using ground truth gives us 75.8 % for semantic scene completion (comp) and 52.9 % for semantic scene completion (seg) on CAD which corresponds to an increase of +1.6 % (comp) and +10.9 %(seg). We see that ground truth prediction still have a huge impact on scene completion performance. For scene completion th effect is almost not significant.

Varying Batch Size During training we compare different batch sizes 1,3 and 4. From Table 2. One can see that the effect of batch size is minimal.

1 vs 3 Channel Input Encoding To provide numerical equidistance between classes, a one hot encoding for every voxel would be useful. However this approach is prohibited due to memory limitations. Additionally this would increase sparsity of the input features. As a compromise we choose a 3 channel encoding where instead of a single value every voxel is mapped to 3 values similar to a RGB color encoding for a 2D semantic segmentation map. Table 2 shows that having 3 channels increases performance over the 1 channel encoding. Therefore we argue that the 3 channel setup provides a good approximation of the one-hot encoding while also being less sparse.

5 Conclusions

We have introduced a two stream approach to 3D semantic scene completion which uses 2D semantic segmentations inferred fom a RGB color image as well as depth as input features. We set a new state of the art in semantic scene completion on the NYU dataset. We also provide an ablation study of our model and demonstrate that a multi-channel volumetric input encoding of the color stream yields the best performance.

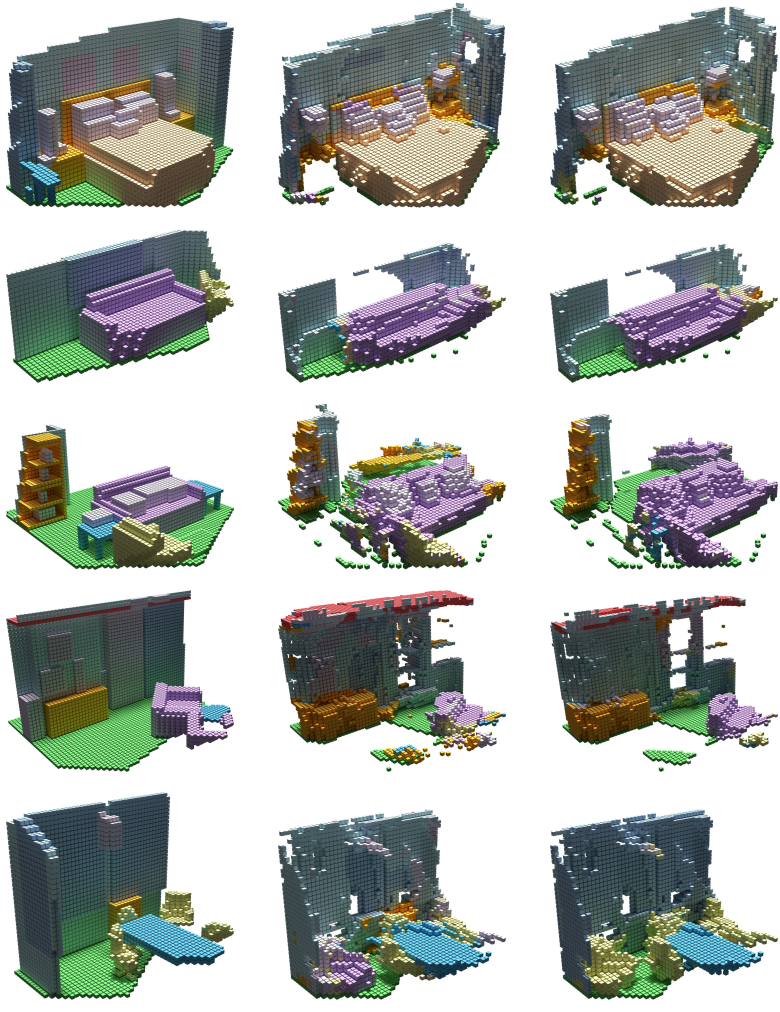

Fig. 4. Qualitative Results. Left: Ground truth, middle: Song et al. [1]. Right: Ours. Overall our predicted 3D scenes look less cluttered and show a higher voxel class accuracy as compared to Song et al. [1]

References

1. Song, S., Yu, F., Zeng, A., Chang, A.X., Savva, M., Funkhouser, T.: Semantic Scene Completion from a Single Depth Image. In: IEEE Conference on Computer Vision and Pattern Recognition. (2017)
2. Lai, K., Bo, L., Fox, D.: Unsupervised feature learning for 3d scene labeling. In: IEEE International Conference on Robotics and Automation (ICRA). (2014) 3050–3057
3. Ren, X., Bo, L., Fox, D.: RGB-(D) scene labeling: Features and Algorithms. In: IEEE Conference on Computer Vision and Pattern Recognition. (2012)
4. Gupta, S., Arbelaez, P., Malik, J.: Perceptual organization and recognition of indoor scenes from RGB-D images. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE (2013) 564–571
5. Nathan Silberman, D.H.P.K., Fergus, R.: Indoor Segmentation and Support Inference from RGBD Images. In: European Conference on Computer Vision. (2012)
6. Rock, J., Gupta, T., Thorse, j., Gwak, J., Shin, D.: Completing 3D object shape from one depth image. In: IEEE Conference on Computer Vision and Pattern Recognition. (2015)
7. Thanh Nguyen, Duc and Hua, Binh-Son and Tran, Khoi and Pham, Quang-Hieu and Yeung, Sai-Kit: A field model for repairing 3D shapes. In: IEEE Conference on Computer Vision and Pattern Recognition. (2016)
8. Varley, J., DeChant, C., Richardson, A., Nair, A.a.J., Allen, P.: Shape completion enabled robotic grasping. In: Arxiv. (2016)
9. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3D ShapeNets: A deep representation for volumetric shapes. In: IEEE Conference on Computer Vision and Pattern Recognition. (2015)
10. Wang, W., Huang, Q., You, S., Yang, C., Neumann, U.: Shape Inpainting using 3D Generative Adversarial Network and Recurrent Convolutional Networks. CoRR (2017)
11. Yang, B., Wen, H., Wang, S., Clark, R., Markham, A., Trigoni, N.: 3D Object Reconstruction from a Single Depth View with Adversarial Learning. CoRR (2017)
12. Han, X., Li, Z., Huang, H., Kalogerakis, E., Yu, Y.: High-Resolution Shape Completion Using Deep Neural Networks for Global Structure and Local Geometry Inference. In: IEEE International Conference on Computer Vision (ICCV). (October 2017)
13. Dai, A., Qi, C.R., Nießner, M.: Shape Completion using 3D-Encoder-Predictor CNNs and Shape Synthesis. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2017)
14. Monszpart, A., Mellado, N., Brostow, G.J., Mitra, N.J.: RAPter: Rebuilding Man-made Scenes with Regular Arrangements of Planes. ACM Transactions on Graphics **34** (2015) 103:1–103:12
15. Kim, Y.M., Mitra, N.J., Yan, D.M., Guibas, L.: Acquiring 3D Indoor Environments with Variability and Repetition. ACM Transactions on Graphics **31** (2012) 138:1–138:11
16. Mattausch, O., Panozzo, D., Mura, C., Sorkine-Hornung, O., Pajarola, R.: Object detection and classification from large-scale cluttered indoor scans. Computer Graphics Forum **33**(2) (2014) 11–21
17. Firman, M., Mac Aodha, O., Julier, S., Brostow, G.J.: Structured Prediction of Unobserved Voxels From a Single Depth Image. In: IEEE Conference on Computer Vision and Pattern Recognition. (2016)

18. Geiger, A., Wang, C.: Joint 3D Object and Layout Inference from a single RGB-D Image. In: German Conference on Pattern Recognition (GCPR). Volume 9358. (2015) 183–195
19. Gupta, S., Arbeláez, P.A., Girshick, R.B., Malik, J.: Aligning 3D models to RGB-D images of cluttered scenes. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society (2015) 4731–4740
20. Lai, K., Fox, D.: Object Recognition in 3D Point Clouds Using Web Data and Domain Adaptation. **29** (2010) 1019–1037
21. Mattausch, O., Panozzo, D., Mura, C., Sorkine-Hornung, O., Pajarola, R.: Object Detection and Classification from Large-Scale Cluttered Indoor Scans. (2014)
22. Nan, L., Xie, K., Sharf, A.: A Search-classify Approach for Cluttered Indoor Scene Understanding. ACM Transactions on Graphics **31** (2012) 137:1–137:10
23. Song, S., Xiao, J.: Deep Sliding Shapes for Amodal 3D Object Detection in RGB-D Images. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2016)
24. Shao, T., Xu, W., Zhou, K., Wang, J., Li, D., Guo, B.: An Interactive Approach to Semantic Modeling of Indoor Scenes with an RGBD Camera. (2012)
25. Li, Y., Dai, A., Guibas, L., Niessner, M.: Database-Assisted Object Retrieval for Real-Time 3D Reconstruction. **34** (2015) 435–446
26. Shi, Y., Long, P., Xu, K., Huang, H., Xiong, Y.: Data-driven Contextual Modeling for 3D Scene Understanding. **55** (2016) 55–67
27. Jiang, H., Xiao, J.: A linear approach to matchig cuboids in RGBD images. (2013)
28. Lin, D., Fidler, S., Urtasun, R.: holistic scene understanding for 3D object detection with RGBD cameras. (2013)
29. Song, S., Xiao, J.: Deep sliding shapes for amodal 3D object detection in rgb-d images. (2016)
30. Zheng, B., Zhao, Y., Yu, J.C., Ikeuchi, K., Sx-Cx, Z.: Beyond point clouds: Scene understanding by reasoning geometry and physics. (2016)
31. Kim, B.S.a.P., Savarese, S.: 3D scene understanding by Voxel-CRF. IEEE Conference on Computer Vision and Pattern Recognition (2013)
32. Blaha, M., Vogel, C., Richard, A., Wegner, D., Pock, T., Schindler, K.: Large-scale semantic 3d reconstruction: An adaptive multi-resolution model for multi-class volumetric labeling. (2016)
33. Häne, C., Zach, C., Cohen, A., Angst, R., Pollefeys, M.: Joint 3D Scene Reconstruction and Class Segmentation. IEEE Conference on Computer Vision and Pattern Recognition (2013) 97–104
34. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T.A., Nießner, M.: ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes. (2017)
35. Zhang, Y., Song, S., Yumer, E., Savva, M., Lee, J.Y., Jin, H., Funkhouser, T.: Physically-Based Rendering for Indoor Scene Understanding Using Convolutional Neural Networks. (2017)
36. Chang, A., Dai, A., Funkhouser, T., Halber, M., Niessner, M., Savva, M., Song, S., Zeng, A., Zhang, Y.: Matterport3D: Learning from RGB-D Data in Indoor Environments. International Conference on 3D Vision (3DV) (2017)
37. Handa, A., Patraucean, V., Badrinarayanan, V., Stent, S., Cipolla, R.: SceneNet: Understanding Real World Indoor Scenes With Synthetic Data. (2015)
38. Richter, S.R., Vineet, V., Roth, S., Koltun, V.: Playing for Data: Ground Truth from Computer Games. In Leibe, B., Matas, J., Sebe, N., Welling, M., eds.: European Conference on Computer Vision (ECCV). (2016)

39. Chang, A.X., Funkhouser, T.A., Guibas, L.J., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., Yu, F.: ShapeNet: An Information-Rich 3D Model Repository. (2015)
40. Dai, A., Rithie, D., Bokeloh, M., Reed, S., Sturm, J., Nießner, M.: ScanComplete: Large-Scale Scene Completion and Semantic Segmentation for 3D Scans. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
41. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
42. Liu, F., Li, S., Zhang, L., Zhou, C., Ye, R., Wang, Y., Lu, J.: 3DCNN-DQN-RNN: A Deep Reinforcement Learning Framework for Semantic Parsing of Large-scale 3D Point Clouds. CoRR (2017)
43. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs. In: International Conference on Learning Representations. (2015)
44. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. In: IEEE Conference on Computer Vision and Pattern Recognition. (2016)
45. Krähenbühl, P., Koltun, V.: Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials. In: Neural Information Processing Systems (NIPS). (2011)
46. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision, Springer (2014) 740–755
47. Rock, J., Gupta, T., Thorsen, J., Gwak, J., Shin, D., Hoiem, D.: Completing 3D object shape from one depth image. In: IEEE Conference on Computer Vision and Pattern Recognition. (2015)
48. Song, S., Lichtenberg, S.P., Xiao, J.: SUN RGB-D: A RGB-D Scene Understanding Benchmark Suite. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2015)
49. Janoch, A., Karayev, S., Jia, Y., Barron, J.T., Fritz, M., Saenko, K., Darrell, T.: A category-level 3d object dataset: Putting the kinect to work. In: Consumer Depth Cameras for Computer Vision. Springer (2013) 141–165
50. Xiao, J., Owens, A., Torralba, A.: Sun3d: A database of big spaces reconstructed using sfm and object labels. In: IEEE International Conference on Computer Vision (ICCV). (2013) 1625–1632