SCFusion: Real-time Incremental Scene Reconstruction with Semantic Completion

Shun-Cheng Wu[1], Keisuke Tateno[2], Nassir Navab[1], Federico Tombari[1,2]
[1]Technische Universitat München, Germany [2]Google Inc., Switzerland
shuncheng.wu@tum.de, ktateno@google.com, nassir.navab@tum.de, tombari@google.com

Abstract

Real-time scene reconstruction from depth data inevitably suffers from occlusion, thus leading to incomplete 3D models. Partial reconstructions, in turn, limit the performance of algorithms that leverage them for applications in the context of, e.g., augmented reality, robotic navigation, and 3D mapping. Most methods address this issue by predicting the missing geometry as an offline optimization, thus being incompatible with real-time applications. We propose a framework that ameliorates this issue by performing scene reconstruction and semantic scene completion jointly in an incremental and real-time manner, based on an input sequence of depth maps. Our framework relies on a novel neural architecture designed to process occupancy maps and leverages voxel states to accurately and efficiently fuse semantic completion with the 3D global model. We evaluate the proposed approach quantitatively and qualitatively, demonstrating that our method can obtain accurate 3D semantic scene completion in real-time.

1. Introduction

The development of consumer depth cameras has fostered impressive advancements in research fields related to 3D geometry. Thanks to the availability of a stream of dense depth maps as input data, 3D computer vision algorithms are now capable of accurately localizing objects in the surrounding scene and reconstructing the 3D map of the environment. However, like any other camera, a depth camera is a viewpoint-dependent sensor which can estimate depth information only for the surface visible from the current vantage point. Since the foreground objects create occlusions for background objects and structure, this will result in missing depth information across multiple different objects. Hence, when depth maps are fused together via Simultaneous Localization and Mapping (SLAM) or Structure-from-Motion (SfM) [35, 7], the obtained 3D reconstructions are geometrically incomplete. Importantly, the incompleteness

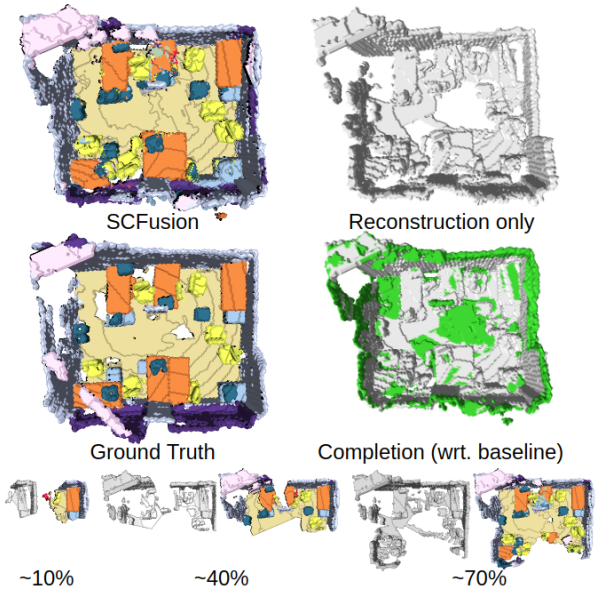

Figure 1: Incremental 3D reconstruction suffers from occlusions, leading to incomplete scenes. Unlike other approaches, which treat semantic scene completion as an offline optimization, SCFusion carries it out incrementally and in real-time along with scene reconstruction, yielding an accuracy comparable to state-of-the-art offline methods.

of the reconstructed scene challenges, in turn, those methods that leverage scene reconstruction for tasks related to, *e.g.*, augmented reality, robotic navigation and scene understanding, as they need to be robust to handle partial shapes.

Several methods have been proposed to recover the missing information in a scene from a given single depth image [39, 50, 46, 23, 47] or a reconstructed 3D scan [8, 4, 6]. These methods demonstrate the possibility of using partial observations to reasonably estimate the full geometry and even the semantic representations. Despite the important steps forward of research in scene completion, a gap remains to bring completion methods from either a single frame or the entire scan to real-time contexts. Existing

single-frame methods [39, 47, 50] treat each input frame individually without exploiting the availability of additional viewpoints, this leading to inaccuracies in the extrapolated shapes, which rely mostly on priors learned by the network from the training set. On the other hand, approaches designed to process an entire scan are off-line methods which treat this task as a post-processing optimization, and consequently cannot be used in real-time applications [6, 8]. In addition, the inconsistency of input and output formats limits the possibility of a direct integration within a unified model. Indeed, most methods take a (truncated) signed distance function (SDF) or one of its variations as input, while predicting occupancy probability [39, 50, 47] or an unsigned distance function (DF) [8]. The only method [6] that has consistent input and output formats does not predict semantic information. All these difficulties hinder the use of these approaches for real-time applications.

We fill this gap by proposing a framework consisting of a novel network and fusion scheme that enables real-time scene reconstruction with incremental semantic scene completion, which we dub *Scene Completion Fusion* (SCFusion). The two main pipelines that compose our proposed approach, *i.e*. scene reconstruction and scene completion, are carried out in parallel over two threads. The reconstruction pipeline continuously fuses input depth maps into a global volumetric map, while the completion pipeline semantically completes the global map by regularizing its sub-maps via a fully-connected conditional random field (CRF). The key component of our method is to use an occupancy representation over the entire system. The explicit voxel states and probability distribution on its estimation from the occupancy mapping [30, 15] provide more information than the use of TSDF fusion. We leverage this advantage and design a network which takes occupancy probability and voxels with unknown state as inputs to predict occupancy and semantic labels. The use of voxels with unknown state represents an additional input for the network, to guide it towards regions where completion might be necessary. The consistent input and output formats allow a direct integration between the predicted occupancy from the neural network and the global occupancy map built by the SLAM engine. In addition, we utilize the explicit known and unknown voxel state from occupancy mapping in our sub-map extraction and fusion process to define a set of integration policies which result in better performance in the task of semantic scene completion.

Notably, the currently available datasets do not have a complete ground truth for scenes since the one they include is obtained from real scans of the environment [5, 45, 3, 33, 24] that inevitably suffer from camera occlusions, while what has been a standard synthetic benchmark (SunCG [39]) is no longer available. Hence, to evaluate the performance of our method we have built a dataset

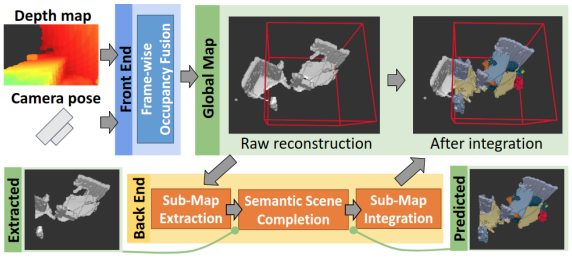

Figure 2: The overall pipeline of SCFusion for real-time scene reconstruction and semantic scene completion. The front-end reconstructs a 3D scene using occupancy maps. The back-end semantically completes the geometry and fuses it back to the global map.

with complete scenes using the alignments available in Scan2CAD [1], where the full 3D models from ShapeNet [49] are fused into the scenes of ScanNet [5]. We refer to this new dataset as CompleteScanNet.

Our contributions can be summarized as follows: (i) A framework for real-time incremental semantic scene completion, to the best of our knowledge the first of this kind. (ii) A novel neural architecture which leverages the voxel states in occupancy maps to predict scene completion and appropriately fuse it with the global 3D model, based on the idea of leveraging known/unknown states for 3D semantic scene completion. (iii) A benchmark dataset based on ShapeNet [49] 3D models and ScanNet [5] scenes that can be used to evaluate semantic scene completion algorithms based on RGB-D sequences. This fills an important gap given the absence of such benchmarks in literature.

2. Related Work

2.1. Real-time dense 3D reconstruction

Research activities in real-time dense 3D reconstruction can be subdivided between depth-based [35, 18] and monocular [36, 38, 51]. From the point of view of scene representation, the methods can be classified into three types: TSDF-based [35, 37, 16], occupancy-based [15, 30] and surfel-based [18, 48]. Given the scope of this paper, we will only review depth camera-based methods with TSDF and occupancy map representation.

For TSDF-based methods, Kinect Fusion [35] relies on a 3D volume which stores a Truncated Signed Distance Field (TSDF) value at each voxel. The input depth maps are back-projected into a 3D volume, which consists of voxel grids, and each depth observation is converted into a TSDF value and averaged together over the voxel grids. The surface is determined by finding the interpolated zero-crossing point via ray-casting. For occupancy-based methods, unlike TSDF which is a surface estimation approach, occupancy-based methods estimate occupancy in each voxel grid. This

restricts the reconstructed resolution to the defined voxel size, and the reconstructed surface is not well-defined. Loop et al. [26] proposed to use a quadratic b-spline instead of a Gaussian noise model which covers the gap of occupancy reconstruction and allows occupancy mapping to have equivalent accuracy as the TSDF method. Vespa et al. [44] propose to set the standard deviation value to be proportional to the measured depth distance to be more realistic to the triangulation-based depth camera noise model.

Since 3D volumetric representations are memory consuming and are difficult to extend to a large scale, Voxel Hashing [37] proposed a reconstruction framework based on a hash-grid 3D representation. Allocating small voxel blocks around observed surfaces instead of doing it on the entire space to reduce memory consumption and use memory more efficiently. [15] uses an octree to efficiently divide space into cubes of different size.

2.2. 3D shape completion

The completion of 3D shapes starts from an input partial 3D shape to provide an occlusion-free and complete 3D shape. Existing methods can be divided into two categories: object completion and scene completion.

Object completion focuses on a single object which includes recovering local surface primitives by using a continuous energy minimization [40, 34, 53], completing shapes by leveraging symmetric information or prior information from a database [43, 28, 41] and replacing partial shape with an aligned CAD model retrieved from a database [2, 20, 31]. However, to apply single object shape completion to scenes, additional object detection and segmentation are required. The completion quality thus additionally relies on how well the detection and segmentation methods perform.

Scene completion focuses on completing the entire scene with or without predicting the semantics. This can be done by using purely geometric approaches [9, 6] or by also considering semantic information. Recent trends target joint prediction of semantics and geometry by leveraging deep learning [39, 10, 46]. Most recent work focus on single viewpoint semantic completion, either based on a single depth image [39, 47] or with RGB information [10]. Only a few works target completion of an entire scan [8]. Although these methods have shown promising results, they target single frame prediction and do not handle sequences of depth maps or temporal information.

2.3. Semantic 3D reconstruction

This line of work focuses on joint optimization for 3D reconstruction and semantic segmentation. Initially, [22] proposed to jointly optimize semantic segmentation and stereo matching by using a random field. Later works tried to solve this problem by using a conditional random field on either single view [19] or multi-view [12, 13]. This idea was also extended to object-centric reconstruction [17, 11]. A common drawback of the above approaches is that they are not able to capture complex relationships in 3D space. Recently [4] suggested an end-to-end trainable way to capture more complex information between the semantic labels and the 3D geometry. However, these methods are computationally expensive and require extensive use of memory. Finally, a different corpus of work focuses on incremental construction of a semantic 3D map by fusing 2D semantic predictions from images [27, 32].

3. Proposed incremental semantic scene sompletion framework

In this section, we propose our framework for real-time incremental semantic scene completion. The flow diagram sketching the algorithm pipeline deployed at each input frame is shown in figure 2. The stages regarding front-end scene reconstruction are shown as blue boxes there, while those regarding back-end semantic scene completion are depicted with orange boxes. The outputs from front- and back-end operations are marked as green boxes. We assume a stream of depth maps acquired from a moving RGB-D sensor, and the corresponding poses for each depth image acquired from an external visual-inertial odometry (VIO) sensor or any pose estimation methods. Our front-end pipeline fuses each paired input pose and depth map into a gravity aligned dense volumetric occupancy map (section 3.1). The back-end pipeline predicts the occupancy and semantic label incrementally with our sub-maps strategy along with online map regularization (section 3.2).

3.1. Frame-wise occupancy map fusion

This section outlines our volumetric occupancy mapping method used in the front-end scene reconstruction pipeline. Our reconstruction system uses occupancy mapping instead of TSDF fusion to provide a consistent format with the prediction from our network. In addition, occupancy mapping provides three explicit states at each voxel: occupied, empty and unknown [15, 26], that can provide additional information to our network architecture as well as to our fusion scheme. As proposed in [37], we use a hashed voxel map representation which stores voxel cubes only when a depth measurement is present. Each voxel cube consists of the same number of voxels storing fused depth values from input measurements and, in our implementation, the predicted label from the back-end pipeline. For every input pair of a depth map and its pose, valid depth measurements are projected and fused into the global map. Each depth measurement is converted to the occupancy probability using a log-odd representation and the noise model proposed in [44].

Given a sensor measurement z_t at time t, each voxel in the global volumetric map stores a fused occupancy probability $P(\mathbf{v}|z_{1:t}) \in \mathbb{R}$ at each voxel location $\mathbf{v} \in \mathbb{R}^3$. During

occupancy fusion, the occupancy probability $P(\mathbf{v}|z_{1:t})$, *i.e.* the probability of a voxel being occupied or empty given the sensor measurements z_t, is estimated according to

$$P(\mathbf{v}|z_{1:t}) = \frac{P(\mathbf{v}|z_t)}{1-P(\mathbf{v}|z_t)} \frac{P(\mathbf{v}|z_{1:t-1})}{1-P(\mathbf{v}|z_{1:t-1})} \frac{1-P(\mathbf{v})}{P(\mathbf{v})} \quad (1)$$

based on the measured occupancy probability $P(\mathbf{v}|z_t)$ at time t, the previous probability $P(\mathbf{v}|z_{1:t-1})$ and a prior probability $P(\mathbf{v})$. The measured occupancy probability $P(\mathbf{v}|z_t)$ can be defined based on the given sensor model. Here, we use a quadratic B-spline sensor noise model as proposed in Vespa et al. [44] to compute $P(\mathbf{v}|z_t)$.

Then, a projected depth measurement corresponding to voxel location $\mathbf{v}$ in the global map coordinate system is computed as

$$z_t(\mathbf{v}) = T_t^{-1} K^{-1} \dot{\mathbf{u}} D_t(\mathbf{u}), \quad (2)$$

$$\mathbf{u} = \pi(K T_t \mathbf{v}) \in \mathbb{R}^2 \quad (3)$$

where $D_t(\cdot)$ is the input depth image at time t, $T_t = [R_t, t_t] \in \mathbb{SE}(3)$ is the current camera pose, composed of a 3×3 rotation matrix $R_t \in \mathbb{SO}(3)$ and a 3D translation vector $t_t \in \mathbb{R}^3$, and $\mathbf{u}$ is the pixel location in the input depth image projected from voxel location $\mathbf{v}$, with its homogeneous representation $\dot{\mathbf{u}}$.

As in Loop et al. [26], by using the log-odds notation and the common assumption of an uniform prior probability $P(\mathbf{v}) = \frac{1}{2}$, equation (1) can be re-written as

$$l(\mathbf{v}|z_{1:t}) = l(\mathbf{v}|z_{1:t-1}) + l(\mathbf{v}|z_t) \quad (4)$$

with $l(\mathbf{v}) = log\left(\frac{P(\mathbf{v})}{1-P(\mathbf{v})}\right)$

In addition to the occupancy probability, in each voxel we stores a semantic label $L_t(\mathbf{v}) \in L$ where L denotes all class labels, its confidence $W_t^L(\mathbf{v}) \in \mathbb{R}$ and a time stamp $T_t(\mathbf{v})$. The time t in our system represents the number of observations being fused to the global map.

3.2. Incremental semantic scene completion

In this section, we describe the stages of the proposed pipeline carrying out incremental sub-map semantic scene completion, which represents the core of our proposal. Given the incremental fashion of scene reconstruction, we propose to also perform scene completion incrementally during reconstruction. Our approach uses the view frustum from the given pose to search for candidate regions, referred to as sub-maps, which will be used to extract the input for the proposed network, as described in section 3.2.1. Then, the input scenes are semantically completed by our network (see section 3.2.2).

The use of sub-maps carries out completion at a local level and, thanks to its modularity, has the benefit of scaling up to large scenes. Nevertheless, by discretizing the scene geometry into sub-maps, this might create ambiguities on overlapping parts across neighboring sub-maps, *e.g.* in case of different predictions. Hence, we propose a novel fusion scheme (see section 3.2.3) with a map regularization method (see section 3.2.4) to handle this issue, aimed at fusing both temporal and spatial information. For the sake of efficiency and thread parallelization, all operations are performed in a separate GPU stream and CPU thread which are isolated from the main front-end pipeline.

3.2.1 Sub-map extraction

A sub-map is defined as an axis-aligned bounding box in the global map coordinate system, consisting of an anchor point and its enclosing bounding box. The anchor point is calculated based on the view frustum. In our implementation the bounding box has a fixed size of $64 \times 64 \times 64$. The view frustum is computed from the camera pose and its intrinsic parameters, with a distance range of [0.01m, 5m]. Given a view frustum, a minimum number of sub-maps are selected to cover the view frustum region. Specifically, by defining the y-axis as the elevation axis, the first sub-map is selected as the view frustum region with minimum x and z values. Then adjacent sub-maps are iteratively added until the view frustum is fully covered.

Next, we want to discard the extracted sub-maps whose voxels did not undergo major changes based on the last depth map update. This is important for efficiency reasons, since we want to avoid to run completion on regions that were not modified in time. For this we adopt a specific criterion: a sub-map is discarded if the percentage of the outdated voxels within its bounding box is above a threshold τ_s. A voxel is considered *outdated* when the difference between the voxel time stamp $T_t(v)$ and the current time t is larger than τ_t. In turn, the time stamp of a voxel is updated when a predicted information is fused with this voxel, being set to 0. With that, the system is able to prevent over completion on the same local map and also to increase the efficiency of the system. In our implementation, we set τ_s to 0.3 and τ_t to 30.

3.2.2 Semantic scene completion

A key characteristic of the proposed network is that it takes, as input, occupancy probabilities and a binary mask to predict a semantically completed scene. Similarly to [39], a voxel is consider occupied if it is assigned a non-empty label. We treat this state as an occupied observation in the map, which enables a direct integration of the associated voxel in the global map. The use of a binary mask is inspired by the research field in image inpainting, where it was shown that using a mask to indicate missing regions improves the prediction accuracy from a neural network [21].

Also, preventing mask vanishing during the recursive convolutional operations can further improve network performance [25, 52]. Inspired by this, we treat semantic scene completion as a 3D inpainting task, and propose a network architecture that leverages the benefits of occupancy maps and combines them with 3D inpainting.

Our network is built on the semantic prediction branch of ForkNet [47] with some modifications. First, the network takes a voxel grid of normalized occupancy probabilities and a voxel grid of binary masks as input. A binary mask is generated using the unknown state encoded with occupancy mapping. Second, we replace all convolutional layers with gated convolutional layers to prevent mask vanishing, as shown in [52]. Third, instance normalization is applied after each layer except for the final one. Last, a discriminator with spectral normalization is added during training [29]. For more details on the network architecture we kindly refer to the supplementary materials.

3.2.3 Sub-map integration

We leverage the three explicit states in occupancy mapping, i.e. *occupied*, *empty* and *unknown*, to design a fusion policy that carefully fuses the predicted results into the global map. Given a voxel prediction from the network, we define the following rules. First, the prediction is discarded if classified as empty or if its corresponding voxel in the global map is in the empty state. Second, the predicted semantic label is instead fused in the global map if the corresponding voxel is in either the unknown or occupied state. Analogously for the completion part, a voxel predicted as occupied by the network is fused only if the corresponding voxel in the global map is in the unknown state. Based on these rules, our method is able to add semantic and geometric information to the scene while being guaranteed to maintain the same reconstruction accuracy for the visible surface as the mapping approach that we employ as backbone, *i.e.* [44]. Remarkably, when fusing occupied voxels into the global map, we consider them as low confidence observations, with an assigned probability of 0.51, and follow the same approach described in section 3.1 to fuse the predictions. This has the benefit of improving wrongly predicted geometry coming from an individual observation (*i.e.*, a depth map).

As for merging labels, unlike occupancy estimation which deals with a continuous space, label prediction is categorical, hence requires a different integration strategy able to handle probability distributions. A simple solution is to save the entire probability distribution, however this would be impractical with a large number of labels, since the memory footprint in this case would grow linearly with the number of labels. Instead, we follow the approach in [42], which stores a single label and a confidence value per voxel. Unlike their method, which applies decrements and increments of the confidence depending on the number of similar observations, we propose to integrate the softmax value of the predicted label as representative of the label confidence. Specifically, for a given voxel $\mathbf{v}$ and its label $l_t(v)$ is predicted with a confidence value $w_t(v)$. If the confidence of the voxel is higher than the predicted confidence, we keep the label of the voxel from the previous state $L_t(v) = L_{t-1}(v)$, and the confidence weight of the voxel is updated as:

$$W_t^L(v) = \begin{cases} W_{t-1}^L(v) + w_t(v), & \text{if } (L_{t-1}(v) = l_t(v)) \\ W_{t-1}^L(v) - w_t(v), & \text{otherwise} \end{cases} \tag{5}$$

If $l_t(v)$ is different from the voxel label and the confidence of the voxel is below the predicted confidence, we replace the voxel label and reduce its weight as:

$$L_t(v) = l_t(v), W_t^L(v) = w_t(v) - W_{t-1}^L(v) \tag{6}$$

The label weight value is clamped with a maximum label confidence W_{max}^L.

3.2.4 Online map regularization

As mentioned, the use of sub-maps enables our framework to be real-time, but it also brings in potential inconsistencies nearby the borders of the sub-maps due to the discretization of the global map. To improve 3D semantic reconstruction accuracy under this aspect, we apply a regularization approach for the global map based on a fully connected CRF model, whose use for 3D maps has been explored in [32, 27], showing promising results.

As explained in 3.2.1, our map stores, at each voxel, only a label with an associated weight rather than the whole probability distribution. Hence, we use and modify the approach from [32] since it also relies on storing individual labels. We use the only voxel locations as regularization term. Differently from [32], which uses the frequency for a certain label as probability estimate, we calculate the probability of a voxel for a certain label L as:

$$p^L(v) = \max(W^L(v)/W_{max}^L, p_{min}^L) \tag{7}$$

where p_{min}^L is set to be slightly above the average label probability (in our case 0.1). Experimental results will be presented showing how our map regularization method is able to improve the accuracy of 3D semantic reconstruction - in particular, quantitatively in table 2), as well as qualitatively in the supplementary material.

4. Data generation

Due to the lack of a dataset including both completed 3D scene reconstructions and depth map sequences, it is

Metric	Method	Ceiling	Floor	Wall	Window	Chair	Bed	Sofa	Table	TV	Furni	Object	Mean
IoU	ForkNet	**0.360**	0.500	0.272	**0.255**	0.181	0.148	**0.335**	0.244	0.508	0.118	0.085	0.273
	Ours	0.197	**0.541**	**0.379**	0.108	**0.310**	**0.194**	0.266	**0.322**	**0.659**	**0.219**	**0.148**	**0.304**
Precision	ForkNet	**0.735**	**0.739**	0.466	**0.460**	0.421	0.278	**0.452**	0.375	0.554	0.245	0.244	0.452
	Ours	0.432	0.680	**0.591**	0.248	**0.528**	**0.304**	0.378	**0.505**	**0.771**	**0.410**	**0.300**	**0.468**
Recall	ForkNet	**0.463**	0.611	0.414	**0.564**	0.339	**0.697**	**0.772**	0.494	**0.913**	0.338	0.212	**0.529**
	Ours	0.343	**0.722**	**0.517**	0.391	**0.450**	0.667	0.714	**0.497**	0.842	**0.347**	**0.253**	0.522

Table 1: Comparison between ForkNet [47] and SCFusion on the test set of CompleteScanNet. Our method outperforms ForkNet in IoU and precision, while reporting slightly lower recall.

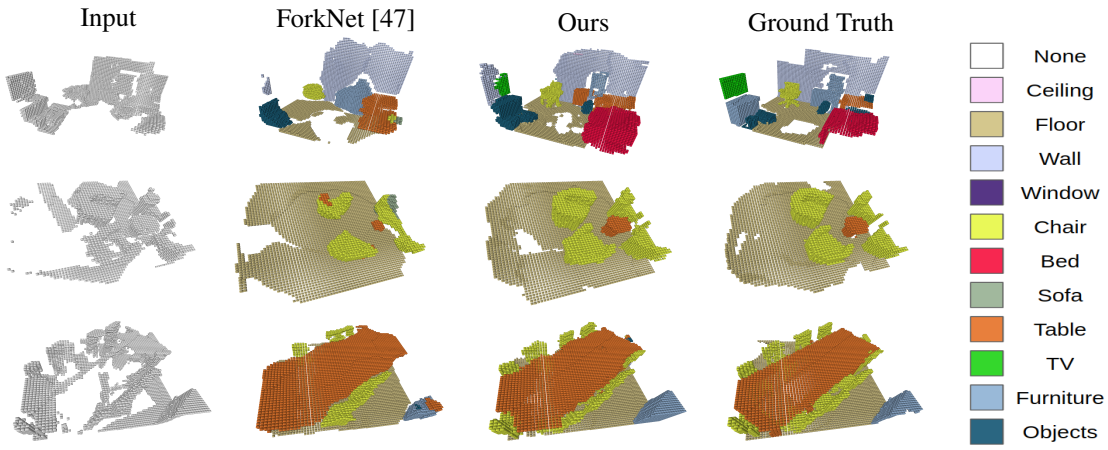

Figure 3: Semantic scene completion comparison on some CompleteScanNet test scenes. While both ForkNet and SCFusion can accurately complete geometry and predict semantics, our method outperforms ForkNet if compared to the ground truth.

hard to evaluate the performance of our approach, as well as to train our semantic scene completion network so that it can be applied on real data. Scene completion approaches [8, 47] relied on the synthetic SunCG dataset [39], which is currently no longer available. Recently, [6] showed how a scene completion network can be trained with incomplete ground truth in a self-supervised manner. However, this approach does not predict semantic labels, this limiting its use on many applications of interest.

To handle this issue, we developed a new benchmark where 3D object models are added to semantically annotated real-world scenes via model fitting. We use the model alignment annotations provided in Scan2CAD dataset [1] to fit 3D models from ShapeNet [49] in the 3D scenes of ScanNet [5]. We refer to this new dataset as **CompleteScanNet**. During model fitting, we replace the original object instances with their corresponding fully 3D object models to prevent geometry misalignments among object shapes. Then, we generate a depth sequence for each scene with full 3D objects by rendering depth maps. Note that the generated scenes still include incomplete parts due to the presence of incomplete background parts, e.g. floor, wall, ceiling. To help the network to better learn completion on incomplete parts, we devised a skip-frame training approach, where only one frame every 200 is processed, while the rest is discarded. This generates relatively more incomplete input scenes for training. Experimental results show that a network trained with this technique is able to predict a more complete and accurate scene than when trained on sequences that include all frames (see figures 3, 4).

Following this data generation pipeline, we use occupancy mapping with ground truth poses from ScanNet and rendered depth sequences to reconstruct a scene with voxel size of 5 cm^3. Then we use uniform sampling to sample sub-maps with a constant size of $[64 \times 64 \times 64]$. To ensure that the extracted sub-maps are not empty and have enough object variation, a sub-map is discarded if the percentage of empty voxels exceeds 95% and the number of different labels appearing in the scene is less than 2. After sub-map extraction and filtering , 45448 training and 11238 testing samples are generated from the initial 1201 training and 312 validation scenes. Finally, ScanNet labels are mapped to the SunCG [39] ones to prevent confusion during training caused by objects having different labels but similar shapes.

	Ceili.	Floor	Wall	Window	Chair	Bed	Sofa	Table	TV	Furni	Object	Mean
ForkNet+Fusion (-s)[47]	0.131	0.390	0.241	0.000	0.147	0.061	0.233	0.291	0.003	0.212	0.033	0.158
ScanComplete (-s)[8]	0.225	**0.541**	**0.440**	0.020	0.312	0.055	0.177	**0.371**	0.007	0.195	0.107	0.222
Proposed w/o CRF (-s)	**0.250**	0.532	0.413	0.121	0.348	0.277	0.339	0.350	0.084	0.287	0.130	0.284
Proposed (-s)	0.236	0.537	0.437	**0.134**	**0.362**	**0.282**	**0.346**	0.356	**0.086**	**0.299**	**0.136**	**0.292**
ForkNet+Fusion (-f)[47]	0.052	0.225	0.123	0.000	0.092	0.057	0.182	0.149	0.001	0.126	0.018	0.093
ScanComplete (-f)[8]	**0.164**	**0.393**	**0.350**	0.018	0.204	0.038	0.112	**0.277**	0.006	0.132	0.078	0.161
Proposed (-f)	0.128	0.329	0.265	**0.096**	**0.225**	**0.207**	**0.264**	0.210	**0.074**	**0.192**	**0.086**	**0.189**

Table 2: Comparison in IoU for semantic scene completion on CompleteScanNet. IoU is measured on both visible surfaces only (-s) and on the entire scan (-f).

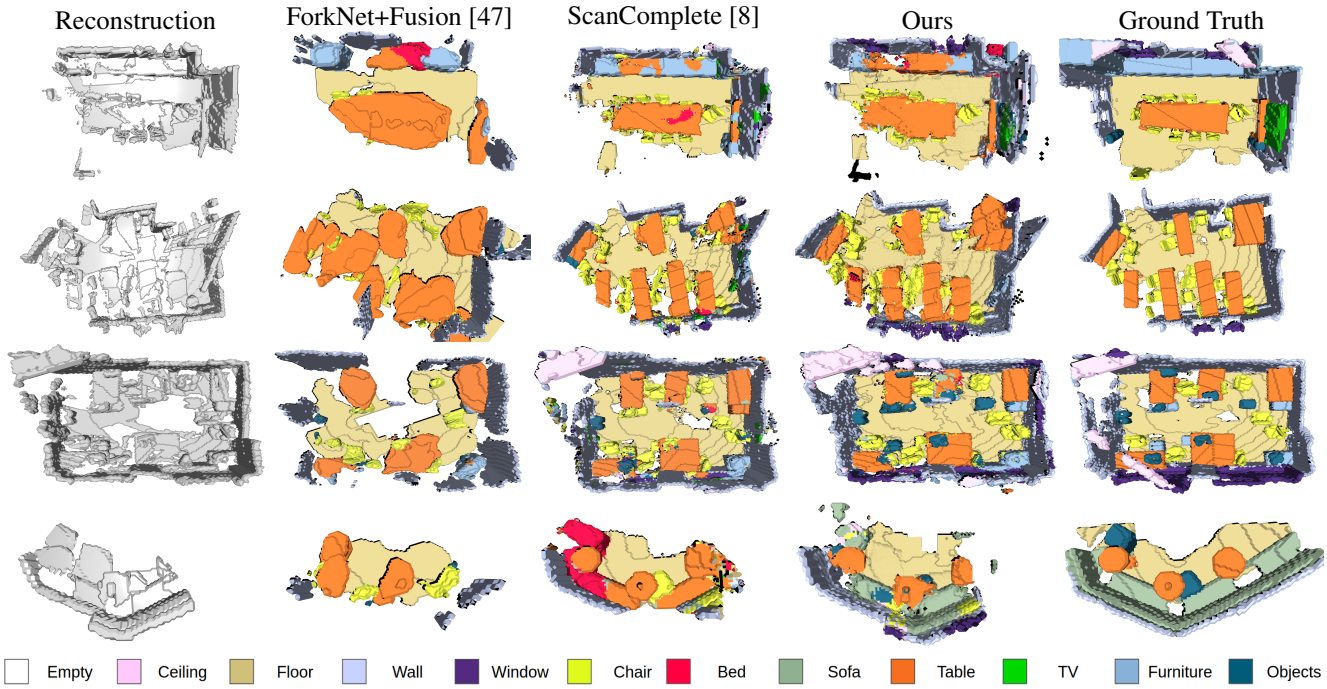

Figure 4: Semantic scene completion comparison on some CompleteScanNet test scenes.

5. Experimental results

Experiments are conducted to validate the performance of, respectively, the proposed network and the overall SCFusion framework on the CompleteScanNet dataset. First we validate the design of our network by comparing the performance on single prediction to its baseline method, *i.e.* ForkNet [47] with the use of intersection-over-union (IoU) as the metric on the CompleteScanNet sub-map test set. Then, we evaluate the performance of SCFusion on entire scenes by comparing it against the state of the art in offline scene completion, i.e. ScanComplete [8]. Finally, we show the effectiveness of the proposed fusion method by comparing it against fusion of ForkNet predictions.

5.1. Parameters and experimental environment

Training and testing is done on an Intel(R) Core i7-8700 CPU @ 3.20GHz, 64GB DDR4 RAM, 2 x NVIDIA GeForce RTX 2080ti. Our network is trained with Adam optimizer with a learning rate of 0.001, batch size of 4 with an accumulated gradient factor of 4. The whole network is trained for 40 hours until convergence.

5.2. Network evaluation

We evaluate the performance of our network against ForkNet [47] on the 11238 test sub-maps of CompleteScanNet. Both networks are trained from scratch until convergence on the CompleteScanNet training set. Since ForkNet takes inverted TSDFs as input, we generate its training data using TSDF fusion. Results in figure 3 and table 1 show that our method tends to predict more precise object shapes than

Thread	Operation	Mean (ms)	Std (ms)
Scene Reconstruction	Input Processing	0.039	0.097
	Mapping	5.956	23.484
Semantic Scene Completion	Sub-Map Extraction	0.620	0.467
	Semantic Scene Completion	123.067	8.898
	Sub-Map Fusion	1.124	0.232
	CRF Regularization	33.261	12.092
Average run-time per-frame		10.863	25.486

Table 3: Runtime analysis of the SCFusion stages averaged on the scene0645_00 sequence from CompleteScanNet.

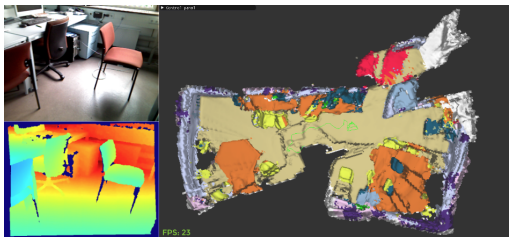

Figure 5: Output of SCFusion on a self-recorded sequence.

ForkNet, which instead predicts coarser geometries. Moreover, the predicted scenes of our method tend to be more complete than ForkNet. Although the dataset still partially present incomplete ground truth surfaces, both ForkNet and our method trained with the proposed skip-frame approach are able to predict reasonably complete scenes.

We provide an ablation study on the changes we made on the ForkNet architecture in the Section 2 of the supplementary material.

5.3. Full framework evaluation

We compare our scene reconstruction approach against the state of the art for scene completion on all 312 test scenes of the CompleteScanNet based on the IoU metric. Since our method predicts occupancies while ScanComplete predicts surfaces, we evaluate IoU on both the visible surface (referred as -s) and the entire occupancy (referred as -f). We use the same trained network as the one in the experiments of section 5.2. As for ScanComplete [8], we use the three hierarchy levels of training data generated by our pipeline described in section 4, and trained it from scratch until all levels converged. In addition, we include as baseline a framework where ForkNet [47] replaces our back-end and its completions are fused into a separated map (ForkNet+Fusion). Finally, as ablation, we include our method without CRF regularization.

All the test scenes are reconstructed with the rendered depths and the skip-frame method as described in our data generation section in order to evaluate both semantic labelling and scene completion. The scene prediction results are obtained using the proposed incremental pipeline, apart from ScanComplete which are predicted hierarchically from three-level of scans.

Results are illustrated in table 2 and figure 4. Our method has the highest IoU score in both visible surface (-s) and completed regions (-f). The naive integration of single predictions has the worst performance both qualitatively and quantitatively, since the predicted geometry often does not match the ground truth geometry. While ScanComplete is able to accurately complete the scene and predict semantic labels, our method obtains a higher accuracy for both reconstruction and semantics. It is noteworthy that our method runs in real-time with a single voxel level input while outperforming all other methods.

5.4. Run-time analysis

We present in table 3 a runtime analysis on one sequence (scene0645_00) of the CompleteScanNet dataset. The submap extraction and fusion operations will block the front-end pipeline for a short time due to the asynchronous access to the global map in the back-end pipeline. Since our method assumes a known pose, the time related to tracking is not included in the analysis.

5.5. Real-world scenario

Finally, we show qualitative results of SCFusion on a recorded office sequence using a Xtion PRO LIVE sensor with the frame-to-model pose estimation method implemented in InfiniTAM [16]. We use the same trained network in section 5.3. The reconstruction results are shown in figure 5. Our method can successfully obtain semantically complete scenes on the recorded sequence. Notably, the network was still trained on CompleteScanNet, which presents remarkable differences with respect to this test sequence. Also noteworthy, SCFusion runs in real-time also when including the tracking process. Please refer to the supplementary material for the full video.

6. Conclusions

We have proposed SCFusion, the first framework for incremental real-time semantic scene completion. Key ideas for our proposal are the design of a network that processes occupancy maps for efficient 3D semantic scene completion, as well as a specific fusion policy that can integrate completion with a global map incrementally built by a SLAM front-end. This fills a gap in the scene completion literature, arguably being the first approach to run semantic scene completion incrementally and in real-time. Experimental results show how SCFusion obtains accurate results both in terms of geometry and semantics, en par or even outperforming the state of the art for both offline completion of entire scans and single depth maps.

References

[1] A. Avetisyan, M. Dahnert, A. Dai, M. Savva, A. X. Chang, and M. Nießner. Scan2cad: Learning cad model alignment in rgb-d scans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2614–2623, 2019.

[2] A. Avetisyan, M. Dahnert, A. Dai, M. Savva, A. X. Chang, and M. Nießner. Scan2cad: Learning cad model alignment in rgb-d scans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2614–2623, 2019.

[3] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang. Matterport3D: Learning from RGB-D data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017.

[4] I. Cherabier, J. L. Schonberger, M. R. Oswald, M. Pollefeys, and A. Geiger. Learning priors for semantic 3d reconstruction. In *Proceedings of the European conference on computer vision (ECCV)*, pages 314–330, 2018.

[5] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5828–5839, 2017.

[6] A. Dai, C. Diller, and M. Nießner. Sg-nn: Sparse generative neural networks for self-supervised scene completion of rgb-d scans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 849–858, 2020.

[7] 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.

[8] A. Dai, D. Ritchie, M. Bokeloh, S. Reed, J. Sturm, and M. Nießner. Scancomplete: Large-scale scene completion and semantic segmentation for 3d scans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2018.

[9] 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.

[10] Y. Guo and X. Tong. View-volume network for semantic scene completion from a single depth image. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 726–732. AAAI Press, 2018.

[11] C. Hane, N. Savinov, and M. Pollefeys. Class specific 3d object shape priors using surface normals. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–659, 2014.

[12] C. Hane, C. Zach, A. Cohen, R. Angst, and M. Pollefeys. Joint 3d scene reconstruction and class segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 97–104, 2013.

[13] C. Häne, C. Zach, A. Cohen, and M. Pollefeys. Dense semantic 3d reconstruction. *IEEE transactions on pattern analysis and machine intelligence*, 39(9):1730–1743, 2016.

[14] K. Hara, H. Kataoka, and Y. Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6546–6555, 2018.

[15] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard. Octomap: An efficient probabilistic 3d mapping framework based on octrees. *Autonomous robots*, 34(3):189–206, 2013.

[16] O. Kahler, 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. In *The IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2015.

[17] R. Karimi Mahabadi, C. Hane, and M. Pollefeys. Segment based 3d object shape priors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2838–2846, 2015.

[18] M. Keller, D. Lefloch, M. Lambers, S. Izadi, T. Weyrich, and A. Kolb. Real-time 3d reconstruction in dynamic scenes using point-based fusion. In *Proceedings of Joint 3DIM/3DPVT Conference (3DV)*, 2013.

[19] B.-s. Kim, P. Kohli, and S. Savarese. 3d scene understanding by voxel-crf. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1425–1432, 2013.

[20] 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.

[21] R. Köhler, C. Schuler, B. Schölkopf, and S. Harmeling. Mask-specific inpainting with deep neural networks. In *German Conference on Pattern Recognition*, pages 523–534. Springer, 2014.

[22] L. Ladickỳ, P. Sturgess, C. Russell, S. Sengupta, Y. Bastanlar, W. Clocksin, and P. H. Torr. Joint optimization for object class segmentation and dense stereo reconstruction. *International Journal of Computer Vision*, 100(2):122–133, 2012.

[23] J. Li, Y. Liu, D. Gong, Q. Shi, X. Yuan, C. Zhao, and I. Reid. Rgbd based dimensional decomposition residual network for 3d semantic scene completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7693–7702, 2019.

[24] W. Li, S. Saeedi, J. McCormac, R. Clark, D. Tzoumanikas, Q. Ye, Y. Huang, R. Tang, and S. Leutenegger. Interiornet: Mega-scale multi-sensor photo-realistic indoor scenes dataset. In *British Machine Vision Conference (BMVC)*, 2018.

[25] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro. Image inpainting for irregular holes using partial convolutions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 85–100, 2018.

[26] C. Loop, Q. Cai, S. Orts-Escolano, and P. A. Chou. A closed-form bayesian fusion equation using occupancy probabilities. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 380–388. IEEE, 2016.

[27] J. McCormac, A. Handa, A. Davison, and S. Leutenegger. Semanticfusion: Dense 3d semantic mapping with convolutional neural networks. In *2017 IEEE International Conference on Robotics and automation (ICRA)*, pages 4628–4635. IEEE, 2017.

[28] 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.

[29] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018.

[30] H. Moravec and A. Elfes. High resolution maps from wide angle sonar. In *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, volume 2, pages 116–121. IEEE, 1985.

[31] 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.

[32] G. Narita, T. Seno, T. Ishikawa, and Y. Kaji. Panopticfusion: Online volumetric semantic mapping at the level of stuff and things. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4205–4212. IEEE, 2019.

[33] P. K. Nathan Silberman, Derek Hoiem and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[34] 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.

[35] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. W. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *ISMAR*, volume 11, pages 127–136, 2011.

[36] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison. Dtam: Dense tracking and mapping in real-time. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2011.

[37] M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger. Real-time 3d reconstruction at scale using voxel hashing. *ACM Transactions on Graphics (ToG)*, 32(6):169, 2013.

[38] T. Schöps, T. Sattler, C. Häne, and M. Pollefeys. 3D modeling on the go: Interactive 3D reconstruction of large-scale scenes on mobile devices. In *Proceedings of Joint 3DIM/3DPVT Conference (3DV)*, 2015.

[39] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser. Semantic scene completion from a single depth image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1746–1754, 2017.

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

[41] 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.

[42] K. Tateno, F. Tombari, and N. Navab. Real-time and scalable incremental segmentation on dense slam. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4465–4472. IEEE, 2015.

[43] 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.

[44] E. Vespa, N. Nikolov, M. Grimm, L. Nardi, P. H. Kelly, and S. Leutenegger. Efficient octree-based volumetric slam supporting signed-distance and occupancy mapping. *IEEE Robotics and Automation Letters*, 3(2):1144–1151, 2018.

[45] J. Wald, A. Avetisyan, N. Navab, F. Tombari, and M. Nießner. Rio: 3d object instance re-localization in changing indoor environments. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7658–7667, 2019.

[46] Y. Wang, D. J. Tan, N. Navab, and F. Tombari. Adversarial semantic scene completion from a single depth image. In *2018 International Conference on 3D Vision (3DV)*, pages 426–434. IEEE, 2018.

[47] Y. Wang, D. J. Tan, N. Navab, and F. Tombari. Forknet: Multi-branch volumetric semantic completion from a single depth image. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8608–8617, 2019.

[48] T. Whelan, S. Leutenegger, R. F. Salas-Moreno, B. Glocker, and A. J. Davison. Elasticfusion: Dense slam without a pose graph. In *The Robotics: Science and Systems Conference (RSS)*, 2015.

[49] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. 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.

[50] B. Yang, H. Wen, S. Wang, R. Clark, A. Markham, and N. Trigoni. 3d object reconstruction from a single depth view with adversarial learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 679–688, 2017.

[51] M. Yokozuka, S. Oishi, T. Simon, and A. Banno. Vitamin-e: Visual tracking and mapping with extremely dense feature points. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[52] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang. Free-form image inpainting with gated convolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4471–4480, 2019.

[53] W. Zhao, S. Gao, and H. Lin. A robust hole-filling algorithm for triangular mesh. *The Visual Computer*, 23(12):987–997, 2007.

7. Supplementary material

7.1. Network architecture

Our network architecture consists of a generator and a discriminator, which are highlighted in a green and an orange box, respectively (Figure 7). The generator consists of gated convolutional layers, denoted as $GConv(c, k, s, d)$, 3D ResNet blocks [14], denoted as $Res3D(\cdot, \cdot)$, 3D transpose convolution layers, denoted as $ConvT(c, k, s, d)$ and a SoftMax layer. Where c is the output channels, k is the number of kernels, s is the stride, d is the dilation. Note that all the convolutional layers, including inside the $Res3D(\cdot, \cdot)$ are replaced with gated convolutional layers. Apart from the final $GConv(\cdot)$ layer and Softmax, all operations in the generator are followed with an instance normalization layer and a LeakyRelu activation function with a negative slope ratio of 0.2. As for the discriminator, it consists of five convolutional layers, denoted as $Conv(c, k, s, d)$, all followed by a spectral normalization operation and a leakyReLu operation, except the final convolutional layer.

7.2. Ablation study

7.2.1 The effect of map regularization

The effect of our map regularization method is illustrated in figure 6. We highlighted the regions which failed to predict labels that are corrected by the regularization.

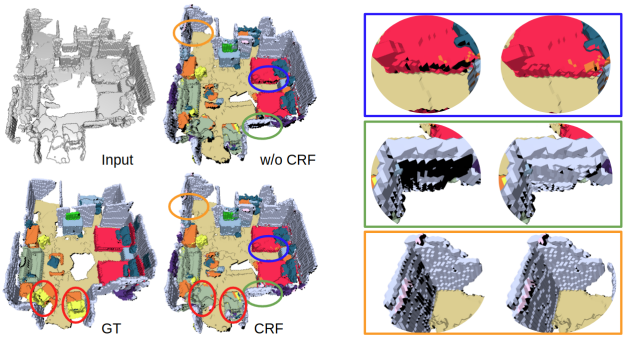

Figure 6: The effect of our regularization method and some failure cases of our SCFusion. The blue, green, and orange areas indicate the regions with noticeable improvement by our regularization method. The red circles show the failure case of misleading geometry of connecting chair and table, which result in wrongly label prediction.

7.2.2 Different network designs

We compare our final design with the two other setups with the metrics we used to evaluate our network in the main paper. First, as a baseline, the semantic scene prediction branch from ForkNet[47] with the replacement of the input format from the inverted truncated distance function to the occupancy probability (denote as **base**). Second, we replace all the convolutional layers with the gated convolutional layer and add a mask, as an additional input, which indicates the regions where a completion process may be needed (denote as **base+G**). The result is shown in table 4 and figure 8. It can be seen that the use of gated convolutional layers and a mask improve the overall network performance in a margin. The effect of discriminator slightly improves the numerical result, while dramatically increases the prediction in fine details.

7.3. Limitations

Our method uses only geometry input which limits its ability to distinguish the objects with ambiguous geometry shape and the objects that only differ from colors. For instance, a small table is adjacent to a chair may be classified as a sofa (See figure 6 red circles).

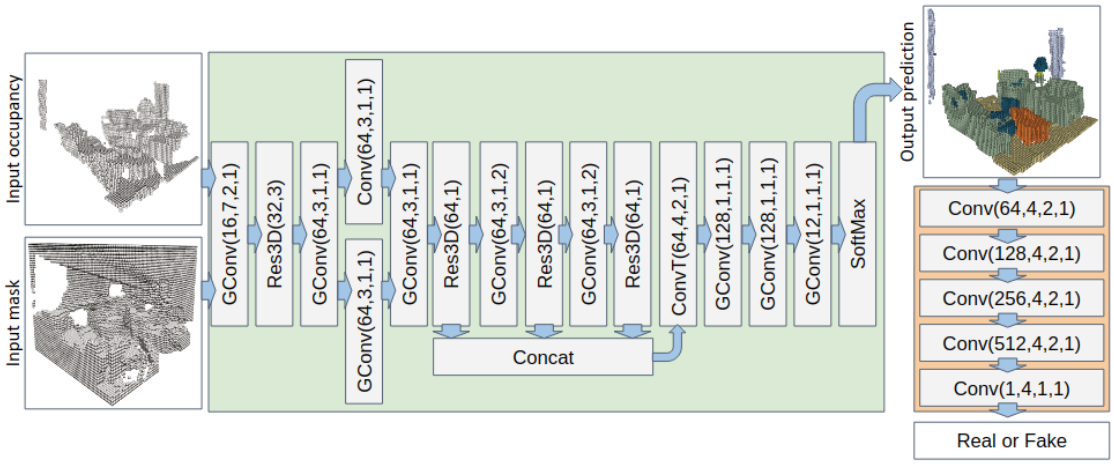

Figure 7: The proposed network architecture. The main network operations is included within the green box, while the operations of discriminator is shown within the orange box.

Metric	Method	Ceiling	Floor	Wall	Window	Chair	Bed	Sofa	Table	TV	Furni	Object	Mean
IoU	base	0.193	0.548	0.372	0.057	0.300	0.211	0.207	0.323	0.295	0.207	0.121	0.258
	base+G	**0.226**	**0.564**	**0.392**	0.068	**0.337**	**0.290**	**0.295**	**0.334**	0.181	0.207	**0.152**	0.277
	Ours	0.197	0.541	0.379	**0.108**	0.310	0.194	0.266	0.322	**0.659**	**0.219**	0.148	**0.304**
Precision	base	**0.561**	0.691	**0.654**	0.189	0.501	0.398	0.319	**0.582**	0 .367	0.381	**0.321**	0.451
	base+G	0.535	**0.757**	0.632	0.179	**0.551**	**0.495**	**0.440**	0.579	0.238	**0.424**	0.313	**0.468**
	Ours	0.432	0.680	0.591	**0.248**	0.528	0.304	0.378	0.505	**0.771**	0.410	0.300	**0.468**
Recall	base	0.248	0.704	**0.536**	0.082	0.447	0.260	0.251	0.471	0.310	0.314	0.202	0.348
	base+G	0.333	0.687	0.511	**0.397**	**0.484**	0.613	0.684	0.459	**0.850**	0.315	**0.257**	0.508
	Ours	**0.343**	**0.722**	0.517	0.391	0.450	**0.667**	**0.714**	**0.497**	0.842	**0.347**	0.253	**0.522**

Table 4: Ablation study of the network design. We compare our method (**ours**) against the baseline ForkNet [47] semantic branch (**base**) and the baseline plus an input mask with gated convolutions (**base+G**).

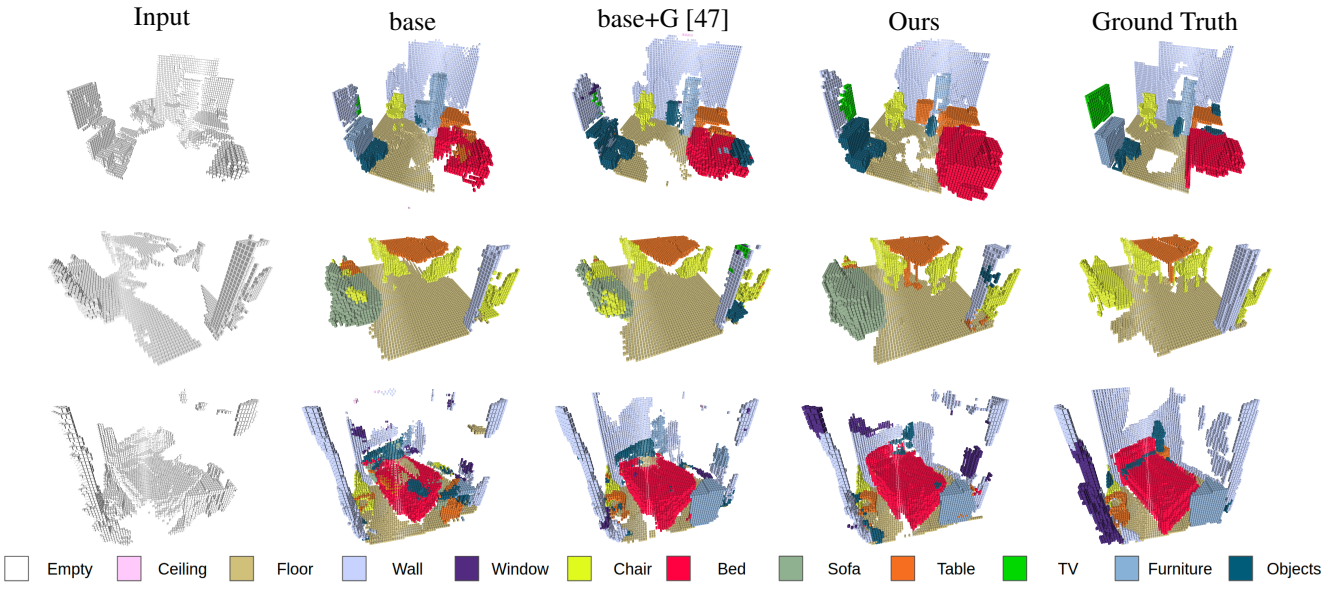

Figure 8: The use of gated convolutional layers with an input mask increase the completion ability significantly. Plus the constraint provided from the discriminator, the network is able to predict more complete scenes more precisely.