RevealNet: Seeing Behind Objects in RGB-D Scans

Ji Hou Angela Dai Matthias Nießner
Technical University of Munich

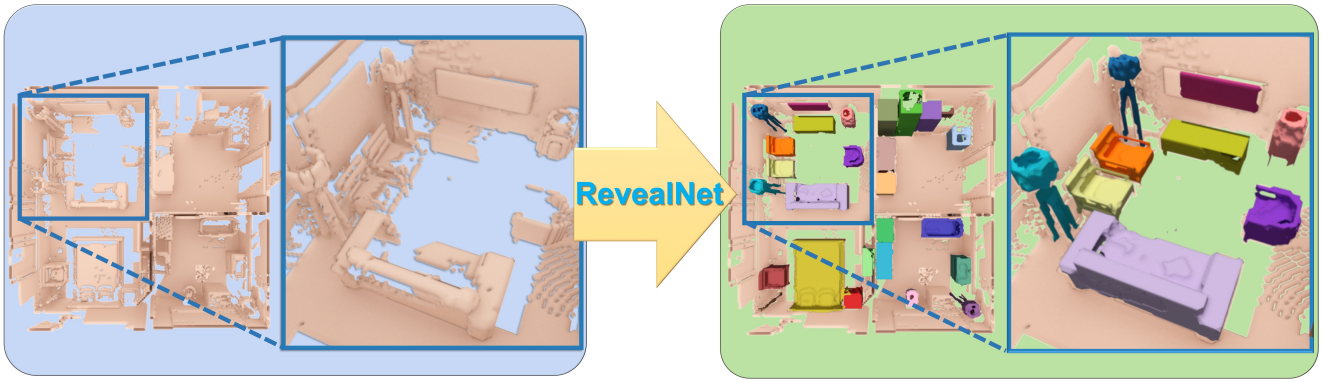

Figure 1: RevealNet takes an RGB-D scan as input and learns to "see behind objects": from the scan's color images and geometry (encoded as a TSDF), objects in the observed scene are detected (as 3D bounding boxes and class labels) and for each object, the complete geometry of that object is predicted as per-instance masks (in both seen and unseen regions).

Abstract

During 3D reconstruction, it is often the case that people cannot scan each individual object from all views, resulting in missing geometry in the captured scan. This missing geometry can be fundamentally limiting for many applications, e.g., a robot needs to know the unseen geometry to perform a precise grasp on an object. Thus, we introduce the task of semantic instance completion: from an incomplete RGB-D scan of a scene, we aim to detect the individual object instances and infer their complete object geometry. This will open up new possibilities for interactions with objects in a scene, for instance for virtual or robotic agents. We tackle this problem by introducing RevealNet, a new data-driven approach that jointly detects object instances and predicts their complete geometry. This enables a semantically meaningful decomposition of a scanned scene into individual, complete 3D objects, including hidden and unobserved object parts. RevealNet is an end-to-end 3D neural network architecture that leverages joint color and geometry feature learning. The fully-convolutional nature of our 3D network enables efficient inference of semantic instance completion for 3D scans at scale of large indoor environments in a single forward pass. We show that predicting complete object geometry improves both 3D detection and instance segmentation performance. We evaluate on both real and synthetic scan benchmark data for the new task, where we outperform state-of-the-art approaches by over 15 *in mAP@0.5 on ScanNet, and over 18 in mAP@0.5 on SUNCG.*

1. Introduction

Understanding 3D environments is fundamental to many tasks spanning computer vision, graphics, and robotics. In particular, in order to effectively navigate, and moreover interact with an environment, an understanding of the geometry of a scene and the objects it comprises of is essential. This is in contrast to the partial nature of reconstructed RGB-D scans; e.g., due to sensor occlusions. For instance, for a robot exploring an environment, it needs to infer where objects are as well as what lies behind the objects it sees in order to efficiently navigate or perform tasks like grasping. That is, it needs not only instance-level knowledge of objects in the scene, but to also estimate the missing geometry of these objects. Additionally, for content creation or mixed reality applications, captured scenes must be decomposable

into their complete object components, in order to enable applications such as scene editing or virtual-real object interactions; i.e., it is often insufficient to segment object instances only for observed regions.

Thus, we aim to address this task of "seeing behind objects," which we refer to as *semantic instance completion*: predicting object detection as well as instance-level completion for an input partial 3D scan of a scene. Previous approaches have addressed these tasks independently: 3D instance segmentation segments object instances from the visible surface of a partial scan [43, 14, 46, 45, 18, 26, 23, 8], and 3D scan completion approaches predict the full scene geometry [39, 7], but lack the notion of individual objects. In contrast, our approach focuses on the instance level, as knowledge of instances is essential towards enabling interaction with the objects in an environment.

In addition, the task of semantic instance completion is not only important towards enabling object-level understanding and interaction with 3D environments, but we also show that the prediction of complete object geometry informs the task of semantic instance segmentation. Thus, in order to address the task of semantic instance completion, we propose to consider instance detection and object completion in an end-to-end, fully differentiable fashion.

From an input RGB-D scan of a scene, our RevealNet model sees behind objects to predict each object's complete geometry. First, object bounding boxes are detected and regressed, followed by object classification and then a prediction of complete object geometry. Our approach leverages a unified backbone from which instance detection and object completion are predicted, enabling information to flow from completion to detection. We incorporate features from both color image and 3D geometry of a scanned scene, as well as a fully-convolutional design in order to effectively predict the complete object decomposition of varying-sized scenes. To address the task of semantic instance completion for real-world scans, where ground truth complete geometry is not readily available, we further introduce a new semantic instance completion benchmark for ScanNet [4], leveraging the Scan2CAD [1] annotations to evaluate semantic instance completion (and semantic instance segmentation).

In summary, we present a fully-convolutional, end-to-end 3D CNN formulation to predict 3D instance completion that outperforms state-of-the-art, decoupled approaches to semantic instance completion by 15.8 in mAP@0.5 on real-world scan data, and 18.5 in mAP@0.5 on synthetic data:

- We introduce the task of *semantic instance completion* for 3D scans;
- we propose a novel, end-to-end 3D convolutional network which predicts 3D semantic instance completion as object bounding boxes, class labels, and complete object geometry,
- and we show that semantic instance completion task can benefit semantic instance segmentation and detection performance.

2. Related Work

Object Detection and Instance Segmentation Recent advances in convolutional neural networks have now begun to drive impressive progress in object detection and instance segmentation for 2D images [9, 33, 23, 32, 20, 13, 21]. Combined with the increasing availability of synthetic and real-world 3D data [4, 39, 2], we are now seeing more advances in object detection [37, 38, 31, 30] for 3D. Sliding Shapes [37] predicted 3D object bounding boxes from a depth image, designing handcrafted features to detect objects in a sliding window fashion. Deep Sliding Shapes [38] then extended this approach to leverage learned features for object detection in a single RGB-D frame. Frustum PointNet [31] tackles the problem of object detection for an RGB-D frame by first detecting object in the 2D image before projecting the detected boxes into 3D to produce final refined box predictions. VoteNet [30] propose a reformulation of Hough voting in the context of deep learning through an end-to-end differentiable architecture for 3D detection purpose.

Recently, several approaches have been introduced to perform 3D instance segmentation, applicable to single or multi-frame RGB-D input. Wang et al. [43] introduced SGPN to operate on point clouds by clustering semantic segmentation predictions. Li et al. [46] leverages an object proposal-based approach to predict instance segmentation for a point cloud. Simultaneously, Hou et al. [14] presented an approach leveraging joint color-geometry feature learning for detection and instance segmentation on volumetric data. Lahoud et al. [18] proposes to use multi-task losses to predict instance segmentation. Yang et al. [45] and Liu et al. [22] both use bottom-up methods to predict instance segmentation for a point cloud. Our approach also leverages an anchor-based object proposal mechanism for detection, but we leverage object completion to predict instance completion, as well as show that completing object-level geometry can improve detection and instance segmentation performance on volumetric data.

3D Scan Completion Scan completion of 3D shapes has been a long-studied problem in geometry processing, particularly for cleaning up broken mesh models. In this context, traditional methods have largely focused on filling small holes by locally fitting geometric primitives, or through continuous energy minimization [40, 27, 47]. Surface reconstruction approaches on point cloud inputs [15, 16] can also be applied in this fashion to locally optimize for missing surfaces. Other shape completion approaches leverage priors such as symmetry and structural priors [42, 24, 29,

36, 41], or CAD model retrieval [25, 34, 17, 19, 35] to predict the scan completion.

Recently, methods leveraging generative deep learning have been developed to predict the complete geometry of 3D shapes [44, 6, 11, 12]. Song et al. [39] extended beyond shapes to predicting the voxel occupancy for a single depth frame leveraging the geometric occupancy prediction to achieve improved 3D semantic segmentation. Recently, Dai et al. [7] presented a first approach for data-driven scan completion of full 3D scenes, leveraging a fully-convolutional, autoregressive approach to predict complete geometry along with 3D semantic segmentation. Both Song et al. [39] and Dai et al. [7] show that inferring the complete scan geometry can improve 3D semantic segmentation. With our approach for 3D semantic instance completion, this task not only enables new applications requiring instance-based knowledge of a scene (e.g., virtual or robotic interactions with objects in a scene), but we also show that instance segmentation can benefit from instance completion.

3. Method Overview

Our network takes as input an RGB-D scan, and learns to join together features from both the color images as well as the 3D geometry to inform the semantic instance completion. The architecture is shown in Fig. 2.

The input 3D scan is encoded as a truncated signed distance field (TSDF) in a volumetric grid. To combine this with color information from the RGB images, we first extract 2D features using 2D convolutional layers on the RGB images, which are then back-projected into a 3D volumetric grid, and subsequently merged with geometric features extracted from the geometry. The joint features are then fed into an encoder-decoder backbone, which leverages a series of 3D residual blocks to learn the representation for the task of semantic instance completion. Objects are detected through anchor proposal and bounding box regression; these predicted object boxes are then used to crop and extract features from the backbone encoder to predict the object class label as well as the complete object geometry for each detected object as per-voxel occupancies.

We adopt in total five losses to supervise the learning process illustrated in Fig. 2. Detection contains three losses: (1) objectness using binary cross entropy to indicate that there is an object, (2) box location using a Huber loss to regress the 3D bounding box locations, and (3) classification of the class label loss using cross entropy. Following detection, the completion head contains two losses: per-instance completion loss using binary cross entropy to predict per-voxel occupancies, and a proxy completion loss using binary cross entropy to classify the surface voxels belonging to all objects in the scene.

Our method operates on a unified backbone for detection followed by instance completion, enabling object completion to inform the object detection process; this results in effective 3D detection as well as instance completion. Its fully-convolutional nature enables us to train on cropped chunks of 3D scans but test on a whole scene in a single forward pass, resulting in an efficient decomposition of a scan into a set of complete objects.

4. Network Architecture

From an RGB-D scan input, our network operates on the scan's reconstructed geometry, encoded as a TSDF in a volumetric grid, as well as the color images. To jointly learn from both color and geometry, color features are first extracted in 2D with a 2D semantic segmentation network [28], and then back-projected into 3D to be combined with the TSDF features, similar to [5, 14]. This enables complementary semantic features to be learned from both data modalities. These features are then input to the backbone of our network, which is structured in an encoder-decoder style.

The encoder-decoder backbone is composed of a series of five 3D residual blocks, which generates five volumetric feature maps $\mathbb{F} = \{f_i | i = 1 \dots 5\}$. The encoder results in a reduction of spatial dimension by a factor of 4, and symmetric decoder results in an expansion of spatial dimension by a factor of 4. Skip connections link spatially-corresponding encoder and decoder features. For a more detailed description of the network architecture, we refer to the appendix.

4.1. Color Back-Projection

As raw color data is often of much higher resolution than 3D geometry, to effectively learn from both color and geometry features, we leverage color information by back-projecting 2D CNN features learned from RGB images to 3D, similar to [5, 14]. For each voxel location $v_i = (x, y, z)$ in the 3D volumetric grid, we find its pixel location $p_i = (x, y)$ in 2D views by camera intrinsic and extrinsic matrices. We assign the voxel feature at location v_i with the learned 2D CNN feature vector at p_i. To handle multiple image observations of the same voxel v_i, we apply element-wise view pooling; this also allows our approach to handle a varying number of input images. Note that this back-projection is differentiable, allowing our model to be trained end-to-end and benefit from both RGB and geometric signal.

4.2. Object Detection

For object detection, we predict the bounding box of each detected object as well as the class label. To inform the detection, features are extracted from feature maps F_2 and F_3 of the backbone encoder. We define two set of anchors on these two features maps, $A_s = \{a_i | i = 1 \dots N_s\}$ and

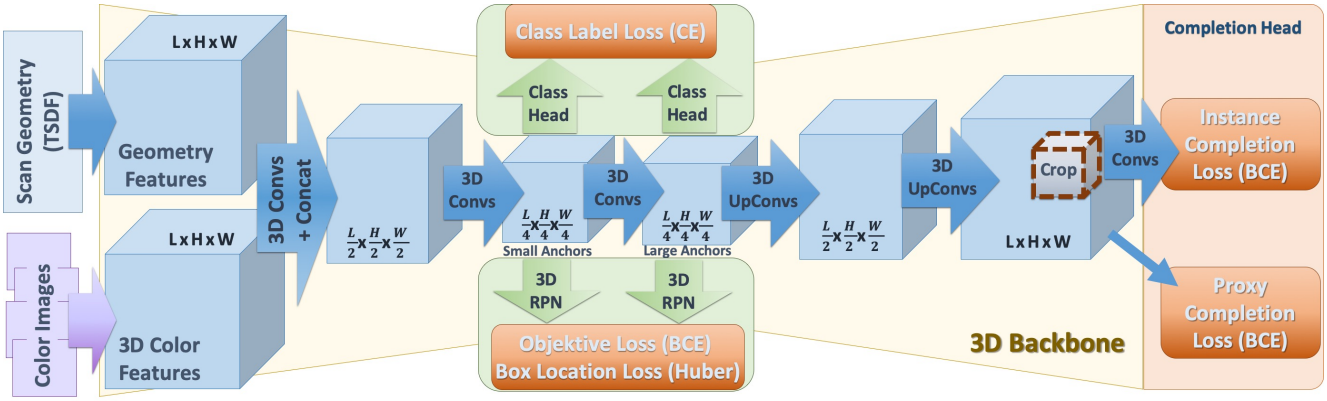

Figure 2: Our RevealNet network architecture takes an RGB-D scan as input. Color images are processed with 2D convolutions to spatially compress the information before back-projecting into 3D, to be merged with the 3D geometry features of the scan (following [5, 14]). These joint features are used for object detection (as 3D bounding boxes and class labels) followed by per-instance geometric completion, for the task of semantic instance completion. In contrast to [14], which leverages separate backbones for detection and instance segmentation, our network maintains one unified backbone for both detection and completion head, allowing the completion task to directly inform the detection parameters.

$A_b = \{a_i | i = 1 \dots N_b\}$ representing 'small' and 'large' anchors for the earlier F_2 and later F_3, respectively, so that the larger anchors are associated with the feature map of larger receptive field. These anchors $A_s \cup A_b$ are selected through a k-means clustering of the ground truth 3D bounding boxes. For our experiments, we use $N_s + N_b = 9$. From these $N_s + N_b$ clusters, A_b are those with any axis > 1.125m, and the rest are in A_s.

The two features maps F_2 and F_3 are then processed by a 3D region proposal to regress the 3D object bounding boxes. The 3D region proposal first employs a $1 \times 1 \times 1$ convolution layer to output objectness scores for each potential anchor, producing an objectness feature map with $2(N_s + N_b)$ channels for the positive and negative objectness probabilities. Another $1 \times 1 \times 1$ convolution layer is used to predict the 3D bounding box locations as 6-dimensional offsets from the anchors; we then apply a non-maximum suppression based on the objectness scores. We use a Huber loss on the log ratios of the offsets to the anchor sizes to regress the final bounding box predictions:

$$\Delta_x = \frac{\mu - \mu_{anchor}}{\phi_{anchor}} \qquad \Delta_w = \ln(\frac{\phi}{\phi_{anchor}})$$

where μ is the box center point and ϕ is the box width. The final bounding box loss is then:

$$L_\Delta = \begin{cases} \frac{1}{2}\Delta^2, & \text{if } |\Delta| \leq 2 \\ |\Delta|, & \text{otherwise.} \end{cases}$$

Using these predicted object bounding boxes, we then predict the object class labels using features cropped from the bounding box locations from F_2 and F_3. We use a 3D region of interest pooling layer to unify the sizes of the cropped feature maps to a spatial dimension of $4 \times 4 \times 4$ to be input to an object classification MLP.

4.3. Instance Completion

For each object, we infer its complete geometry by predicting per-voxel occupancies. Here, we crop features from feature map F_5 of the backbone, which has a feature map resolution matching the input spatial resolution, using the predicted object bounding box. These features are processed through a series of five 3D convolutions which maintain the spatial resolution of their input. The complete geometry is then predicted as voxel occupancy using a binary cross entropy loss.

We predict N_{classes} potential object completions for each class category, and select the final prediction based on the predicted object class. We define ground truth bounding boxes b_i and masks m_i as $\gamma = \{(b_i, m_i) | i = 1 \dots N_b\}$. Further, we define predicted bounding boxes $\hat{b_i}$ along with predicted masks $\hat{m_i}$ as $\hat{\gamma} = \{(\hat{b_i}, \hat{m_i}) | i = 1 \dots \hat{N_b}\}$. During training, we only train on predicted bounding boxes that overlap with the ground truth bounding boxes:

$$\Omega = \{(\hat{b_i}, \hat{m_i}, b_i, m_i) \mid \text{IoU}(\hat{b_i}, b_i) \geq 0.5, \\ \forall(\hat{b_i}, \hat{m_i}) \in \hat{\gamma}, \forall(b_i, m_i) \in \gamma\}$$

We can then define the instance completion loss for each

	display	table	bathtub	trashbin	sofa	chair	cabinet	bookshelf	**avg**
Scene Completion + Instance Segmentation	1.65	0.64	4.55	11.25	9.09	9.09	0.18	5.45	5.24
Instance Segmentation + Shape Completion	2.27	3.90	1.14	1.68	14.86	9.93	7.11	3.03	5.49
Ours – RevealNet (no color)	13.16	11.28	13.64	18.19	24.79	15.87	8.60	10.60	14.52
Ours – RevealNet (no proxy)	21.94	7.63	12.55	28.24	20.38	22.58	13.42	9.51	17.03
Ours – RevealNet	**26.86**	**13.21**	**22.31**	**28.93**	**29.41**	**23.64**	**15.35**	**14.48**	**21.77**

Table 1: 3D Semantic Instance Completion on ScanNet [4] scans with Scan2CAD [1] targets at mAP@0.5. Our end-to-end formulation achieves significantly better performance than alternative, decoupled approaches that first use state-of-the-art scan completion [7] and then instance segmentation [14] method or first instance segmentation [14] and then shape completion [6].

associated pair in Ω:

$$L_{\text{compl}} = \frac{1}{|\Omega|} \sum_{\Omega} \text{BCE}(\text{sigmoid}(\hat{m_i}), m_i'),$$

$$m_i'(v) = \begin{cases} m_i(v) & \text{if } v \in \hat{b_i} \cap b_i \\ 0 & \text{otherwise.} \end{cases}$$

We further introduce a global geometric completion loss on entire scene level that serves as an intermediate proxy. To this end, we use feature map F_5 as input to a binary cross entropy loss whose target is the composition of all complete object instances of the scene:

$$L_{\text{geometry}} = \text{BCE}(\text{sigmoid}(F_5), \cup_{(b_i, m_i) \in \gamma}).$$

Our intuition is to obtain a strong gradient during training by adding this additional constraint to each voxel in the last feature map F_5. We find that this global geometric completion loss further helps the final instance completion performance; see Sec 6.

5. Network Training

5.1. Data

The input 3D scans are represented as truncated signed distance fields (TSDFs) encoded in volumetric grids. The TSDFs are generated through volumetric fusion [3] during the 3D reconstruction process. For all our experiments, we used a voxel size of $\approx$ 4.7cm and truncation of 3 voxels. We also input the color images of the RGB-D scan, which we project to the 3D grid using their camera poses. We train our model on both synthetic and real scans, computing 9 anchors through k-means clustering; for real-world ScanNet [4] data, this results in 4 small anchors and 5 large anchors, and for synthetic SUNCG [39] data, this results in 3 small anchors and 6 large anchors.

At test time, we leverage the fully-convolutional design to input the full scan of a scene along with its color images. During training, we use random $96 \times 48 \times 96$ crops ($4.5 \times 2.25 \times 4.5$ meters) of the scanned scenes, along with a greedy selection of ≤ 5 images covering the most object geometry in the crop. Only objects with 50% of their complete geometry inside the crop are considered.

5.2. Optimization

We train our model jointly, end-to-end from scratch. We use an SGD optimizer with batch size 64 for object proposals and 16 for object classification, and all positive bounding box predictions (> 0.5 IoU with ground truth box) for object completion. We use a learning rate of 0.005, which is decayed by a factor of 0.1 every 100k steps. We train our model for 200k steps ($\approx$ 60 hours) to convergence, on a single Nvidia GTX 1080Ti. Additionally, we augment the data for training the object completion using ground truth bounding boxes and classification in addition to predicted object detection.

6. Results

We evaluate our approach on semantic instance completion performance on synthetic scans of SUNCG [39] scenes as well as on real-world ScanNet [4] scans, where we obtain ground truth object locations and geometry from CAD models aligned to ScanNet provided by Scan2CAD [1]. To evaluate semantic instance completion, we use a mean average precision metric on the complete masks (at IoU 0.5). Qualitative results are shown in Figs. 3 and 4.

Comparison to state-of-the-art approaches for semantic instance completion. Tables 1 and 2 evaluate our method against state of the art for the task of semantic instance completion on our real and synthetic scans, respectively. Qualitative comparisons on ScanNet scans [4] with Scan2CAD [1] targets (which provide ground truth for complete object geometry) are shown in Fig. 3. We compare to state-of-the-art 3D instance segmentation and scan completion approaches used sequentially; that is, first applying a 3D instance segmentation approach followed by a shape completion method on the predicted instance segmentation, as well as first applying a scene completion approach to the input partial scan, followed by a 3D instance segmentation method. For 3D instance segmentation, we evaluate 3D-SIS [14], which achieves state-of-the-art performance on a dense volumetric grid representation (the representation we use), and for scan completion we evaluate the 3D-

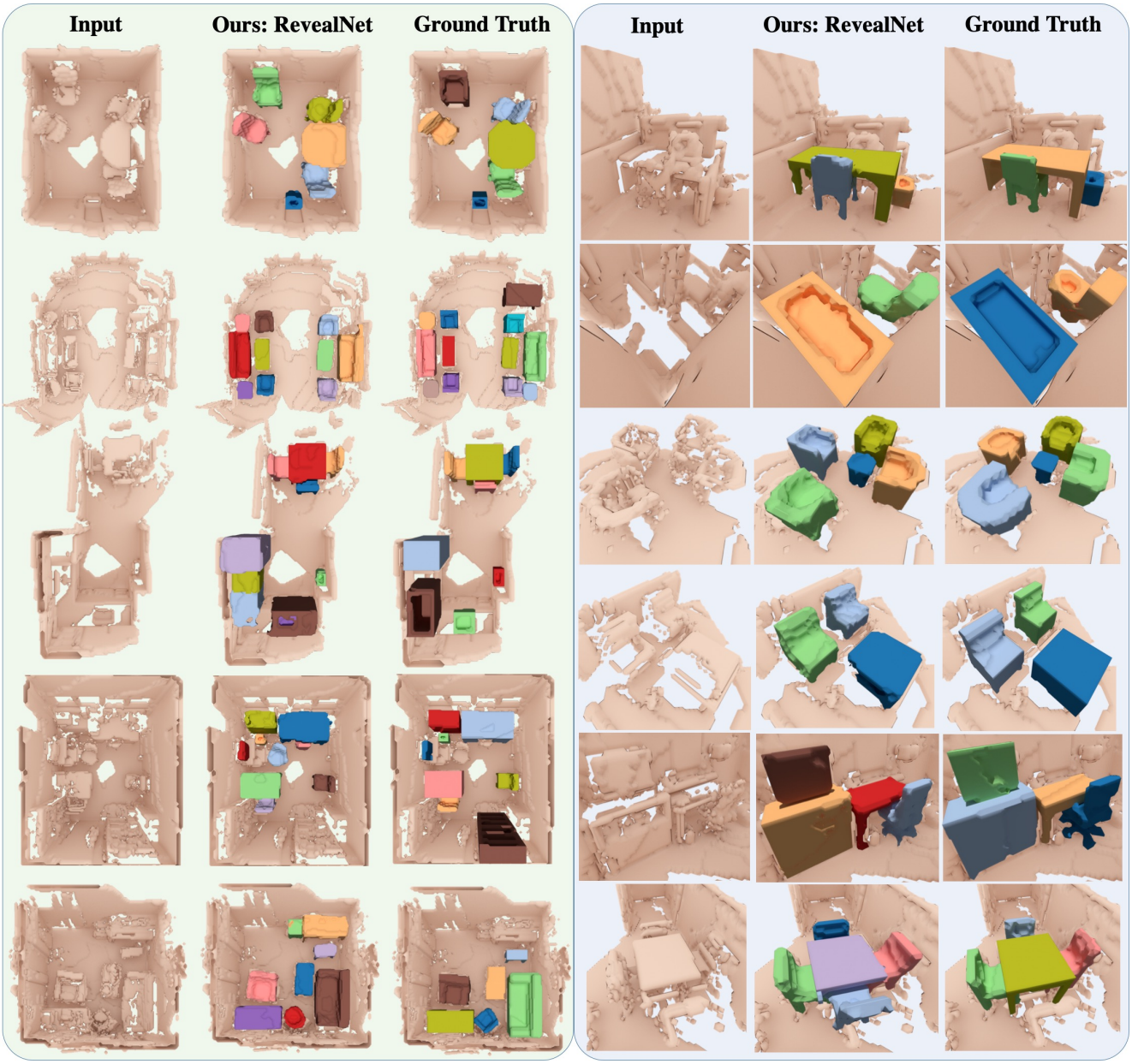

Figure 3: Qualitative results on real-world ScanNet [4] scenes with Scan2CAD [1] targets. Close-ups are shown on the right. Note that different colors denote distinct object instances in the visualization. Our approach effectively predicts complete individual object geometry, including missing structural components (e.g., missing chair legs), across varying degrees of partialness in input scan observations.

EPN [6] shape completion approach and ScanComplete [7] scene completion approach. Our end-to-end approach for semantic instance completion results in significantly improved performance due to information flow from instance completion to object detection. For instance, this allows our instance completion to more easily adapt to some inaccuracies in detection, which strongly hinders a decoupled approach. Note that the ScanComplete model applied on ScanNet data is trained on synthetic data, due to the lack of complete ground truth scene data (Scan2CAD provides only object ground truth) for real-world scans.

Does instance completion help instance detection and segmentation? We can also evaluate our semantic instance completion predictions on the task of semantic instance segmentation by taking the intersection between the predicted complete mask and the input partial scan geom-

	cab	bed	chair	sofa	tabl	door	wind	bkshf	cntr	desk	shlf	curt	drsr	mirr	tv	nigh	toil	sink	lamp	bath	ostr	ofurn	oprop	**avg**
SC + IS	3.0	0.6	19.5	0.8	18.1	**15.9**	0.00	0.0	1.0	2.3	3.0	0.0	0.5	0.0	9.2	10.4	23.9	3.4	9.1	0.0	0.0	0.0	9.1	5.5
IS + SC	0.3	0.0	7.4	0.4	3.0	9.1	0.0	0.0	0.2	0.0	0.0	0.0	2.3	0.0	3.0	0.0	2.6	0.0	1.8	0.0	0.0	0.0	4.6	1.5
no color	**19.05**	41.8	38.2	11.9	23.9	9.1	0.0	0.0	2.5	**21.6**	9.1	0.0	**12.6**	**4.6**	49.4	33.8	63.4	**36.9**	**38.8**	14.7	15.9	0.0	**23.8**	20.5
no proxy	12.9	46.1	**39.4**	26.8	**30.3**	1.0	15.9	0.0	9.1	18.2	3.4	0.0	1.1	0.0	43.6	**34.0**	**69.1**	32.4	29.6	**31.1**	14.6	0.0	23.3	20.9
Ours	14.7	**58.3**	38.2	**28.8**	29.5	0.0	**15.9**	**54.6**	**9.1**	12.1	**9.1**	**0.0**	6.2	0.0	**49.4**	33.5	61.2	34.5	29.5	27.1	**16.4**	**0.0**	23.5	**24.0**

Table 2: 3D Semantic Instance Completion on synthetic SUNCG [39] scans at mAP@0.5. Our semantic instance completion approach achieves significantly better performance than alternative approaches with decoupled state-of-the-art scan completion (SC) [7] followed by instance segmentation (IS) [14], as well as instance segmentation followed by shape completion [6]. We additionally evaluate our approach without color input (no color) and without a completion proxy loss on the network backbone (no proxy).

Figure 4: Qualitative results on SUNCG dataset [39] (left: full scans, right: close-ups). We sample RGB-D images to reconstruct incomplete 3D scans from random camera trajectories inside SUNCG scenes. Note that different colors denote distinct object instances in the visualization.

etry to be the predicted instance segmentation mask. We show that predicting instance completion helps instance segmentation, evaluating our method on 3D semantic instance segmentation with and without completion, on ScanNet [4] and SUNCG [39] scans in Tables 3 and 4, as well as 3D-SIS [14], an approach jointly predicts 3D detection and instance segmentation, which also operates on dense volumetric data, achieving state-of-the-art performance on this

	3D Detection	Instance Segmentation
3D-SIS [14]	25.70	20.78
Ours (no compl)	31.93	24.49
Ours (no color)	29.29	23.55
Ours (no proxy)	31.52	25.92
Ours	**36.39**	**30.52**

Table 3: 3D Detection and Instance Segmentation on ScanNet [4] scans with Scan2CAD [1] annotations at mAP@0.5. We evaluate our instance completion approach on the task of instance segmentation and detection to justify our contribution that instance completion task helps instance segmentation and detection. We evaluate our approach without completion (no compl), without color input (no color), and without a completion proxy loss on the network backbone (no proxy). Predicting instance completion notably increases performance of predicting both instance segmentation and detection (Ours vs. no compl). We additionally compare against 3D-SIS [14], a state-of-the-art approach for both 3D detection and instance segmentation on 3D dense volumetric data (the representation we use).

	3D Detection	Instance Segmentation
3D-SIS [14]	24.70	20.61
Ours (no compl)	29.80	23.86
Ours (no color)	31.75	31.59
Ours (no proxy)	34.05	32.59
Ours	**37.81**	**36.28**

Table 4: 3D Detection and Instance Segmentation on synthetic SUNCG [39] scans at mAP@0.5. To demonstrate the benefits of instance completion task for instance segmentation and 3D detection, we evaluate our semantic instance completion approach on the task of instance segmentation and 3D detection. Predicting instance completion notably benefits 3D detection and instance segmentation (Ours vs. no compl).

representation. We find that predicting instance completion significantly benefits instance segmentation, due to a more unified understanding of object geometric structures.

Additionally, we evaluate the effect on 3D detection in Tables 3 and 4; predicting instance completion also significantly improves 3D detection performance. Note that in contrast to 3D-SIS [14] which uses separate backbones for detection and instance segmentation, our unified backbone helps 3D mask information (complete or non-complete) propagate through detection parameters to improve 3D detection performance.

What is the effect of a global completion proxy? In Tables 1 and 2, we demonstrate the impact of the geometric completion proxy loss; here, we see that this loss improves the semantic instance completion performance on both real and synthetic data. In Tables 3 and 4, we can see that it also improves 3D detection and semantic instance segmentation performance.

Can color input help? Our approach takes as input the 3D scan geometry as a TSDF as well as the corresponding color images. We evaluate our approach with and without the color input stream; on both real and synthetic scans, the color input notably improves semantic instance completion performance, as shown in Tables 1 and 2.

7. Limitations

Our approach shows significant potential in the task of semantic instance completion, but several important limitations still remain. First, we output a binary mask for the complete object geometry, which can limit the amount of detail represented by the completion; other 3D representations such as distance fields or sparse 3D representations [10] could potentially resolve greater geometric detail. Our approach also uses axis-aligned bounding boxes for object detection; it would be helpful to additionally predict the object orientation. We also do not consider object movement over time, which contains significant opportunities for semantic instance completion in the context of dynamic environments.

8. Conclusion

In this paper, we tackle the problem of "seeing behind objects" by predicting the missing geometry of individual objects in RGB-D scans. This opens up many possibilities for complex interactions with objects in 3D, for instance for efficient navigation or robotic grasping. To this end, we introduced the new task of semantic instance completion along with RevealNet, a new 3D CNN-based approach to jointly detect objects and predict their complete geometry. Our proposed 3D CNN learns from both color and geometry features to detect and classify objects, then predicts the voxel occupancy for the complete geometry of the object in an end-to-end fashion, which can be run on a full 3D scan in a single forward pass. On both real and synthetic scan data, we significantly outperform state-of-the-art approaches for semantic instance completion. We believe that our approach makes an important step towards higher-level scene understanding and helps to enable object-based interactions and understanding of scenes, which we hope will open up new research avenues.

Acknowledgments

This work was supported by the ZD.B, a Google Research Grant, an Nvidia Professor Partnership, a TUM-IAS Rudolf Mößbauer Fellowship, and the ERC Starting Grant *Scan2CAD (804724)*.

References

[1] Armen Avetisyan, Manuel Dahnert, Angela Dai, Manolis Savva, Angel X. Chang, and Matthias Nießner. Scan2cad: Learning cad model alignment in rgb-d scans. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2019. 2, 5, 6, 8

[2] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3D: Learning from RGB-D data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017. 2

[3] Brian Curless and Marc 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. 5

[4] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. 2, 5, 6, 7, 8

[5] Angela Dai and Matthias Nießner. 3dmv: Joint 3d-multi-view prediction for 3d semantic scene segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 3, 4

[6] Angela Dai, Charles Ruizhongtai Qi, and Matthias Nießner. Shape completion using 3d-encoder-predictor cnns and shape synthesis. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. 3, 5, 6, 7

[7] Angela Dai, Daniel Ritchie, Martin Bokeloh, Scott Reed, Jürgen Sturm, and Matthias Nießner. Scancomplete: Large-scale scene completion and semantic segmentation for 3d scans. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2018. 2, 3, 5, 6, 7

[8] Cathrin Elich, Francis Engelmann, Jonas Schult, Theodora Kontogianni, and Bastian Leibe. 3d-bevis: Birds-eye-view instance segmentation. *arXiv preprint arXiv:1904.02199*, 2019. 2

[9] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 2

[10] Benjamin Graham and Laurens van der Maaten. Submanifold sparse convolutional networks. *arXiv preprint arXiv:1706.01307*, 2017. 8

[11] Xiaoguang Han, Zhen Li, Haibin Huang, Evangelos Kalogerakis, and Yizhou 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. 3

[12] Christian Häne, Shubham Tulsiani, and Jitendra Malik. Hierarchical surface prediction for 3d object reconstruction. *arXiv preprint arXiv:1704.00710*, 2017. 3

[13] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE, 2017. 2

[14] Ji Hou, Angela Dai, and Matthias Nießner. 3d-sis: 3d semantic instance segmentation of rgb-d scans. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2019. 2, 3, 4, 5, 7, 8

[15] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, volume 7, 2006. 2

[16] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics (TOG)*, 32(3):29, 2013. 2

[17] Young Min Kim, Niloy J Mitra, Dong-Ming Yan, and Leonidas Guibas. Acquiring 3d indoor environments with variability and repetition. *ACM Transactions on Graphics (TOG)*, 31(6):138, 2012. 3

[18] Jean Lahoud, Bernard Ghanem, Marc Pollefeys, and Martin R Oswald. 3d instance segmentation via multi-task metric learning. *arXiv preprint arXiv:1906.08650*, 2019. 2

[19] Yangyan Li, Angela Dai, Leonidas Guibas, and Matthias Nießner. Database-assisted object retrieval for real-time 3d reconstruction. In *Computer Graphics Forum*, volume 34, pages 435–446. Wiley Online Library, 2015. 3

[20] Tsung-Yi Lin, Piotr Dollár, Ross B Girshick, Kaiming He, Bharath Hariharan, and Serge J Belongie. Feature pyramid networks for object detection. In *CVPR*, volume 1, page 4, 2017. 2

[21] Tsung-Yi Lin, Priyal Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *IEEE transactions on pattern analysis and machine intelligence*, 2018. 2

[22] Chen Liu and Yasutaka Furukawa. Masc: Multi-scale affinity with sparse convolution for 3d instance segmentation. *arXiv preprint arXiv:1902.04478*, 2019. 2

[23] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 2

[24] Niloy J Mitra, Leonidas J Guibas, and Mark Pauly. Partial and approximate symmetry detection for 3d geometry. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 560–568. ACM, 2006. 3

[25] Liangliang Nan, Ke Xie, and Andrei Sharf. A search-classify approach for cluttered indoor scene understanding. *ACM Transactions on Graphics (TOG)*, 31(6):137, 2012. 3

[26] Gaku Narita, Takashi Seno, Tomoya Ishikawa, and Yohsuke Kaji. Panopticfusion: Online volumetric semantic mapping at the level of stuff and things. *arXiv preprint arXiv:1903.01177*, 2019. 2

[27] Andrew Nealen, Takeo Igarashi, Olga Sorkine, and Marc 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. 2

[28] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv:1606.02147*, 2016. 3

[29] Mark Pauly, Niloy J Mitra, Johannes Wallner, Helmut Pottmann, and Leonidas J Guibas. Discovering structural

regularity in 3d geometry. In *ACM transactions on graphics (TOG)*, volume 27, page 43. ACM, 2008. 3

[30] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. *arXiv preprint arXiv:1904.09664*, 2019. 2

[31] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. *arXiv preprint arXiv:1711.08488*, 2017. 2

[32] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 2

[33] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 2

[34] Tianjia Shao, Weiwei Xu, Kun Zhou, Jingdong Wang, Dongping Li, and Baining Guo. An interactive approach to semantic modeling of indoor scenes with an rgbd camera. *ACM Transactions on Graphics (TOG)*, 31(6):136, 2012. 3

[35] Yifei Shi, Pinxin Long, Kai Xu, Hui Huang, and Yueshan Xiong. Data-driven contextual modeling for 3d scene understanding. *Computers & Graphics*, 55:55–67, 2016. 3

[36] Ivan Sipiran, Robert Gregor, and Tobias Schreck. Approximate symmetry detection in partial 3d meshes. In *Computer Graphics Forum*, volume 33, pages 131–140. Wiley Online Library, 2014. 3

[37] Shuran Song and Jianxiong Xiao. Sliding shapes for 3d object detection in depth images. In *European conference on computer vision*, pages 634–651. Springer, 2014. 2

[38] Shuran Song and Jianxiong Xiao. Deep sliding shapes for amodal 3d object detection in rgb-d images. *arXiv preprint arXiv:1511.02300*, 2015. 2

[39] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. *Proceedings of 30th IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2, 3, 5, 7, 8

[40] Olga Sorkine and Daniel Cohen-Or. Least-squares meshes. In *Shape Modeling Applications, 2004. Proceedings*, pages 191–199. IEEE, 2004. 2

[41] Pablo Speciale, Martin R Oswald, Andrea Cohen, and Marc Pollefeys. A symmetry prior for convex variational 3d reconstruction. In *European Conference on Computer Vision*, pages 313–328. Springer, 2016. 3

[42] Sebastian Thrun and Ben Wegbreit. Shape from symmetry. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1824–1831. IEEE, 2005. 3

[43] Weiyue Wang, Ronald Yu, Qiangui Huang, and Ulrich Neumann. Sgpn: Similarity group proposal network for 3d point cloud instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2569–2578, 2018. 2

[44] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1912–1920, 2015. 3

[45] Bo Yang, Jianan Wang, Ronald Clark, Qingyong Hu, Sen Wang, Andrew Markham, and Niki Trigoni. Learning object bounding boxes for 3d instance segmentation on point clouds. *arXiv preprint arXiv:1906.01140*, 2019. 2

[46] Li Yi, Wang Zhao, He Wang, Minhyuk Sung, and Leonidas Guibas. Gspn: Generative shape proposal network for 3d instance segmentation in point cloud. *arXiv preprint arXiv:1812.03320*, 2018. 2

[47] Wei Zhao, Shuming Gao, and Hongwei Lin. A robust hole-filling algorithm for triangular mesh. *The Visual Computer*, 23(12):987–997, 2007. 2