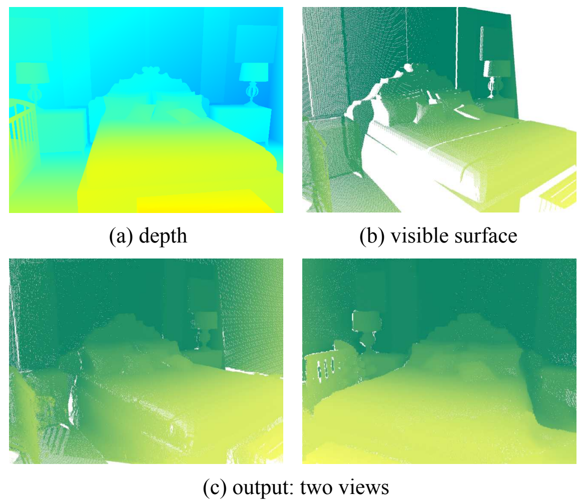
Deep Reinforcement Learning of Volume-guided Progressive View Inpainting for 3D Point Scene Completion from a Single Depth Image

[1,3]Xiaoguang Han, [2,3]Zhaoxuan Zhang, [3,4]Dong Du, [1,3]Mingdai Yang, [5]Jingming Yu, [5]Pan Pan
[2]Xin Yang, [4]Ligang Liu, [6]Zixiang Xiong, [1,3]Shuguang Cui

[1]CUHK(Shenzhen), [2]DUT, [3]SRIBD, [4]USTC, [5]Alibaba Group, [6]TAMU

Abstract

We present a deep reinforcement learning method of progressive view inpainting for 3D point scene completion under volume guidance, achieving high-quality scene reconstruction from only a single depth image with severe occlusion. Our approach is end-to-end, consisting of three modules: 3D scene volume reconstruction, 2D depth map inpainting, and multi-view selection for completion. Given a single depth image, our method first goes through the 3D volume branch to obtain a volumetric scene reconstruction as a guide to the next view inpainting step, which attempts to make up the missing information; the third step involves projecting the volume under the same view of the input, concatenating them to complete the current view depth, and integrating all depth into the point cloud. Since the occluded areas are unavailable, we resort to a deep Q-Network to glance around and pick the next best view for large hole completion progressively until a scene is adequately reconstructed while guaranteeing validity. All steps are learned jointly to achieve robust and consistent results. We perform qualitative and quantitative evaluations with extensive experiments on the SUNCG data, obtaining better results than the state of the art.



(a) depth (b) visible surface

(c) output: two views

Figure 1. Surface-generated Scene Completion. (a) A single-view depth map as input; (b) Visible surface from the depth map, which is represented as the point cloud. In our paper, the color of depth and point cloud is for visualization only; (c) Our scene completion results: directly recovering the missing points of the occluded regions. Here we choose two views for a better display.

1. Introduction

Recovering missing information in occluded regions of a 3D scene from a single depth image is a very active research area of late [36, 56, 12, 23, 9, 47]. This is due to its importance in robotics and vision tasks such as indoor navigation, surveillance, and augmented reality. Although this problem is mild in human vision system, it becomes severe in machine vision because of the sheer imbalance between input and output information. One class of popular approaches [32, 2, 13, 11] to this problem is based on classify-and-search: pixels of the depth map are classified into several semantic object regions, which are mapped to most similar 3D ones in a prepared dataset to construct a fully 3D scene. Owing to the limited capacity of the database, results from classify-and-search are often far away from the ground truth. By transforming the depth map into an incomplete point cloud, Song et al. [36] recently presented the first end-to-end deep network to map it to a fully voxelized scene, while simultaneously outputting the class labels each voxel belongs to. The availability of volumetric representations makes it possible to leverage 3D convolutional neural networks (3DCNN) to effectively capture the global contextual information, however, starting with an incomplete point cloud results in loss of input information and consequently low-resolution outputs. Several recent works [23, 12, 9, 47] attempt to compensate the lost information by extracting features from the 2D input domain in parallel and feeding them to the 3DCNN stream. To our best knowledge, no work has been done on addressing the second issue of improving output quality.

Taking an incomplete depth map as input, in this work we advocate the approach of straightforwardly reconstructing 3D points to fill missing region and achieve high-resolution completion (Figure 1). To this end, we propose to carry out completion on multi-view depth maps in an iterative fashion until all holes are filled, with each iteration focusing on one viewpoint. At each iteration/viewpoint, we render a depth image relative to the current view and fill the produced holes using 2D inpainting. The recovered pixels are re-projected to 3D points and used for the next iteration. Our approach has two issues: First, different choices of sequences of viewpoints strongly affect the quality of final results because given a partial point cloud, different visible contexts captured from myriad perspectives present various levels of difficulties in the completion task, producing diverse prediction accuracies; moreover, selecting a larger number of views for the sake of easier inpainting to fill smaller holes in each iteration will lead to error accumulation in the end. Thus we need a policy to determine the next best view as well as the appropriate number of selected viewpoints. Second, although existing deep learning based approaches [28, 16, 20] show excellent performance for image completion, directly applying them to depth maps across different viewpoints usually yields inaccurate and inconsistent reconstructions. The reason is because of lack of global context understanding. To address the first issue, we employ a reinforcement learning optimization strategy for view path planning. In particular, the current state is defined as the updated point cloud after the previous iteration and the action space is spanned by a set of pre-sampled viewpoints chosen to maximize 3D content recovery. The policy that maps the current state to the next action is approximated by a multi-view convolutional neural network (MVCNN) [38] for classification. The second issue is handled by a volume-guided view completion deepnet. It combines one 2D inpainting network [20] and another 3D completion network [36] to form a joint learning machine. In it low-resolution volumetric results of the 3D net are projected and concatenated to inputs of the 2D net, lending better global context information to depth map inpainting. At the same time, losses from the 2D net are back-propagated to the 3D stream to benefit its optimization and further help improve the quality of 2D outputs. As demonstrated in our experimental results, the proposed joint learning machine significantly outperforms existing methods quantitatively and qualitatively.

In summary, our contributions are

- The first surface-generated algorithm for 3D scene completion from a single depth image by directly generating the missing points.
- A novel deep reinforcement learning strategy for determining the optimal sequence of viewpoints for progressive scene completion.
- A volume-guided view inpainting network that not only produces high-resolution outputs but also makes full use of the global context.

2. Related Works

Many prior works are related to scene completion. The literature review is conducted in the following aspects.

Geometry Completion Geometry completion has a long history in 3D processing, known for cleaning up broken single objects or incomplete scenes. Small holes can be filled by primitives fitting[31, 19], smoothness minimization[37, 58, 17], or structures analysis[25, 35, 39]. These methods however seriously depend on prior knowledge. Template or part based approaches can successfully recover the underlying structures of a partial input by retrieving the most similar shape from a database, matching with the input, deforming disparate parts and assembling them[34, 18, 30, 39]. However, these methods require manually segmented data, and tend to fail when the input does not match well with the template due to the limited capacity of the database. Recently, deep learning based methods have gained much attentions for shape completion[30, 42, 33, 45, 5, 14], while scene completion from sparse observed views remains challenging due to large-scale data loss in occluded regions. Song et al.[36] first propose an end-to-end network based on 3DCNNs, named SSCNet, which takes a single depth image as input and simultaneously outputs occupancy and semantic labels for all voxels in the camera view frustum. ScanComplete[6] extends it to handle larger scenes with varying spatial extent. Wang et al.[47] combine it with an adversarial mechanism to make the results more plausible. Zhang et al.[56] apply a dense CRF model followed with SSCNet to further increase the accuracy. In order to exploit the information of input images, Garbade et al.[9] adopt a two stream neural network, leveraging both depth information and semantic context features extracted from the RGB images. Guo et al.[12] present a view-volume CNN which extracts detailed geometric features from the 2D depth image and projects them into a 3D volume to assist completed scene inference. However, all these works based on the volumetric representation result in low-resolution outputs. In this paper, we directly predict point cloud to achieve high-resolution completion by conducting inpainting on multi-view depth images.

Depth Inpainting Similar to geometry completion, researchers have employed various priors or optimized models to complete a depth image[15, 21, 27, 41, 3, 22, 51, 55]. The patch-based image synthesis idea is also applied[7, 10]. Recently, significant progresses have been achieved in im-

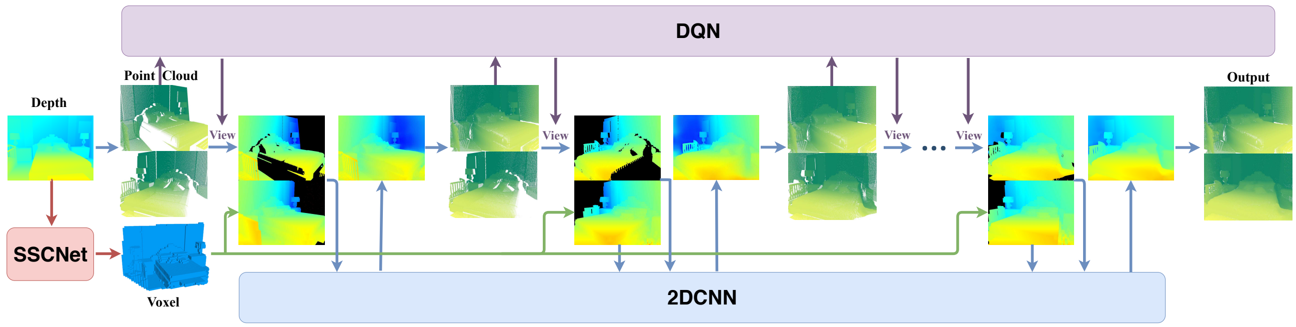


Figure 2. The pipeline of our method. Given a single depth image D_0, we convert it to a point cloud P, here shown in two different views. DQN is used to seek the next-best-view, under which the point cloud is projected to a new depth image D_1, causing holes. In parallel, the P is also completed in volumetric space by SSCNet, resulting in V. Under the view of D_1, V is projected and guide the inpainting of D_1 with a 2DCNN network. Repeating this process several times, we can achieve the final high-quality scene completion.

age or video inpainting field with deep convolutional networks and generative adversarial networks (GANs) for regular or free-form holes[16, 20, 54, 59, 46]. Zhang et al.[57] imitate them with a deep end-to-end model for depth inpainting. Compared with inpainting task on colorful images, recovering missing information from a single depth map is more challenging due to the absence of strong context features in depth maps. To address it, an additional 3D global context is provided in our paper, guiding the inpainting on diverse views to reach more accurate and consistent output.

View Path Planing Projecting a scene or an object to the image plane will severely cause information loss because of self-occusions. A straightforward solution is utilizing dense views for making up[38, 29, 40], yet it will lead to heavy computation cost. Choy et al.[4] propose a 3D recurrent neural networks to integrate information from multi-views which decreases the number of views to five or less. Even so, how many views are sufficient for completion and which views are better to provide the most informative features, are still open questions. Optimal view path planning, as the problem to predict next best view from current state, has been studied in recent years. It plays critical roles for scene reconstruction as well as environment navigation in autonomous robotics system[24, 1, 60, 49]. Most recently, this problem is also explored in the area of object-level shape reconstruction[52]. A learning framework is designed in [50], by exploiting the spatial and temporal structure of the sequential observations, to predict a view sequence for groundtruth fitting. Our work explores the approaches of view path planning for scene completion. We propose to train a Deep Q-Network (DQN)[26] to choose the best view sequence in a reinforcement learning framework.

3. Algorithm

Overview

Taking a depth image D_0 as input, we first convert it to a point cloud P_0, which suffers from severe data loss. Our goal is to generate 3D points to complete P_0. The main thrust of our proposed algorithm is to represent the incomplete point cloud as multi-view depth maps and perform 2D inpainting tasks on them. To take full advantage of the context information, we execute these inpainting operations view by view in an accumulative way, with inferred points for the current viewpoint kept and used to help inpainting of the next viewpoint. Assume D_0 is rendered from P_0 under viewpoint v_0, we start our completion procedure with a new view v_1 and render P_0 under v_1 to obtain a new depth map D_1, which potentially has many holes. We fill these holes in D_1 with 2D inpainting, turning D_1 to $\hat{D}_1$. The inferred depth pixels in $\hat{D}_1$ are then converted to 3D points and aggregated with P_0 to output a denser point cloud P_1. This procedure is repeated for a sequence of new viewpoints $v_2, v_3, ..., v_n$, yielding point clouds $P_2, P_3, ..., P_n$, with P_n being our final output. Figure 2 depicts the overall pipeline of our proposed algorithm. Since P_n depends on the view path $v_2, v_3, ..., v_n$, we describe in section 3.2 a deep reinforcement learning framework to seek the best view path. Before that, we introduce our solution to another critical problem of 2D inpainting, i.e., transforming D_i to $\hat{D}_i$, in section 3.1 first.

3.1. Volume-guided View Inpainting

Deep Convolutional Neural Network (CNN) has been widely utilized to effectively extract context features for image inpainting tasks, achieving excellent performance. Although it can be directly applied to each viewpoint independently, this simplistic approach will lead to inconsistencies across views because of lack of global context understandings. We propose a volume-guided view inpainting framework by first conducting completion in the voxel space, converting P_0's volumetric occupancy grid V to its completed version V^c. Denote the projected depth map from V^c to the view v_i as D_i^c. Our inpainting of the i_{th} view takes both D_i

and D_i^c as input and outputs $\hat{D}_i$. As shown in Figure 2, this is implemented using a three-module neural network architecture consisting of a volume completion network, a depth inpainting network, and a differentiate projection layer connecting them. The details of each module and our training strategy are described below.

Volume Completion We employ SSCNet proposed in [36] to map V to V^c for volume completion. SSCNet predicts not only volumetric occupancy but also the semantic labels for each voxel. Such a multi-task learning scheme helps us better capture object-aware context features and contributes to higher accuracy. The readers are referred to [36] for details on how to set up this network architecture. We train the network as a voxel-wise binary classification task and take the output 3D probability map as V^c. The resolution of input is $240 \times 144 \times 240$, and the output is $60 \times 36 \times 60$.

Depth Inpainting In our work, the depth map is rendered as a 512×512 grayscale image. Among various existing approaches, the method of [20] is chosen to handle our case with holes of irregular shapes. Specifically, D_i and D_i^c are first concatenated to form a map with 2 channels. The resulting map is then fed into a U-Net structure implemented with a masked and re-normalized convolution operation (also called partial convolution), followed by an automatic mask-updating step. The output is also in 512×512. We refer the readers to [20] for details of the architecture settings and the design of loss functions.

Projection Layer As validated in our experiments described in 4.2, the projection of V^c greatly benefits inpainting of 2D depth maps. We further exploit the benefit of 2D inpainting to volume completion by propagating the 2D loss back to optimize the parameters of 3D CNNs. Doing so requires a differentiable projection layer. There are two options for the implementation of this layer: the technique proposed in [43] and the homography warping method in [53]. The first one is chosen for a more accurate projection. Thus, we connect V^c and D_i^c using this layer. For the sake of notational convenience, we use V to represent V^c and D to represent D_i^c. Specifically, for each pixel x in D, we launch a ray that starts from the viewpoint v_i, passes through x, and intersects a sequence of voxels in V, noted as $l_1, l_2, ..., l_{N_x}$. We denote the value of the k_{th} voxel in V as V_k, which represents the probability of this voxel being empty. Then, we define the depth value of this pixel x as

$$D(x) = \sum_{k=1}^{N_x} P_k^x d_k \tag{1}$$

where d_k is the distance from the viewpoint to voxel l_k and P_k^x the probability of the ray corresponding to x first meets the l_k voxel

$$P_k^x = (1 - V_k) \prod_{j=1}^{k-1} V_j, \ k = 1, 2, ..., N_x \tag{2}$$

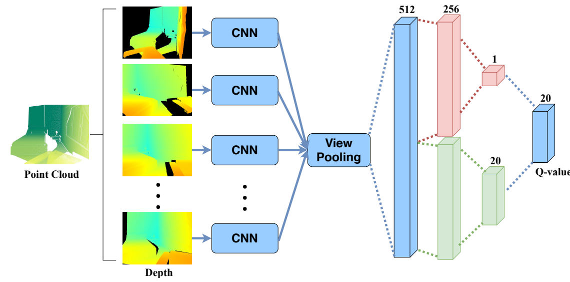


Figure 3. The architecture of our DQN. For a point cloud state, MVCNN is used to predict the best view for the next inpainting.

The derivative of $D(x)$ with respect to V_k can be calculated as

$$\frac{\partial D(x)}{\partial V_k} = \sum_{i=k}^{N_x} (d_{i+1} - d_i) \prod_{1 \le t \le i, t \ne k} V_t. \tag{3}$$

This guarantees back propagation of the projection layer. In order to speed up implementation, the processing of all rays are implemented in parallel via GPUs.

Joint Training Because our network consists of three subnetworks, we divide the entire training process into three stages to guarantee convergence: 1) The 3D convolution network is trained independently for scene completion; 2) With fixed parameters of the 3D convolution network, we train the 2D convolution network for depth image inpaintng under the guidance of 3D models; 3) We train the entire network jointly and fine tune it with all the parameters freed in 2D and 3D convolution networks.

The training data are generated based on the SUNCG synthetic scene dataset provided in [36]. We first create N depth images by rendering randomly selected scenes under randomly picked camera viewpoints. Each depth image D is then converted to a point cloud P. Assuming D is the projection of P under the viewpoint v, we project P to m depth maps from m randomly sampled views near v to avoid causing large holes and to ensure that sufficient contextual information is available in the learning process.

3.2. Progressive Scene Completion

Given an incomplete point cloud P_0 that is converted from D_0 with respect to view v_0, we describe in this subsection how to obtain the optimal next view sequence $v_1, v_2, ..., v_n$. The problem is defined as a Markov decision process (MDP) consisting of state, action, reward, and an agent which takes actions during the process. The agent inputs the current state, outputs the corresponding optimal action, and receives the most reward from the environment. We train our agent using DQN [26], an algorithm of deep reinforcement learning. The definition of the proposed MDP and the training procedure are given below.

State We define the state as the updated point cloud at each iteration, with the initial state being P_0. As the iteration continues, the state for performing completion on the i_{th} view is P_{i-1}, which is accumulated from all previous iteration updates.

Action Space The action at the i_{th} iteration is to determine the next best view v_i. To ease the training process and support the use of DQN, we evenly sample a set of scene-centric camera views to form a discrete action space. Specifically, we first place P_0 in its bounding sphere and keep it upright. Then, two circle paths are created for both the equatorial and 45-degree latitude line. In our experiments, 20 camera views are uniformly selected on these two paths, 10 per circle. All views are facing to the center of the bounding sphere. We fixed these views for all training samples. The set of 20 views is denoted as $C = \{c_1, c_2, ..., c_{20}\}$.

Reward An reward function is commonly unitized to evaluate the result for an action executed by the agent. In our work, at the i_{th} iteration, the input is an incomplete depth map D_i rendered from P_{i-1} under view v_i chosen in the action space C. The result of the agent action is an inpainted depth image $\hat{D}_i$. Hence the accuracy of this inpainting operation can be used as the primary rewarding strategy. It can be measured by the mean error of the pixels inside the holes between $\hat{D}_i$ and its ground truth D_i^{gt}. All the ground truth depth maps are pre-rendered from SUNCG dataset. Thus we define the award function as

$$R_i^{acc} = -\frac{1}{|\Omega|} L_\Omega^1(\hat{D}_i, D_i^{gt}), \tag{4}$$

where L^1 denotes the L_1 loss, Ω the set of pixels inside the holes, and $|\Omega|$ the number of pixels inside Ω.

If we only use the above reward function R_i^{acc}, the agent tends to change the viewpoint slightly in each action cycle, since doing this results in small holes. However, this incurs higher computational cost while accumulating errors. We thus introduce a new reward term to encourage inferring more missing points at each step. This is implemented by measuring the percentage of filled original holes. To do so, we need to calculate the area of missing regions in an incomplete point cloud P, which is not trivial in a 3D space. Therefore, we project P under all camera views to the action space C and count the number of pixels inside the generated holes in each rendered image. The sum of these numbers is denoted as $Area^h(P)$ for measuring the area. We thus define the new reward term as

$$R_i^{hole} = \frac{Area^h(P_{i-1}) - Area^h(P_i)}{Area^h(P_0)} - 1 \tag{5}$$

to avoid the agent from choosing the same action as in previous steps. We further define a termination criterion to stop view path search by $Area^h(P_i)/Area^h(P_0) < 5\%$, which means that all missing points of P_0 have been nearly recovered. We set the reward for terminal to zero.

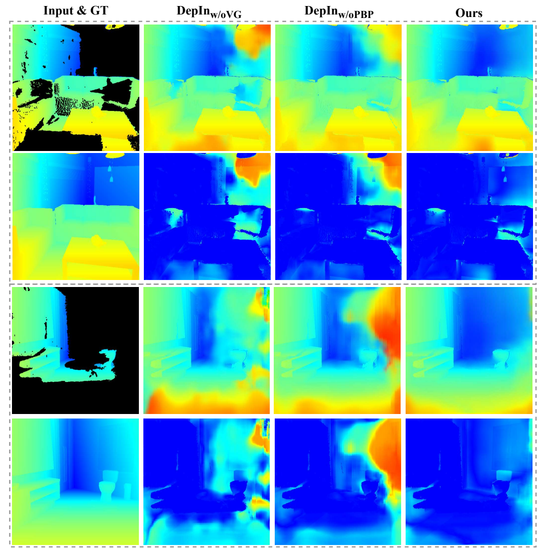


Figure 4. Comparisons on variants of depth inpainting network. Given incompleted depth images, we show results of our proposed method w/o volume-guidance, w/o projection back-propagation and also ours, compared with the groundtruth. Both the inpainted map and its error map are shown.

Therefore, our final reward function is

$$R_i^{total} = wR_i^{acc} + (1-w)R_i^{hole}, \tag{6}$$

where w is a fractional weight that balances the two reward terms.

DQN Training Our DQN is built upon MVCNN[38]. It takes mutil-view depth maps projected from P_{i-1} as inputs and outputs the Q-value of different actions. The whole network is trained to approximate the action-value function $Q(P_{i-1}, v_i)$, which is the expected reward that the agent receives when taking action v_i at state P_{i-1}.

To ensure stability of the learning process, we introduce a target network separated from the architecture of [26], whose loss function for training DQN is

$$Loss(\theta) = \mathbb{E}[(r + \gamma \max_{v_{i+1}} Q(P_i, v_{i+1}; \theta') - Q(P_{i-1}, v_i; \theta))^2]. \tag{7}$$

where r is the reward, γ a discount factor, and θ' the parameters of the target network. For effective learning, we create an experience replay buffer to reduce the correlation between data. The buffer stores the tuples (P_{i-1}, v_i, r, P_i) proceeded with the episode. We also employ the technique of [44] to remove upward bias caused by $\max_{v_{i+1}} Q(P_i, v_{i+1}; \theta')$ and change the loss function to

$$\begin{aligned} \mathbb{L}_{our} = \mathbb{E}[(r + \gamma Q(P_i, \arg\max_{v_{i+1}} Q(P_i, v_{i+1}; \theta); \theta') \\ - Q(P_{i-1}, v_i; \theta))^2]. \end{aligned} \tag{8}$$

Combining with the dueling DQN structure [48], our network structure is shown in Figure 3. At state P_{i-1}, we render at all viewpoints $c_1, c_2, ..., c_{20}$ in the action space C in 224×224 resolution and get the corresponding multi-view depth maps $D_i^1, D_i^2, ..., D_i^{20}$. These depth maps are then sent to the same CNN as inputs. After a view pooling layer and a fully-connected layer, we obtain a 512-D vector, which is split evenly into two parts to learn the advantage function $A(v, P)$ and the state value function $V(P)$ [48]. Finally, after combining the results of the two functions, we have our final result, which is a 20-D Q-values based on the action space C. We use an ϵ-greedy policy to choose action v_i for state P_{i-1}, i.e., a random action with probability $1-\epsilon$ or an action that maximizes the Q-values with probability ϵ. In the end, we reach the decision on depth map D_i for inpainting.

The training data are also generated from SUNCG. We use the same N depth images as in section 3.1. We also choose the action space C to generate new data. The ground truth depth maps, which are used in the reward calculation, are generated in the same viewpoint from the action space C.

4. Experimental Results

Dataset The dataset we used to train our 2DCNN and DQN is generated from SUNCG [36]. Specifically, for 2DCNN, we set $N = 3,000$ and $m = 10$ and get $30,000$ depth maps. We further remove the maps whose camera views are occluded by doors or walls. Then, $3,000$ of them are took for testing and the rest is used for training. For DQN, we set $N = 2,500$ with 2300 for the training episode and 200 for the testing.

Implementation Details Our network architecture is implemented in PyTorch. The provided pre-trained model of SSCNet [36] is used to initialize parameters of our 3DCNN part. It takes 30 hours to train inpainting network on our training dataset and 20 hours to fine-tune the whole network after the addition of projection layer. During DQN training process, we first use 200 episodes to fill experience replay buffer. In each episode, the DQN chooses the action randomly in each iteration step, and store the tuple (P_{i-1}, v_i, r, P_i) in the buffer. After those episodes being pre-trained, the network begins to learn by randomly sampled batches in buffers for each step during different episodes. The buffer can store $5,000$ tuples and the batch size is set to 16. The weight w for reward calculation is set as 0.7 and the discount factor γ is set to 0.9, while ϵ decreases from 0.9 to 0.2 over $10,000$ steps and then be fixed to 0.2. Training DQN takes 3 days and running our complete algorithm once takes about $60s$ which adopts five view points on average.

4.1. Comparisons Against State-of-the-Arts

In this part, we evaluate our proposed method against SSCNet [36] and ScanComplete [6], which are the most popular approaches in this area. Based on SSCNet, there although exists many incremental works such as [47] and [12], they all produce volumetric outputs in the same resolution as SSCNet. Regarding neither the code nor the pre-trained model of these methods is public, we propose to compare our result with the corresponding 3D groundtruth volume, whose output accuracy can be treated as the upper bound of all existing volume-based scene completion methods. We denote this method as $Volume-GT_1$. For evaluation, we first render the volume obtained from SSCNet and the volume gt to several depth maps under the same viewpoints as our method. We then convert these depth maps to point cloud. Note that, the method of [6] is also built upon SSCNet, but can output higher resolution volume. The accuracy of the groundtruth volume in that resolution, denoted as $Volume-GT_2$, is also reported.

Quantitative Comparisons The Chamfer Distance (CD) [8] is used as one of our metrics for evaluate the accuracy of our generated point set P, compared with the goundtruth point cloud P_{GT}. Similar to [8], we also use another completeness metric to evaluate how complete of the generated result. We define it as:

$$C_r(P, P_{GT}) = \frac{|\{d(x, P) < r | x \in P_{GT}\}|}{|\{y | y \in P_{GT}\}|} \tag{9}$$

where $d(x, P)$ denotes the distance from a point x to a point set P, $|\cdot|$ denotes the number of the elements in the set, and r means the distance threshold. In our experiments, we report the completeness w.r.t five different r ($0.02, 0.04, 0.06, 0.08, 0.10$ are used). The results are reported in Tab 1. As seen, our approach significantly outperforms all the others. This also validates that the using of volumetric representation greatly reduces the quality of the outputs.

Qualitative Comparisons The visual comparisons of these methods are shown in Figure 5. It can be seen that, the generated point cloud from SSCNet is of no surface details. Although our method shows more errors than volume-gt in some local regions, it overall produces more accurate results. This can be validated in Tab 1. In addition, by conducting completion in multiple views, our approach also recovers more missing points, showing better completeness as validated in Tab 1.

4.2. Ablation Studies

To ensure the effectiveness of several key components of our system, we do some control experiments by removing each component.

On Depth Inpainting Firstly, to evaluate the efficacy of the volume guidance, we propose two variants of our method: 1) we train a 2D inpainting network directly without projecting volume as guidance, which is denoted as

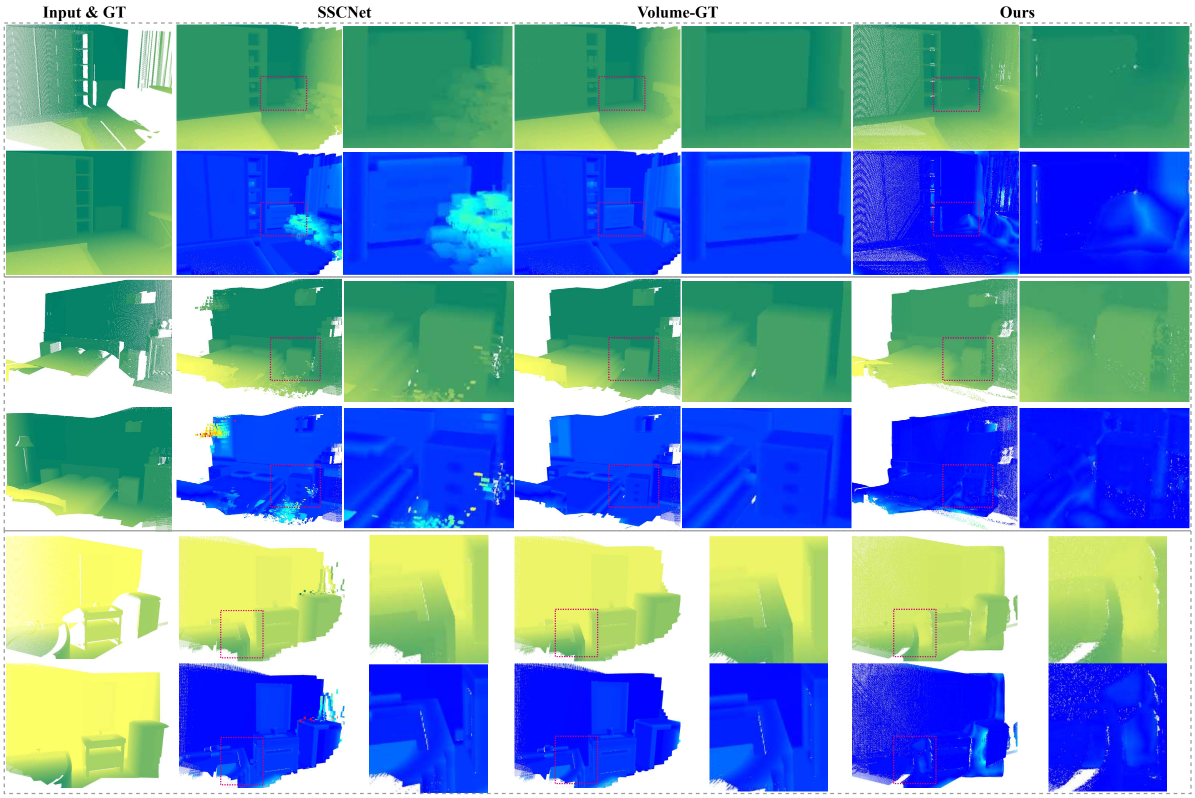


Figure 5. Comparisons against the state-of-the-arts. Given different inputs and the referenced groundtruth, we show the completion results of three methods, with the corresponding point cloud error maps below, and zoom-in areas beside. More blue more accurate.

Table 1. Quantitative comparisons of our method against existing methods and its variants. The CD metric and the completeness metric (w.r.t different thresholds) are used. $Volume - GT_1$ has same resolution with SSCNet, $Volume - GT_2$ has same resolution with ScanComplete.

	$SSCNet$	$Volume - GT_1$	$ScanComplete$	$Volume - GT_2$	U_5	U_{10}	$DQN_{w/o-hole}$	$Ours$
CD	0.5162	0.5140	0.2193	0.2058	0.1642	0.1841	0.1495	**0.1148**
$C_{r=0.02}(\%)$	14.61	13.28	34.46	31.18	79.18	80.17	79.22	79.26
$C_{r=0.04}(\%)$	30.10	32.23	58.83	61.11	83.33	84.15	83.50	83.68
$C_{r=0.06}(\%)$	52.82	50.14	74.60	74.88	85.81	86.56	86.02	86.28
$C_{r=0.08}(\%)$	71.24	72.33	79.59	81.04	87.66	88.33	87.81	88.20
$C_{r=0.10}(\%)$	78.23	78.96	81.01	81.61	89.06	89.70	89.24	89.68

$DepIn_{w/oVG}$; 2) we train the volume guided 2D inpainting network without projection back-propagation, which is denoted as $DepIn_{w/oPBP}$. We use the metrics of L^1_Ω, $PSNR$ and $SSIM$ for the comparisons. The quantitative results are reported in Tab 2 and the visual comparisons are shown in Figure 4. All of them show the superiority of our design.

Table 2. Quantitative ablation studies on inpainting network.

	$DepIn_{w/oVG}$	$DepIn_{w/oPBP}$	$Ours$
L^1_Ω	0.0717	0.0574	**0.0470**
$PSNR$	22.15	23.12	**24.73**
$SSIM$	0.910	0.926	**0.930**

On View Path Planning Without using DQN for path planning, there exists a straightforward way to do completion: we can uniformly sample a fixed number of views from C and directly perform depth implanting on them. In this uniform manner, two methods with two different numbers of views (5 and 10 are selected) are evaluated. We denote them as U_5 and U_{10}. The results of CD and $C_r(P, P_{GT})$ using these two methods and ours are reported in Tab 1. As seen, increasing the uniform sampled views causes accuracy reducing. This might be because of the increased accumulated errors. Using DQN greatly improves the accuracy, which validates the importance of a better view path. And all of them give rise to similar completeness. In addition, we also train a new DQN with only the reward R_i^{acc}, denoted as $DQN_{w/o-hole}$, which chooses seven view points on average since it tends to pick views with small holes for higher R_i^{acc}. The results in Tab 1 verify

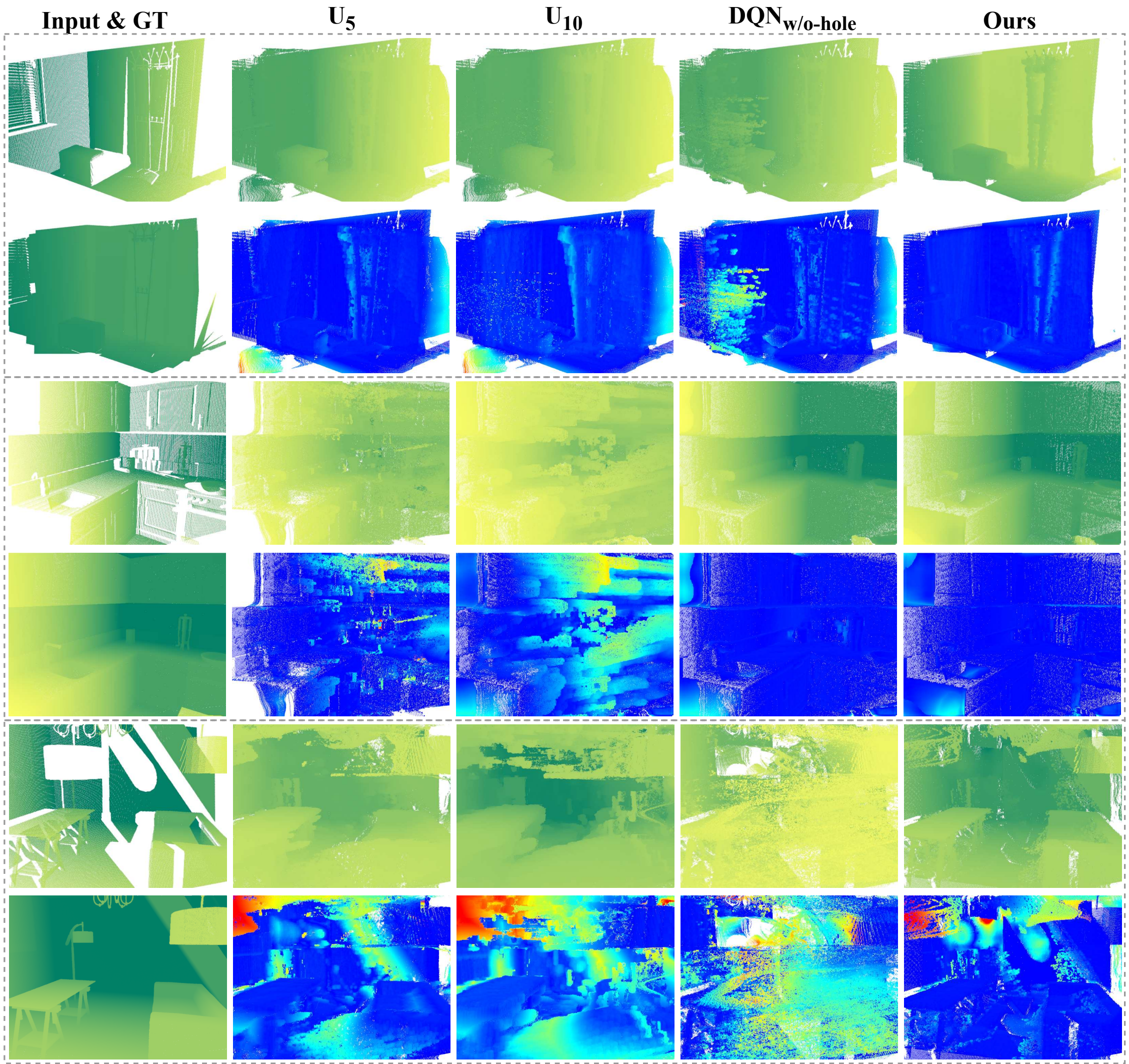


Figure 6. Comparisons on the variants of view path planning. Given different inputs and the referenced groundtruth, we show the completion results of four different approaches, with the corresponding point cloud error maps below.

the efficiency of the reward R_i^{hole}. Visual comparison results on some sampled scenes are shown in Figure 6, where our proposed model results in much better appearances than others.

5. Conclusion

In this paper, we propose the first surface-generated approach for 3D scene completion from a single depth image. The missing 3D points are inferred by conducting completion on multi-view depth maps. To guarantee a more accurate and consistent output, a volume-guided view inpianting network is proposed. In addition, a deep reinforcement learning framework is devised to seek the optimal view path to contribute the best result in accuracy. The experiments demonstrate that our model is the best choice and significantly outperforms existing methods. There are three research directions worth further exploration in the future: 1) how to make use of the texture information from the input RGBD images to achieve more accurate depth inpainting; 2) how to do texture completion together with the depth inpainting, to output a complete textured 3D scene; 3) how to guarantee a watertight completion.

Acknowledgements

We thank the anonymous reviewers for the insightful and constructive comments.This work was funded in part by The Pearl River Talent Recruitment Program Innovative and Entrepreneurial Teams in 2017 under grant No. 2017ZT07X152, Shenzhen Fundamental Research Fund under grants No. KQTD2015033114415450 and No. ZDSYS201707251409055, and by the National Natural Science Foundation of China under Grant 91748104, Grant 61632006, Grant 61751203.

References

[1] Paul S Blaer and Peter K Allen. Data acquisition and view planning for 3-d modeling tasks. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 417–422. IEEE, 2007.

[2] Kang Chen, Yu-Kun Lai, Yu-Xin Wu, Ralph Martin, and Shi-Min Hu. Automatic semantic modeling of indoor scenes from low-quality rgb-d data using contextual information. *ACM Trans. Graph.*, 33(6):208:1–208:12, Nov. 2014.

[3] Weihai Chen, Haosong Yue, Jianhua Wang, and Xingming Wu. An improved edge detection algorithm for depth map inpainting. *Optics and Lasers in Engineering*, 55:69–77, 2014.

[4] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European conference on computer vision*, pages 628–644. Springer, 2016.

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

[6] Angela Dai, Daniel Ritchie, Martin Bokeloh, Scott Reed, Jürgen Sturm, and Matthias Nießner. Scancomplete: Large-scale scene completion and semantic segmentation for 3d scans. *computer vision and pattern recognition*, 2018.

[7] David Doria and Richard J Radke. Filling large holes in lidar data by inpainting depth gradients. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 65–72. IEEE, 2012.

[8] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *CVPR*, volume 2, page 6, 2017.

[9] Martin Garbade, Johann Sawatzky, Alexander Richard, and Juergen Gall. Two stream 3d semantic scene completion. *arXiv preprint arXiv:1804.03550*, 2018.

[10] Josselin Gautier, Olivier Le Meur, and Christine Guillemot. Depth-based image completion for view synthesis. In *3DTV Conference: The True Vision-capture, Transmission and Display of 3D Video (3DTV-CON), 2011*, pages 1–4. IEEE, 2011.

[11] Ruiqi Guo, Chuhang Zou, and Derek Hoiem. Predicting complete 3d models of indoor scenes. *arXiv preprint arXiv:1504.02437*, 2015.

[12] Yu-Xiao Guo and Xin Tong. View-volume network for semantic scene completion from a single depth image. In *IJCAI 2018: 27th International Joint Conference on Artificial Intelligence*, pages 726–732, 2018.

[13] Saurabh Gupta, Pablo Andrés Arbeláez, Ross B. Girshick, and Jitendra Malik. Aligning 3d models to RGB-D images of cluttered scenes. In *CVPR*, pages 4731–4740. IEEE Computer Society, 2015.

[14] 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 *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2017.

[15] Daniel Herrera, Juho Kannala, Janne Heikkilä, et al. Depth map inpainting under a second-order smoothness prior. In *Scandinavian Conference on Image Analysis*, pages 555–566. Springer, 2013.

[16] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics (TOG)*, 36(4):107, 2017.

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

[18] Vladimir G Kim, Wilmot Li, Niloy J Mitra, Siddhartha Chaudhuri, Stephen DiVerdi, and Thomas Funkhouser. Learning part-based templates from large collections of 3d shapes. *ACM Transactions on Graphics (TOG)*, 32(4):70, 2013.

[19] Yangyan Li, Xiaokun Wu, Yiorgos Chrysathou, Andrei Sharf, Daniel Cohen-Or, and Niloy J Mitra. Globfit: Consistently fitting primitives by discovering global relations. In *ACM Transactions on Graphics (TOG)*, volume 30, page 52. ACM, 2011.

[20] Guilin Liu, Fitsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. *arXiv preprint arXiv:1804.07723*, 2018.

[21] Junyi Liu, Xiaojin Gong, and Jilin Liu. Guided inpainting and filtering for kinect depth maps. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 2055–2058. IEEE, 2012.

[22] Miaomiao Liu, Xuming He, and Mathieu Salzmann. Building scene models by completing and hallucinating depth and semantics. In *European Conference on Computer Vision*, pages 258–274. Springer, 2016.

[23] Shice Liu, Yu Hu, Yiming Zeng, Qiankun Tang, Beibei Jin, Yinhe Han, and Xiaowei Li. See and think: Disentangling semantic scene completion. In *NIPS 2018: The 32nd Annual Conference on Neural Information Processing Systems*, 2018.

[24] Kok-Lim Low and Anselmo Lastra. An adaptive hierarchical next-best-view algorithm for 3d reconstruction of indoor scenes. In *Proceedings of 14th Pacific Conference on Computer Graphics and Applications (Pacific Graphics 2006)*, pages 1–8, 2006.

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

[26] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

[27] Suryanarayana M Muddala, Marten Sjostrom, and Roger Olsson. Depth-based inpainting for disocclusion filling. In *3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON), 2014*, pages 1–4. IEEE, 2014.

[28] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2536–2544, 2016.

[29] Charles R Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas J Guibas. Volumetric and multi-view cnns for object classification on 3d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5648–5656, 2016.

[30] Jason Rock, Tanmay Gupta, Justin Thorsen, JunYoung Gwak, Daeyun Shin, and Derek Hoiem. Completing 3d object shape from one depth image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2484–2493, 2015.

[31] Ruwen Schnabel, Patrick Degener, and Reinhard Klein. Completion and reconstruction with primitive shapes. In *Computer Graphics Forum*, volume 28, pages 503–512. Wiley Online Library, 2009.

[32] 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 Trans. Graph.*, 31(6):136:1–136:11, Nov. 2012.

[33] Abhishek Sharma, Oliver Grau, and Mario Fritz. Vconv-dae: Deep volumetric shape learning without object labels. In *European Conference on Computer Vision*, pages 236–250. Springer, 2016.

[34] Chao-Hui Shen, Hongbo Fu, Kang Chen, and Shi-Min Hu. Structure recovery by part assembly. *ACM Transactions on Graphics (TOG)*, 31(6):180, 2012.

[35] 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.

[36] Shuran Song, Fisher Yu, Andy Zeng, Angel X. Chang, Manolis Savva, and Thomas A. Funkhouser. Semantic scene completion from a single depth image. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 190–198, 2017.

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

[38] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik G. Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proc. ICCV*, 2015.

[39] Minhyuk Sung, Vladimir G Kim, Roland Angst, and Leonidas Guibas. Data-driven structural priors for shape completion. *ACM Transactions on Graphics (TOG)*, 34(6):175, 2015.

[40] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Multi-view 3d models from single images with a convolutional network. In *European Conference on Computer Vision*, pages 322–337. Springer, 2016.

[41] Ali K Thabet, Jean Lahoud, Daniel Asmar, and Bernard Ghanem. 3d aware correction and completion of depth maps in piecewise planar scenes. In *Asian Conference on Computer Vision*, pages 226–241. Springer, 2014.

[42] Duc Thanh Nguyen, Binh-Son Hua, Khoi Tran, Quang-Hieu Pham, and Sai-Kit Yeung. A field model for repairing 3d shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5676–5684, 2016.

[43] Shubham Tulsiani, Tinghui Zhou, Alexei A. Efros, and Jitendra Malik. Multi-view supervision for single-view reconstruction via differentiable ray consistency. *CoRR*, abs/1704.06254, 2017.

[44] Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *AAAI*, volume 2, page 5. Phoenix, AZ, 2016.

[45] Jacob Varley, Chad DeChant, Adam Richardson, Joaquín Ruales, and Peter Allen. Shape completion enabled robotic grasping. In *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*, pages 2442–2447. IEEE, 2017.

[46] Chuan Wang, Haibin Huang, Xiaoguang Han, and Jue Wang. Video inpainting by jointly learning temporal structure and spatial details. *arXiv preprint arXiv:1806.08482*, 2018.

[47] Yida Wang, David Joseph Tan, Nassir Navab, and Federico Tombari. Adversarial semantic scene completion from a single depth image. In *2018 International Conference on 3D Vision (3DV)*, pages 426–434. IEEE, 2018.

[48] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Van Hasselt, Marc Lanctot, and Nando De Freitas. Dueling network architectures for deep reinforcement learning. *arXiv preprint arXiv:1511.06581*, 2015.

[49] Shihao Wu, Wei Sun, Pinxin Long, Hui Huang, Daniel Cohen-Or, Minglun Gong, Oliver Deussen, and Baoquan Chen. Quality-driven poisson-guided autoscanning. *ACM Transactions on Graphics*, 33(6), 2014.

[50] Kai Xu, Yifei Shi, Lintao Zheng, Junyu Zhang, Min Liu, Hui Huang, Hao Su, Daniel Cohen-Or, and Baoquan Chen. 3d attention-driven depth acquisition for object identification. *ACM Transactions on Graphics (TOG)*, 35(6):238, 2016.

[51] Hongyang Xue, Shengming Zhang, and Deng Cai. Depth image inpainting: Improving low rank matrix completion with low gradient regularization. *IEEE Transactions on Image Processing*, 26(9):4311–4320, 2017.

[52] Xin Yang, Yuanbo Wang, Yaru Wang, Baocai Yin, Qiang Zhang, Xiaopeng Wei, and Hongbo Fu. Active object reconstruction using a guided view planner. *arXiv preprint arXiv:1805.03081*, 2018.

[53] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 767–783, 2018.

[54] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. *arXiv preprint arXiv:1806.03589*, 2018.

[55] Hai-Tao Zhang, Jun Yu, and Zeng-Fu Wang. Probability contour guided depth map inpainting and superresolution using non-local total generalized variation. *Multimedia Tools and Applications*, 77(7):9003–9020, 2018.

[56] Liang Zhang, Le Wang, Xiangdong Zhang, Peiyi Shen, Mohammed Bennamoun, Guangming Zhu, Syed Afaq Ali Shah, and Juan Song. Semantic scene completion with dense crf from a single depth image. *Neurocomputing*, 318:182–195, 2018.

[57] Yinda Zhang and Thomas Funkhouser. Deep depth completion of a single rgb-d image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 175–185, 2018.

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

[59] Yajie Zhao, Weikai Chen, Jun Xing, Xiaoming Li, Zach Bessinger, Fuchang Liu, Wangmeng Zuo, and Ruigang Yang. Identity preserving face completion for large ocular region occlusion. *arXiv preprint arXiv:1807.08772*, 2018.

[60] Qian-Yi Zhou and Vladlen Koltun. Dense scene reconstruction with points of interest. *ACM Transactions on Graphics (ToG)*, 32(4):112, 2013.