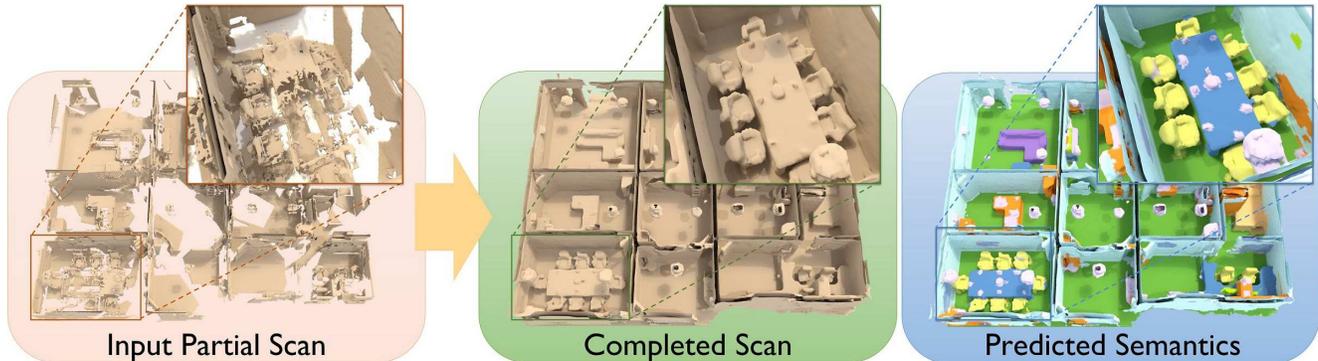


ScanComplete: Large-Scale Scene Completion and Semantic Segmentation for 3D Scans

Angela Dai^{1,3,5} Daniel Ritchie² Martin Bokeloh³ Scott Reed⁴ Jürgen Sturm³ Matthias Nießner⁵
¹Stanford University ²Brown University ³Google ⁴DeepMind ⁵Technical University of Munich



3D scans of indoor environments suffer from sensor occlusions, leaving 3D reconstructions with highly incomplete 3D geometry (left). We propose a novel data-driven approach based on fully-convolutional neural networks that transforms incomplete signed distance functions (SDFs) into complete meshes at unprecedented spatial extents (middle). In addition to scene completion, our approach infers semantic class labels even for previously missing geometry (right). Our approach outperforms existing approaches both in terms of completion and semantic labeling accuracy by a significant margin.

Abstract

We introduce *ScanComplete*, a novel data-driven approach for taking an incomplete 3D scan of a scene as input and predicting a complete 3D model along with per-voxel semantic labels. The key contribution of our method is its ability to handle large scenes with varying spatial extent, managing the cubic growth in data size as scene size increases. To this end, we devise a fully-convolutional generative 3D CNN model whose filter kernels are invariant to the overall scene size. The model can be trained on scene subvolumes but deployed on arbitrarily large scenes at test time. In addition, we propose a coarse-to-fine inference strategy in order to produce high-resolution output while also leveraging large input context sizes. In an extensive series of experiments, we carefully evaluate different model design choices, considering both deterministic and probabilistic models for completion and semantic inference. Our results show that we outperform other methods not only in the size of the environments handled and processing efficiency, but also with regard to completion quality and semantic segmentation performance by a significant margin.

1. Introduction

With the wide availability of commodity RGB-D sensors such as Microsoft Kinect, Intel RealSense, and Google Tango, 3D reconstruction of indoor spaces has gained momentum [21, 10, 23, 41, 5]. 3D reconstructions can help create content for graphics applications, and virtual and augmented reality applications rely on obtaining high-quality 3D models from the surrounding environments. Although significant progress has been made in tracking accuracy and efficient data structures for scanning large spaces, the resulting reconstructed 3D model quality remains unsatisfactory.

One fundamental limitation in quality is that, in general, one can only obtain partial and incomplete reconstructions of a given scene, as scans suffer from occlusions and the physical limitations of range sensors. In practice, even with careful scanning by human experts, it is virtually impossible to scan a room without holes in the reconstruction. Holes are both aesthetically displeasing and can lead to severe problems in downstream processing, such as 3D printing or scene editing, as it is unclear whether certain areas of the scan represent free space or occupied space. Traditional approaches, such as Laplacian hole filling [35, 20, 43] or Poisson Surface reconstruction [12, 13] can fill small holes. However, completing high-level scene geometry, such as

missing walls or chair legs, is much more challenging.

One promising direction towards solving this problem is to use machine learning for completion. Very recently, deep learning approaches for 3D completion and other generative tasks involving a single object or depth frame have shown promising results [28, 38, 9, 8, 6]. However, generative modeling and structured output prediction in 3D remains challenging. When represented with volumetric grids, data size grows cubically as the size of the space increases, which severely limits resolution. Indoor scenes are particularly challenging, as they are not only large but can also be irregularly shaped with varying spatial extents.

In this paper, we propose a novel approach, ScanComplete, that operates on large 3D environments without restrictions on spatial extent. We leverage fully-convolutional neural networks that can be trained on smaller subvolumes but applied to arbitrarily-sized scene environments at test time. This ability allows efficient processing of 3D scans of very large indoor scenes: we show examples with bounds of up to $1480 \times 1230 \times 64$ voxels ($\approx 70 \times 60 \times 3$ m). We specifically focus on the tasks of scene completion and semantic inference: for a given partial input scan, we infer missing geometry and predict semantic labels on a per-voxel basis. To obtain high-quality output, the model must use a sufficiently high resolution to predict fine-scale detail. However, it must also consider a sufficiently large context to recognize large structures and maintain global consistency. To reconcile these competing concerns, we propose a coarse-to-fine strategy in which the model predicts a multi-resolution hierarchy of outputs. The first hierarchy level predicts scene geometry and semantics at low resolution but large spatial context. Following levels use a smaller spatial context but higher resolution, and take the output of the previous hierarchy level as input in order to leverage global context.

In our evaluations, we show scene completion and semantic labeling at unprecedented spatial extents. In addition, we demonstrate that it is possible to train our model on synthetic data and transfer it to completion of real RGB-D scans taken from commodity scanning devices. Our results outperform existing completion methods and obtain significantly higher accuracy for semantic voxel labeling.

In summary, our contributions are

- 3D fully-convolutional completion networks for processing 3D scenes with arbitrary spatial extents.
- A coarse-to-fine completion strategy which captures both local detail and global structure.
- Scene completion and semantic labeling, both of outperforming existing methods by significant margins.

2. Related Work

3D Shape and Scene Completion Completing 3D shapes has a long history in geometry processing and is often applied as a post-process to raw, captured 3D data. Traditional

methods typically focus on filling small holes by fitting local surface primitives such as planes or quadrics, or by using a continuous energy minimization [35, 20, 43]. Many surface reconstruction methods that take point cloud inputs can be seen as such an approach, as they aim to fit a surface and treat the observations as data points in the optimization process; e.g., Poisson Surface Reconstruction [12, 13].

Other shape completion methods have been developed, including approaches that leverage symmetries in meshes or point clouds [39, 18, 25, 33, 36] or part-based structural priors derived from a database [37]. One can also ‘complete’ shapes by replacing scanned geometry with aligned CAD models retrieved from a database [19, 31, 14, 16, 32]. Such approaches assume exact database matches for objects in the 3D scans, though this assumption can be relaxed by allowing modification of the retrieved models, e.g., by non-rigid registration such that they better fit the scan [24, 30].

To generalize to entirely new shapes, data-driven structured prediction methods show promising results. One of the first such methods is Voxlets [7], which uses a random decision forest to predict unknown voxel neighborhoods.

Deep Learning in 3D With the recent popularity of deep learning methods, several approaches for shape generation and completion have been proposed. 3D ShapeNets [2] learns a 3D convolutional deep belief network from a shape database. This network can generate and complete shapes, and also repair broken meshes [22].

Several other works have followed, using 3D convolutional neural networks (CNNs) for object classification [17, 26] or completion [6, 8]. To more efficiently represent and process 3D volumes, hierarchical 3D CNNs have been proposed [29, 40]. The same hierarchical strategy can be also used for generative approaches which output higher-resolution 3D models [28, 38, 9, 8]. One can also increase the spatial extent of a 3D CNN with dilated convolutions [42]. This approach has recently been used for predicting missing voxels and semantic inference [34]. However, these methods operate on a fixed-sized volume whose extent is determined at training time. Hence, they focus on processing either a single object or a single depth frame. In our work, we address this limitation with our new approach, which is invariant to differing spatial extent between train and test, thus allowing processing of large scenes at test time while maintaining a high voxel resolution.

3. Method Overview

Our ScanComplete method takes as input a partial 3D scan, represented by a truncated signed distance field (TSDF) stored in a volumetric grid. The TSDF is generated from depth frames following the volumetric fusion approach of Curless and Levoy [3], which has been widely adopted by modern RGB-D scanning methods [21, 10, 23,

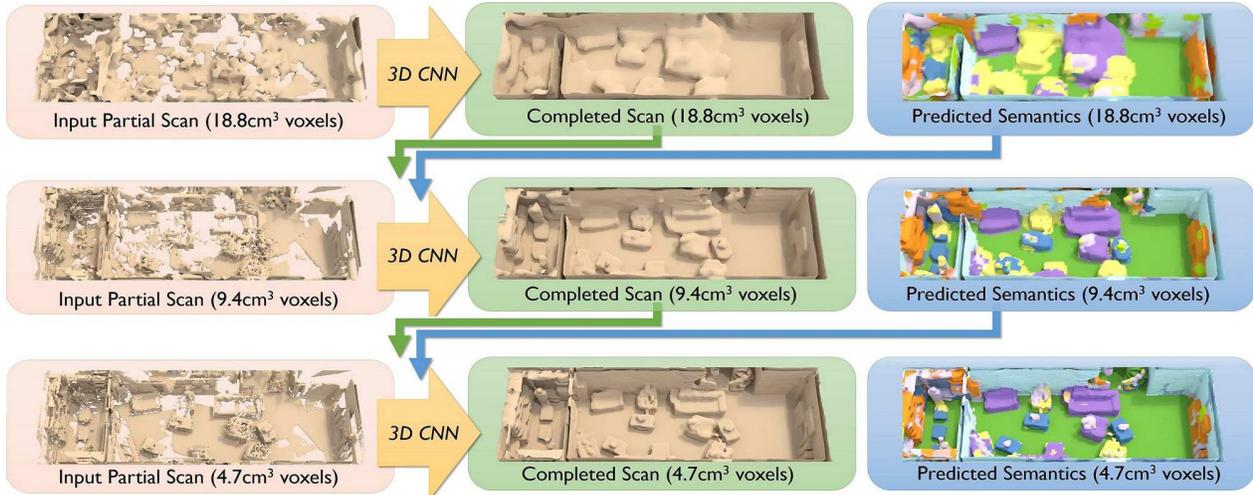


Figure 1. Overview of our method: we propose a hierarchical coarse-to-fine approach, where each level takes a partial 3D scan as input, and predicts a completed scan as well as per-voxel semantic labels at the respective level’s voxel resolution using our autoregressive 3D CNN architecture (see Fig. 3). The next hierarchy level takes as input the output of the previous levels (both completion and semantics), and is then able to refine the results. This process allows leveraging a large spatial context while operating on a high local voxel resolution. In the final result, we see both global completion, as well as local surface detail and high-resolution semantic labels.

[11, 5]. We feed this partial TSDF into our new volumetric neural network, which outputs a truncated, unsigned distance field (TDF). At train time, we provide the network with a target TDF, which is generated from a complete ground-truth mesh. The network is trained to output a TDF which is as similar as possible to this target complete TDF.

Our network uses a fully-convolutional architecture with three-dimensional filter banks. Its key property is its invariance to input spatial extent, which is particularly critical for completing large 3D scenes whose sizes can vary significantly. That is, we can train the network using random spatial crops sampled from training scenes, and then test on different spatial extents at test time.

The memory requirements of a volumetric grid grow cubically with spatial extent, which limits manageable resolutions. Small voxel sizes capture local detail but lack spatial context; large voxel sizes provide large spatial context but lack local detail. To get the best of both worlds while maintaining high resolution, we use a coarse-to-fine hierarchical strategy. Our network first predicts the output at a low resolution in order to leverage more global information from the input. Subsequent hierarchy levels operate at a higher resolution and smaller context size. They condition on the previous level’s output in addition to the current-level incomplete TSDF. We use three hierarchy levels, with a large context of several meters ($\sim 6\text{m}^3$) at the coarsest level, up to a fine-scale voxel resolution of $\sim 5\text{cm}^3$; see Fig. 1.

Our network uses an autoregressive architecture based on that of Reed et al. [27]. We divide the volumetric space of a given hierarchy level into a set of eight voxel groups, such that voxels from the same group do not neighbor each other; see Fig. 2. The network predicts all voxels in group one, followed by all voxels in group two, and so on. The prediction for each group is conditioned on the predictions

for the groups that precede it. Thus, we use eight separate networks, one for each voxel group; see Fig. 2.

We also explore multiple options for the training loss function which penalizes differences between the network output and the ground truth target TDF. As one option, we use a deterministic ℓ_1 -distance, which forces the network to focus on a single mode. This setup is ideal when partial scans contain enough context to allow for a single explanation of the missing geometry. As another option, we use a probabilistic model formulated as a classification problem, i.e., TDF values are discretized into bins and their probabilities are weighted based on the magnitude of the TDF value. This setup may be better suited for very sparse inputs, as the predictions can be multi-modal.

In addition to predicting complete geometry, the model jointly predicts semantic labels on a per-voxel basis. The semantic label prediction also leverages the fully-convolution autoregressive architecture as well as the coarse-to-fine prediction strategy to obtain an accurate semantic segmentation of the scene. In our results, we demonstrate how completion greatly helps semantic inference.

4. Data Generation

To train our ScanComplete CNN architecture, we prepare training pairs of partial TSDF scans and their complete TDF counterparts. We generate training examples from SUNCG [34], using 5359 train scenes and 155 test scenes from the train-test split from prior work [34]. As our network requires only depth input, we virtually scan depth data by generating scanning trajectories mimicking real-world scanning paths. To do this, we extract trajectory statistics from the ScanNet dataset [4] and compute the mean and variance of camera heights above the ground as well as the

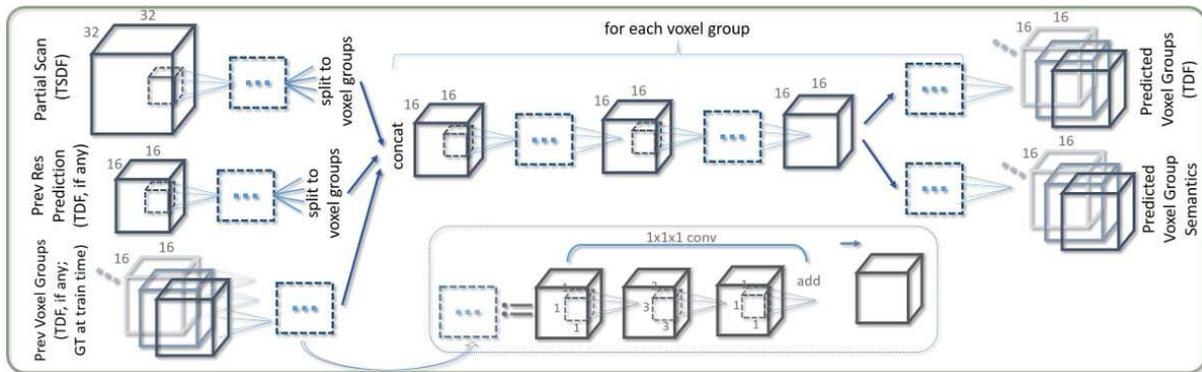


Figure 3. Our ScanComplete network architecture for a single hierarchy level. We take as input a TSDF partial scan, and autoregressively predict both the completed geometry and semantic segmentation. Our network trains for all eight voxel groups in parallel, as we use ground truth for previous voxel groups at train time. In addition to input from the current hierarchy level, the network takes the predictions (TDF and semantics) from the previous level (i.e., next coarser resolution as input), if available; cf. Fig. 1.

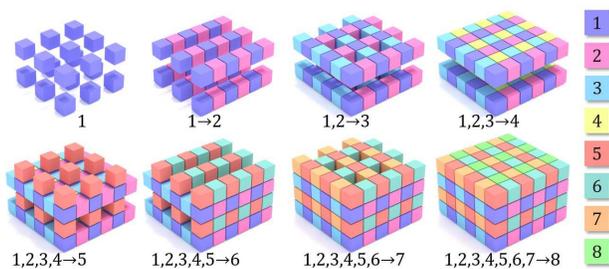


Figure 2. Our model divides volumetric space into eight interleaved voxel groups, such that voxels from the same group do not neighbor each other. It then predicts the contents of these voxel groups autoregressively, predicting voxel group i conditioned on the predictions for groups $1 \dots i - 1$. This approach is based on prior work in autoregressive image modeling [27].

camera angle between the look and world-up vectors. For each room in a SUNCG scene, we then sample from this distribution to select a camera height and angle.

Within each 1.5m^3 region in a room, we select one camera to add to the training scanning trajectory. We choose the camera c whose resulting depth image $D(c)$ is most similar to depth images from ScanNet. To quantify this similarity, we first compute the histogram of depth of values $H(D(c))$ for all cameras in ScanNet, and then compute the average histogram, \bar{H} . We then compute the Earth Mover’s Distance between histograms for all cameras in ScanNet and \bar{H} , i.e., $\text{EMD}(H(D(c)), \bar{H})$ for all cameras c in ScanNet. We take the mean μ_{EMD} and variance σ_{EMD}^2 of these distance values. This gives us a Gaussian distribution over distances to the average depth histogram that we expect to see in real scanning trajectories. For each candidate camera c , we compute its probability under this distribution, i.e., $\mathcal{N}(\text{EMD}(H(D(c)), \bar{H}), \mu_{\text{EMD}}, \sigma_{\text{EMD}})$. We take a linear combination of this term with the percentage of pixels in $D(c)$ which cover scene objects (i.e., not floor, ceiling, or wall), reflecting the assumption that people tend to focus scans on interesting objects rather than pointing a depth sensor directly at the ground or a wall. The highest-scoring camera c^* under this combined objective is added to the

training scanning trajectory. This way, we encourage a realistic scanning trajectory, which we use for rendering virtual views from the SUNCG scenes.

For rendered views, we store per-pixel depth in meters. We then volumetrically fuse [3] the data into a dense regular grid, where each voxel stores a truncated signed distance value. We set the truncation to $3 \times$ the voxel size, and we store TSDF values in voxel-distance metrics. We repeat this process independently for three hierarchy levels, with voxel sizes of 4.7cm^3 , 9.4cm^3 , and 18.8cm^3 .

We generate target TDFs for training using complete meshes from SUNCG. To do this, we employ the level set generation toolkit by Batty [1]. For each voxel, we store a truncated distance value (no sign; truncation of $3 \times$ voxel size), as well as a semantic label of the closest object to the voxel center. As with TSDFs, TDF values are stored in voxel-distance metrics, and we repeat this ground truth data generation for each of the three hierarchy levels.

For training, we uniformly sample subvolumes at 3m intervals out of each of the train scenes. We keep all subvolumes containing any non-structural object voxels (e.g., tables, chairs), and randomly discard subvolumes that contain only structural voxels (i.e., wall/ceiling/floor) with 90% probability. This results in a total of 225, 414 training subvolumes. We use voxel grid resolutions of $[32 \times 16 \times 32]$, $[32 \times 32 \times 32]$, and $[32 \times 64 \times 32]$ for each level, resulting in spatial extents of $[6\text{m} \times 3\text{m} \times 6\text{m}]$, $[3\text{m}^3]$, $[1.5\text{m} \times 3\text{m} \times 1.5\text{m}]$, respectively. For testing, we test on entire scenes. Both the input partial TSDF and complete target TDF are stored as uniform grids spanning the full extent of the scene, which varies across the test set. Our fully-convolutional architecture allows training and testing on different sizes and supports varying training spatial extents.

Note that the sign of the input TSDF encodes known and unknown space according to camera visibility, i.e., voxels with a negative value lie behind an observed surface and are thus unknown. In contrast, we use an unsigned distance field (TDF) for the ground truth target volume, since all vox-

els are known in the ground truth. One could argue that the target distance field should use a sign to represent space inside objects. However, this is infeasible in practice, since the synthetic 3D models from which the ground truth distance fields are generated are rarely watertight. The use of implicit functions (TSDF and TDF) rather than a discrete occupancy grid allows for better gradients in the training process; this is demonstrated by a variety of experiments on different types of grid representations in prior work [6].

5. ScanComplete Network Architecture

Our ScanComplete network architecture for a single hierarchy level is shown in Fig. 3. It is a fully-convolutional architecture operating directly in 3D, which makes it invariant to different training and testing input data sizes.

At each hierarchy level, the network takes the input partial scan as input (encoded as an TSDF in a volumetric grid) as well as the previous low-resolution TDF prediction (if not the base level) and any previous voxel group TDF predictions. Each of the input volumes is processed with a series of 3D convolutions with $1 \times 1 \times 1$ convolution shortcuts. They are then all concatenated feature-wise and further processed with 3D convolutions with shortcuts. At the end, the network splits into two paths, one outputting the geometric completion, and the other outputting semantic segmentation, which are measured with an ℓ_1 loss and voxel-wise softmax cross entropy, respectively. An overview of the architectures between hierarchy levels is shown in Fig. 1.

5.1. Training

To train our networks, we use the training data generated from the SUNCG dataset as described in Sec. 4.

At train time, we feed ground truth volumes as the previous voxel group inputs to the network. For the previous hierarchy level input, however, we feed in volumes predicted by the previous hierarchy level network. Initially, we trained on ground-truth volumes here, but found that this tended to produce highly over-smoothed final output volumes. We hypothesize that the network learned to rely heavily on sharp details in the ground truth volumes that are sometimes not present in the predicted volumes, as the network predictions cannot perfectly recover such details and tend to introduce some smoothing. By using previous hierarchy level predicted volumes as input instead, the network must learn to use the current-level partial input scan to resolve details, relying on the previous level input only for more global, lower-frequency information (such as how to fill in large holes in walls and floors). The one downside to this approach is that the networks for each hierarchy level can no longer be trained in parallel. They must be trained sequentially, as the networks for each hierarchy level depend on output predictions from the trained networks at the previous level. Ideally, we would train all hierarchy levels

in a single, end-to-end procedure. However, current GPU memory limitations make this intractable.

Since we train our model on synthetic data, we introduce height jittering for training samples to counter overfitting, jittering every training sample in height by a (uniform) random jitter in the range $[0, 0.1875]$ m. Since our training data is skewed towards walls and floors, we apply re-weighting in the semantic loss, using a 1:10 ratio for structural classes (e.g. wall/floor/ceiling) versus all other object classes.

For our final model, we train all networks on a NVIDIA GTX 1080, using the Adam optimizer [15] with learning rate 0.001 (decayed to 0.0001) We train one network for each of the eight voxel groups at each of the three hierarchy levels, for a total of 24 trained networks. Note that the eight networks within each hierarchy level are trained in parallel, with a total training time for the full hierarchy of ~ 3 days.

6. Results and Evaluation

Completion Evaluation on SUNCG We first evaluate different architecture variants for geometric scene completion in Tab. 1. We test on 155 SUNCG test scenes, varying the following architectural design choices:

- **Hierarchy Levels:** our three-level hierarchy (3) vs. a single 4.7cm-only level (1). For the three-level hierarchy, we compare training on ground truth volumes (*gt train*) vs. predicted volumes (*pred. train*) from the previous hierarchy level.
- **Probabilistic/Deterministic:** a probabilistic model (*prob.*) that outputs per-voxel a discrete distribution over some number of quantized distance value bins (*#quant*) vs. a deterministic model that outputs a single distance value per voxel (*det.*).
- **Autoregressive:** our autoregressive model that predicts eight interleaved voxel groups in sequence (*autoreg.*) vs. a non-autoregressive variant that predicts all voxels independently (*non-autoreg.*).
- **Input Size:** the width and depth of the input context at train time, using either 16 or 32 voxels

We measure completion quality using ℓ_1 distances with respect to the entire target volume (*entire*), predicted surface (*pred. surf.*), target surface (*target surf.*), and unknown space (*unk. space*). Using only a single hierarchy level, an autoregressive model improves upon a non-autoregressive model, and reducing the number of quantization bins from 256 to 32 improves completion (further reduction reduces the discrete distribution’s ability to approximate a continuous distance field). Note that the increase in *pred. surf.* error from the hierarchy is tied to the ability to predict more unknown surface, as seen by the decrease in *unk. space* error. Moreover, for our scene completion task, a deterministic model performs better than a probabilistic one, as intu-

Hierarchy Levels	Probabilistic/ Deterministic	Autoregressive	Input Size	ℓ_1 -Err (entire)	ℓ_1 -Err (pred. surf.)	ℓ_1 -Err (target surf.)	ℓ_1 -Err (unk. space)
1	prob. (#quant=256)	non-autoreg.	32	0.248	0.311	0.969	0.324
1	prob. (#quant=256)	autoreg.	16	0.226	0.243	0.921	0.290
1	prob. (#quant=256)	autoreg.	32	0.218	0.269	0.860	0.283
1	prob. (#quant=32)	autoreg.	32	0.208	0.252	0.839	0.271
1	prob. (#quant=16)	autoreg.	32	0.212	0.325	0.818	0.272
1	prob. (#quant=8)	autoreg.	32	0.226	0.408	0.832	0.284
1	det.	non-autoreg.	32	0.248	0.532	0.717	0.330
1	det.	autoreg.	16	0.217	0.349	0.808	0.282
1	det.	autoreg.	32	0.204	0.284	0.780	0.266
3 (gt train)	prob. (#quant=32)	autoreg.	32	0.336	0.840	0.902	0.359
3 (pred. train)	prob. (#quant=32)	autoreg.	32	0.202	0.405	0.673	0.251
3 (gt train)	det.	autoreg.	32	0.303	0.730	0.791	0.318
3 (pred. train)	det.	autoreg.	32	0.182	0.419	0.534	0.225

Table 1. Quantitative scene completion results for different variants of our completion-only model evaluated on synthetic SUNCG ground truth data. We measure the ℓ_1 error against the ground truth distance field (in voxel space, up to truncation distance of 3 voxels). Using an autoregressive model with a three-level hierarchy and large input context size gives the best performance.

Method	ℓ_1 -Err (entire)	ℓ_1 -Err (pred. surf.)	ℓ_1 -Err (target surf.)	ℓ_1 -Err (unk. space)
Poisson Surface Reconstruction [12, 13]	0.531	1.178	1.695	0.512
SSCNet [34]	0.536	1.106	0.931	0.527
3D-EPN (unet) [6]	0.245	0.467	0.650	0.302
Ours (completion + semantics)	0.202	0.462	0.569	0.248
Ours (completion only)	0.182	0.419	0.534	0.225

Table 2. Quantitative scene completion results for different methods on synthetic SUNCG data. We measure the ℓ_1 error against the ground truth distance field in voxel space, up to truncation distance of 3 voxels (i.e., 1 voxel corresponds to 4.7cm^3). Our method outperforms others in reconstruction error.

itively we aim to capture a single output mode—the physical reality behind the captured 3D scan. An autoregressive, deterministic, full hierarchy with the largest spatial context provides the highest accuracy.

We also compare our method to alternative scene completion methods in Tab. 2. As a baseline, we compare to Poisson Surface Reconstruction [12, 13]. We also compare to 3D-EPN, which was designed for completing single objects, as opposed to scenes [6]. Additionally, we compare to SSCNet, which completes the subvolume of a scene viewed by a single depth frame [34]. For this last comparison, in order to complete the entire scene, we fuse the predictions from all cameras of a test scene into one volume, then evaluate ℓ_1 errors over this entire volume. Our method achieves lower reconstruction error than all the other methods. Note that while jointly predicting semantics along with completion does not improve on completion, Tab. 3 shows that it significantly improves semantic segmentation performance.

We show a qualitative comparison of our completion against state-of-the-art methods in Fig. 4. For these results, we use the best performing architecture according to Tab. 1. We can run our method on arbitrarily large scenes as test input, thus predicting missing geometry in large areas even when input scans are highly partial, and producing more complete results as well as more accurate local de-

tail. Note that our method is $\mathcal{O}(1)$ at test time in terms of forward passes; we run more efficiently than previous methods which operate on fixed-size subvolumes and must iteratively make predictions on subvolumes of a scene, typically $\mathcal{O}(wd)$ for a $w \times h \times d$ scene.

Completion Results on ScanNet (real data) We also show qualitative completion results on real-world scans in Fig. 6. We run our model on scans from the publicly-available RGB-D ScanNet dataset [4], which has data captured with an Occipital Structure Sensor, similar to a Microsoft Kinect or Intel PrimeSense sensor. Again, we use the best performing network according to Tab. 1. We see that our model, trained only on synthetic data, learns to generalize and transfer to real data.

Semantic Inference on SUNCG In Tab. 3, we evaluate and compare our semantic segmentation on the SUNCG dataset. All methods were trained on the train set of scenes used by SSCNet [34] and evaluated on the test set. We use the SUNCG 11-label set. Our semantic inference benefits significantly from the joint completion and semantic task, significantly outperforming current state of the art.

Fig. 5 shows qualitative semantic segmentation results on SUNCG scenes. Our ability to process the entire scene at test time, in contrast to previous methods which operate

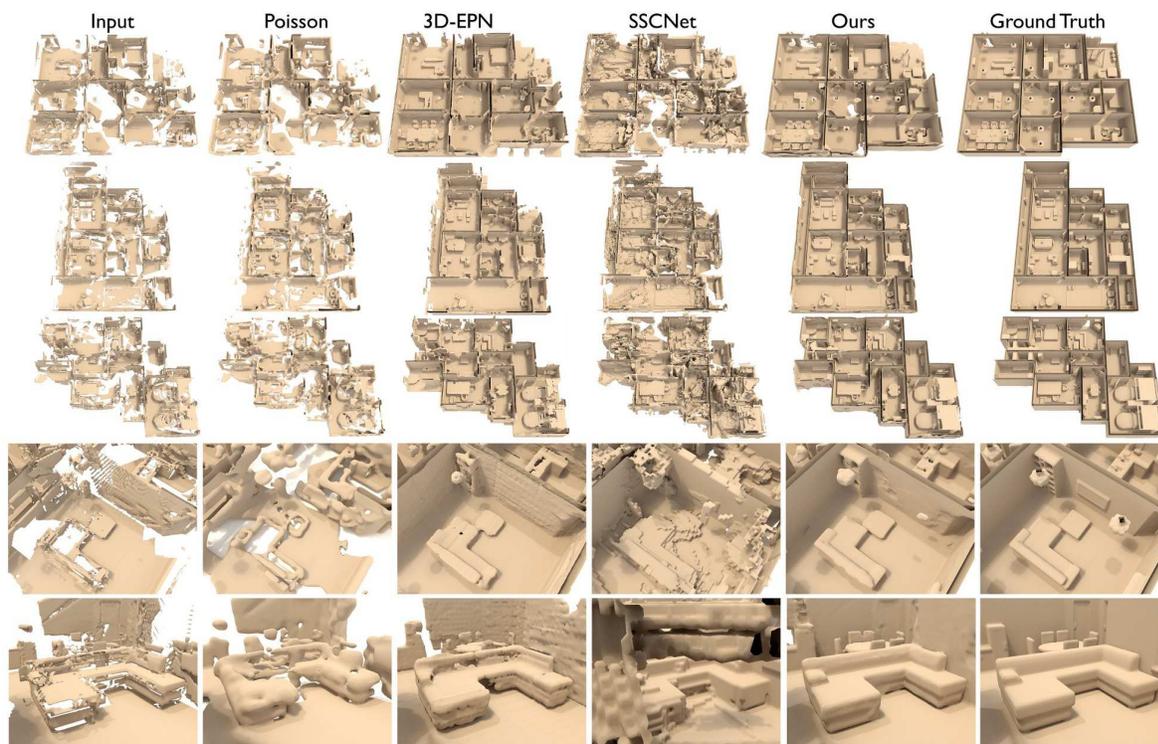


Figure 4. Completion results on synthetic SUNCG scenes; left to right: input, Poisson Surface Reconstruction [13], 3D-EPN [6], SSCNet [34], Ours, ground truth.

	bed	ceiling	chair	floor	furn.	obj.	sofa	table	tv	wall	wind.	avg
(vis) ScanNet [4]	44.8	90.1	32.5	75.2	41.3	25.4	51.3	42.4	9.1	60.5	4.5	43.4
(vis) SSCNet [34]	67.4	95.8	41.6	90.2	42.5	40.7	50.8	58.4	20.2	59.3	49.7	56.1
(vis) Ours [sem-only, no hier]	63.6	92.9	41.2	58.0	27.2	19.6	55.5	49.0	9.0	58.3	5.1	43.6
(is) Ours [sem-only]	82.9	96.1	48.2	67.5	64.5	40.8	80.6	61.7	14.8	69.1	13.7	58.2
(vis) Ours [no hier]	70.3	97.6	58.9	63.0	46.6	34.1	74.5	66.5	40.9	86.5	43.1	62.0
(vis) Ours	80.1	97.8	63.4	94.3	59.8	51.2	77.6	65.4	32.4	84.1	48.3	68.6
(int) SSCNet [34]	65.6	81.2	48.2	76.4	49.5	49.8	61.1	57.4	14.4	74.0	36.6	55.8
(int) Ours [no hier]	68.6	96.9	55.4	71.6	43.5	36.3	75.4	68.2	33.0	88.4	33.1	60.9
(int) Ours	82.3	97.1	60.0	93.2	58.0	51.6	80.6	66.1	26.8	86.9	37.3	67.3

Table 3. Semantic labeling accuracy on SUNCG scenes. We measure per-voxel class accuracies for both the voxels originally visible in the input partial scan (*vis*) as well as the voxels in the intersection of our predictions, SSCNet, and ground truth (*int*). Note that we show significant improvement over a semantic-only model that does not perform completion (*sem-only*) as well as the current state-of-the-art.

on fixed subvolumes, along with the autoregressive, joint completion task, produces more globally consistent and accurate voxel labels.

For semantic inference on real scans, we refer to the appendix.

7. Conclusion and Future Work

In this paper, we have presented ScanComplete, a novel data-driven approach that takes an input partial 3D scan and predicts both completed geometry and semantic voxel labels for the entire scene at once. The key idea is to use a fully-convolutional network that decouples train and test resolutions, thus allowing for variably-sized test scenes with unbounded spatial extents. In addition, we use a coarse-

to-fine prediction strategy combined with a volumetric autoregressive network that leverages large spatial contexts while simultaneously predicting local detail. As a result, we achieve both unprecedented scene completion results as well as volumetric semantic segmentation with significantly higher accuracy than previous state of the art.

Our work is only a starting point for obtaining high-quality 3D scans from partial inputs, which is a typical problem for RGB-D reconstructions. One important aspect for future work is to further improve output resolution. Currently, our final output resolution of $\sim 5\text{cm}^3$ voxels is still not enough—ideally, we would use even higher resolutions in order to resolve fine-scale objects, e.g., cups. In addition, we believe that end-to-end training across all hierarchy lev-

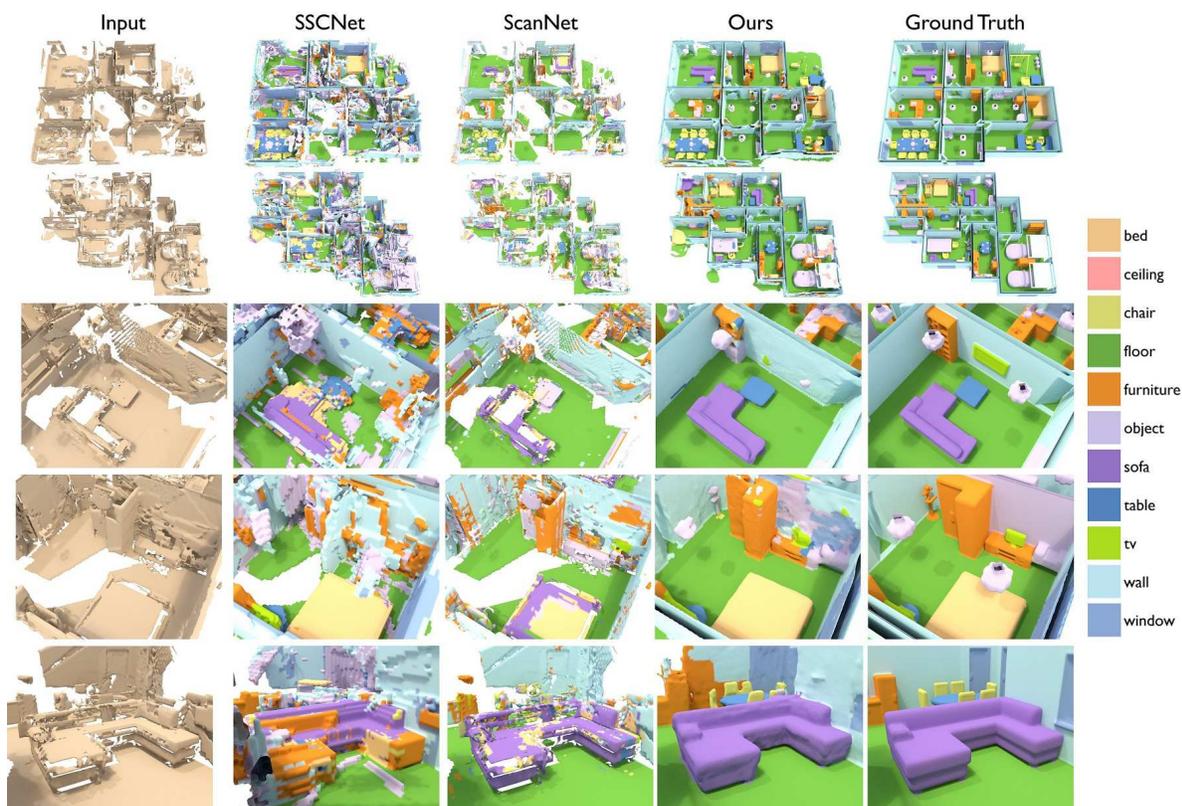


Figure 5. Semantic voxel labeling results on SUNCG; from left to right: input, SSCNet [34], ScanNet [4], Ours, and ground truth.

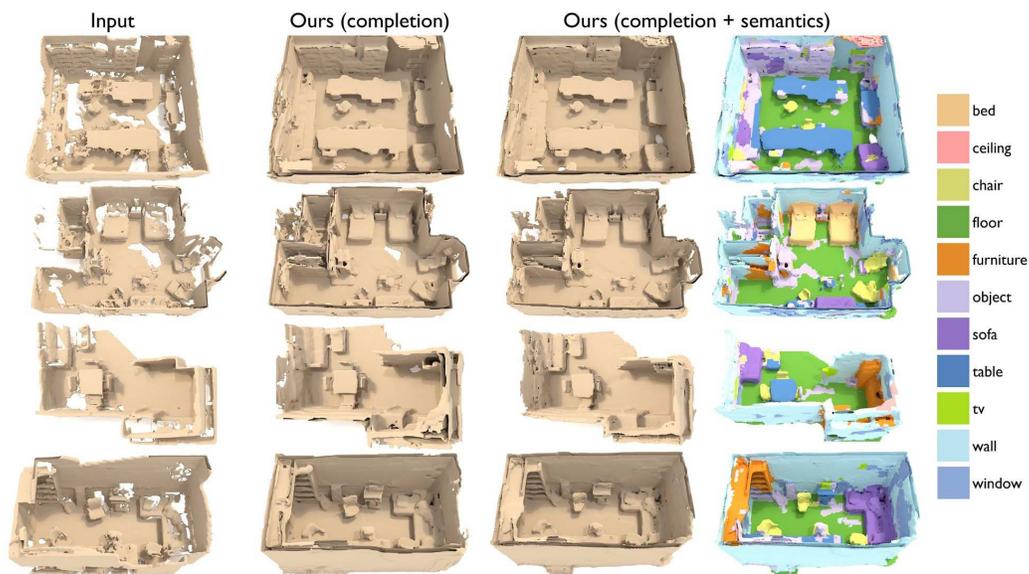


Figure 6. Completion results on real-world scans from ScanNet [4]. Despite being trained only on synthetic data, our model is also able to complete many missing regions of real-world data.

els would further improve performance with the right joint optimization strategy. Nonetheless, we believe that we have set an important baseline for completing entire scenes. We hope that the community further engages in this exciting task, and we are convinced that we will see many improvements along these directions.

Acknowledgments

This work was supported by a Google Research Grant, a Stanford Graduate Fellowship, and a TUM-IAS Rudolf Mößbauer Fellowship. We would also like to thank Shuran Song for helping with the SSCNet comparison.

References

- [1] C. Batty. SDFGen. <https://github.com/christopherbatty/SDFGen>. 4
- [2] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015. 2
- [3] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312. ACM, 1996. 2, 4
- [4] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. 3, 6, 7, 8
- [5] A. Dai, M. Nießner, M. Zollhöfer, S. Izadi, and C. Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics (TOG)*, 36(3):24, 2017. 1, 3
- [6] A. Dai, C. R. Qi, and M. Nießner. Shape completion using 3d-encoder-predictor cnns and shape synthesis. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. 2, 5, 6, 7
- [7] M. Firman, O. Mac Aodha, S. Julier, and G. J. Brostow. Structured prediction of unobserved voxels from a single depth image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5431–5440, 2016. 2
- [8] X. Han, Z. Li, H. Huang, E. Kalogerakis, and Y. Yu. High Resolution Shape Completion Using Deep Neural Networks for Global Structure and Local Geometry Inference. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 2
- [9] C. Häne, S. Tulsiani, and J. Malik. Hierarchical surface prediction for 3d object reconstruction. *arXiv preprint arXiv:1704.00710*, 2017. 2
- [10] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568. ACM, 2011. 1, 3
- [11] O. Kähler, V. A. Prisacariu, C. Y. Ren, X. Sun, P. Torr, and D. Murray. Very high frame rate volumetric integration of depth images on mobile devices. *IEEE transactions on visualization and computer graphics*, 21(11):1241–1250, 2015. 3
- [12] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, volume 7, 2006. 1, 2, 6
- [13] M. Kazhdan and H. Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics (TOG)*, 32(3):29, 2013. 1, 2, 6, 7
- [14] Y. M. Kim, N. J. Mitra, D.-M. Yan, and L. Guibas. Acquiring 3d indoor environments with variability and repetition. *ACM Transactions on Graphics (TOG)*, 31(6):138, 2012. 2
- [15] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5
- [16] Y. Li, A. Dai, L. Guibas, and M. Nießner. Database-assisted object retrieval for real-time 3d reconstruction. In *Computer Graphics Forum*, volume 34, pages 435–446. Wiley Online Library, 2015. 2
- [17] D. Maturana and S. Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 922–928. IEEE, 2015. 2
- [18] N. J. Mitra, L. J. Guibas, and M. Pauly. Partial and approximate symmetry detection for 3d geometry. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 560–568. ACM, 2006. 2
- [19] L. Nan, K. Xie, and A. Sharf. A search-classify approach for cluttered indoor scene understanding. *ACM Transactions on Graphics (TOG)*, 31(6):137, 2012. 2
- [20] A. Nealen, T. Igarashi, O. Sorkine, and M. Alexa. Laplacian mesh optimization. In *Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia*, pages 381–389. ACM, 2006. 1, 2
- [21] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, pages 127–136. IEEE, 2011. 1, 3
- [22] D. T. Nguyen, B.-S. Hua, M.-K. Tran, Q.-H. Pham, and S.-K. Yeung. A field model for repairing 3d shapes. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 5, 2016. 2
- [23] M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger. Real-time 3d reconstruction at scale using voxel hashing. *ACM Transactions on Graphics (TOG)*, 2013. 1, 3
- [24] M. Pauly, N. J. Mitra, J. Giesen, M. H. Gross, and L. J. Guibas. Example-based 3d scan completion. In *Symposium on Geometry Processing*, number EPFL-CONF-149337, pages 23–32, 2005. 2
- [25] M. Pauly, N. J. Mitra, J. Wallner, H. Pottmann, and L. J. Guibas. Discovering structural regularity in 3d geometry. In *ACM transactions on graphics (TOG)*, volume 27, page 43. ACM, 2008. 2
- [26] C. R. Qi, H. Su, M. Nießner, A. Dai, M. Yan, and L. Guibas. Volumetric and multi-view cnns for object classification on 3d data. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2016. 2
- [27] S. E. Reed, A. van den Oord, N. Kalchbrenner, S. Gómez, Z. Wang, D. Belov, and N. de Freitas. Parallel multi-scale autoregressive density estimation. In *Proceedings of The 34th International Conference on Machine Learning (ICML)*, 2017. 3, 4
- [28] G. Riegler, A. O. Ulusoy, H. Bischof, and A. Geiger. Octnetfusion: Learning depth fusion from data. *arXiv preprint arXiv:1704.01047*, 2017. 2

- [29] G. Riegler, A. O. Ulusoy, and A. Geiger. Octnet: Learning deep 3d representations at high resolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2
- [30] J. Rock, T. Gupta, J. Thorsen, J. Gwak, D. Shin, and D. Hoiem. Completing 3d object shape from one depth image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2484–2493, 2015. 2
- [31] T. Shao, W. Xu, K. Zhou, J. Wang, D. Li, and B. Guo. An interactive approach to semantic modeling of indoor scenes with an rgbd camera. *ACM Transactions on Graphics (TOG)*, 31(6):136, 2012. 2
- [32] Y. Shi, P. Long, K. Xu, H. Huang, and Y. Xiong. Data-driven contextual modeling for 3d scene understanding. *Computers & Graphics*, 55:55–67, 2016. 2
- [33] I. Sipiran, R. Gregor, and T. Schreck. Approximate symmetry detection in partial 3d meshes. In *Computer Graphics Forum*, volume 33, pages 131–140. Wiley Online Library, 2014. 2
- [34] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser. Semantic scene completion from a single depth image. *Proceedings of 30th IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2, 3, 6, 7, 8
- [35] O. Sorkine and D. Cohen-Or. Least-squares meshes. In *Shape Modeling Applications, 2004. Proceedings*, pages 191–199. IEEE, 2004. 1, 2
- [36] P. Speciale, M. R. Oswald, A. Cohen, and M. Pollefeys. A symmetry prior for convex variational 3d reconstruction. In *European Conference on Computer Vision*, pages 313–328. Springer, 2016. 2
- [37] M. Sung, V. G. Kim, R. Angst, and L. Guibas. Data-driven structural priors for shape completion. *ACM Transactions on Graphics (TOG)*, 34(6):175, 2015. 2
- [38] M. Tatarchenko, A. Dosovitskiy, and T. Brox. Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. *arXiv preprint arXiv:1703.09438*, 2017. 2
- [39] S. Thrun and B. Wegbreit. Shape from symmetry. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1824–1831. IEEE, 2005. 2
- [40] P.-S. Wang, Y. Liu, Y.-X. Guo, C.-Y. Sun, and X. Tong. O-cnn: Octree-based convolutional neural networks for 3d shape analysis. *ACM Transactions on Graphics (TOG)*, 36(4):72, 2017. 2
- [41] T. Whelan, S. Leutenegger, R. F. Salas-Moreno, B. Glocker, and A. J. Davison. Elasticfusion: Dense slam without a pose graph. *Proc. Robotics: Science and Systems, Rome, Italy*, 2015. 1
- [42] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015. 2
- [43] W. Zhao, S. Gao, and H. Lin. A robust hole-filling algorithm for triangular mesh. *The Visual Computer*, 23(12):987–997, 2007. 1, 2