High-Resolution Shape Completion Using Deep Neural Networks for Global Structure and Local Geometry Inference

Xiaoguang Han[1,*] Zhen Li[1,*] Haibin Huang[2] Evangelos Kalogerakis[2] Yizhou Yu[1]
[1]The University of Hong Kong [2]University of Massachusetts, Amherst

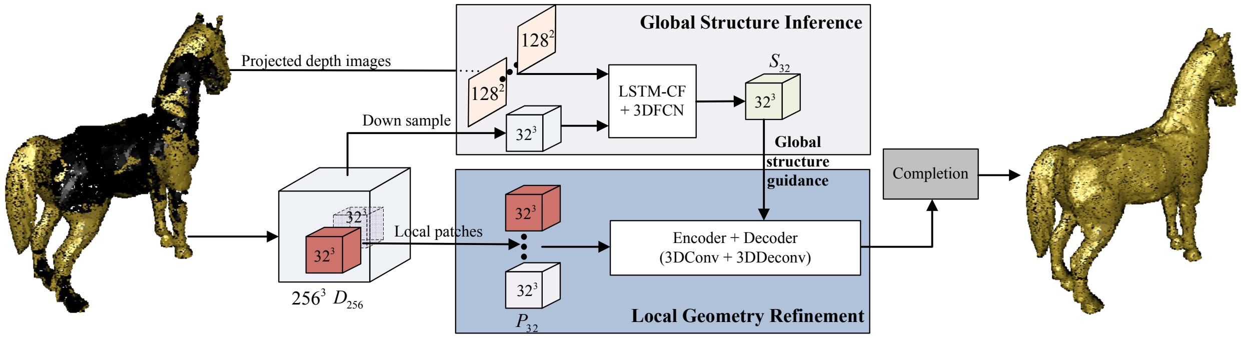

Figure 1: Pipeline of our high-resolution shape completion method. Given a 3D shape with large missing regions, our method outputs a complete shape through global structure inference and local geometry refinement. Our architecture consists of two jointly trained sub-networks: one network predicts the global structure of the shape while the other locally generates the repaired surface under the guidance of the first network.

Abstract

We propose a data-driven method for recovering missing parts of 3D shapes. Our method is based on a new deep learning architecture consisting of two sub-networks: a global structure inference network and a local geometry refinement network. The global structure inference network incorporates a long short-term memorized context fusion module (LSTM-CF) that infers the global structure of the shape based on multi-view depth information provided as part of the input. It also includes a 3D fully convolutional (3DFCN) module that further enriches the global structure representation according to volumetric information in the input. Under the guidance of the global structure network, the local geometry refinement network takes as input local 3D patches around missing regions, and progressively produces a high-resolution, complete surface through a volumetric encoder-decoder architecture. Our method jointly trains the global structure inference and local geometry refinement networks in an end-to-end manner. We perform qualitative and quantitative evaluations on six object categories, demonstrating that our method outperforms existing state-of-the-art work on shape completion.

1. Introduction

Inferring geometric information for missing regions of 3D shapes is a fundamental problem in the fields of computer vision, graphics and robotics. With the increasing availability of consumer depth cameras and geometry acquisition devices, robust reconstruction of complete 3D shapes from noisy, partial geometric data remains a challenging problem. In particular, a significant complication is the existence of large missing regions in the acquired 3D data due to occlusions, reflective material properties, and insufficient lighting conditions. Traditional geometry-based methods, such as Poisson surface reconstruction ([12]), are only able to handle relatively small gaps in the acquired 3D data. Unfortunately, these methods often fail to repair large missing regions. Learning-based approaches are more suitable for this task because of their ability to learn powerful 3D shape priors from large online 3D model collections (e.g., ShapeNet, Trimble Warehouse) for repairing such missing regions.

In the past, volumetric convolutional networks have been utilized [8] to learn a mapping from an incomplete 3D shape to a complete one, where both the input and output shapes are represented with voxel grids. Due to the high computational

[*] equal contribution

cost and memory requirement of three-dimensional convolutions, the resolution of the voxel grids used in these methods is severely limited (32^3 in most cases). Such a coarse representation gives rise to loss of surface details as well as low-resolution, implausible outputs. Post-processing methods, such as volumetric patch synthesis, could be applied to refine the shape, yet producing a high-quality shape still remains challenging as synthesis starts from a low-resolution intermediate result in the first place.

In this paper, we propose a deep learning framework pursuing high-resolution shape completion through joint inference of global structure and local geometry. Specifically, we train a global structure inference network including a 3D fully convolutional (3DFCN) module and a view-based long short-term memorized context fusion module (LSTM-CF). The representation generated from these modules encodes the inferred global structure of the shape that needs to be repaired. Under the guidance of the global structure inference network, a 3D encoder-decoder network reconstructs and fills missing surface regions. This second network operates at the local patch level so that it can synthesize detailed geometry. Our method jointly trains these two sub-networks so that it is not only able to infer an overall shape structure but also refine local geometric details on the basis of the recovered structure and context.

Our method utilizes the trained deep model to convert an incomplete point cloud into a complete 3D shape. The missing regions are progressively reconstructed patch by patch starting from their boundaries. Experimental results demonstrate that our method is capable of performing high-quality shape completion. Qualitative and quantitative evaluations also show that our algorithm outperforms existing state-of-the-art methods.

In summary, this paper has the following contributions:

- A novel global structure inference network based on a 3D FCN and LSTM. It is able to map an incomplete input shape to a representation encoding a complete global structure.
- A novel patch-level 3D CNN for local geometry refinement under the guidance of our global structure inference network. Our patch-level network is able to perform detailed surface synthesis from the starting point of a low-resolution voxel representation.
- A pipeline for jointly training the global structure and local geometry inference networks in an end-to-end manner.

2. Related work

There exist a large body of work on shape reconstruction from an incomplete point cloud. A detailed survey can be referred to [4].

Geometric approaches. By assuming local surface or volumetric smoothness, a number of geometry-based methods ([12] [26] [30]) can successfully recover the underlying surface in the case of small gaps. To fill larger missing regions, some methods employ hand-designed heuristics for particular shape categories. For example, Schnabel *et al.* [22] developed an approach to reconstruct CAD objects from incomplete point clouds, under the assumption that the shape is composed of many primitives (e.g., planes, cylinders, cones etc.). Li *et al.* [15] further considered geometric relationships (eg. orientation, placement, equality, etc.) between primitives in the reconstruction procedure. For objects with arterial-like structures, Li *et al.* [14] proposed snake deformable models and successfully recovered the topology and geometry simultaneously from a noisy and incomplete input. Based on the observation that urban objects are usually made of non-local repetitions, many methods ([20] [32]) attempt to discover symmetries from input data and use them to complete the unknown geometry. Harary *et al.* [11] also utilizes self-similarities to recover shape surfaces. Compared to these techniques, we propose a learning-based approach that learns a generic 3D shape prior for reconstruction without resorting to hand-designed heuristics or strict geometric assumptions.

Template-based approaches. Another commonly used strategy is to resort to deformable templates or nearest-neighbors to reconstruct an input shape. One simple approach is to retrieve the most similar shape from a database and use it as a template that can be deformed to fit the input raw data ([19] [21]). These template-based approaches usually require user interaction to specify sparse correspondences ([19]) or result in wrong structure ([21]) especially for complex input. These approaches can also fail when the input does not match well with the template, which often happens due to the limited capacity of the shape database. To address this issue, some recent works ([24] [25]) involved the concept of part assembly. They can successfully recover the underlying structure of a partial scan by solving a combinatorial optimization problem that aims to find the best part candidates from a database as well as their combination. However, these methods also have a number of limitations. First, each shape in the database needs to be accurately segmented and labeled. Second, for inputs with complicated structure, these methods may fail to find the global optimum due to the large solution space. Lastly, even if coarse structure is well recovered, obtaining the exact underlying geometry for missing regions remains challenging especially when the input geometry does not match well any shape parts in the database.

Deep learning-based methods. Recently, 3D convolutional networks have been proposed for shape completion.

Wu *et al.* [31] learn a probability distribution over binary variables representing voxel occupancy in a 3D grid based on Convolutional Deep Belief Networks (CDBNs). CDBNs are generative models that can also be applied for shape completion. Nguyen *et al.* [18] combines CDBNs and Markov Random Fields (MRFs) to formulate shape completion as a Maximum a Posteriori (MAP) inference problem. More recent methods employ encoder-decoder networks that are trained end-to-end for shape reconstruction [23, 28]. However, all these techniques operate on low-resolution grids (30^3 voxels to represent global shape) due to the high computational cost of convolution in three dimensions. The recent work of Dai *et al.* [8] is most related to ours. It proposes a 3D Encoder-Predictor Network (EPN) to infer a coarse shape with complete structure, which is then further refined through nearest-neighbor-based volumetric patch synthesis. Our method also learns a global shape structure model. However, in contrast to Dai *et al.*, we also learn a *local* encoder-predictor network to perform patch-level surface inference. Our network produces a more detailed output in a much higher resolution (256^3 grid) by processing local shape patches through this network (32^3 patches cropped from the 256^3 grid). Local surface inference is performed under the guidance of our global structure network that captures the necessary contextual information to achieve a globally consistent, and at the same time, high-resolution reconstruction.

3. Overview

Given a partial scan or an incomplete 3D object as input, our method aims to generate a complete object as output. At a high level, our pipeline is similar to PatchMatch-based image completion [1]. Starting from the boundary of missing regions, our method iteratively extends the surface into these regions, and at the same time updates their boundary for further completion until these missing regions are filled. To infer new geometry along the boundary, instead of retrieving the best matching patch from a large database as in [11], we designed a local surface inference model based on volumetric encoder-decoder networks.

The overall pipeline of our method is shown in Figure 1. Our shape completion is performed patch-by-patch. At first, the input point cloud is voxelized in a 256^3 grid, and then 32^3 patches are extracted along the boundary of missing regions. Our local surface inference network maps the volumetric distance field of a surface, potentially with missing regions, to an implicit representation (0 means inside while 1 means outside) of a complete shape (Section 4.2). The distance field can then be extracted with Marching Cubes [17]. To improve the global consistency of local predictions during shape completion, another global structure inference network is designed to infer complete global shape structure and guide the local geometry refinement network. Our global structure inference network generates a 32^3 shape representation, capturing its overall coarse structure, using both view-based and volumetric deep neural networks. To make use of high-resolution shape information, depth images generated over the six faces of the bounding cube are considered as one type of input data (view-based input). Six 2D feature representations of these depth images are extracted through six parallel streams of 2D convolutional and LSTM recurrent layers. These 2D feature maps are assembled into a 32^3 feature representation, which is fused with another volumetric-based feature representation extracted through 3D convolutional layers operating on volumetric input. The resulting fused representation is used for final voxel-wise prediction. Both our global and local network are trained jointly (Section 4).

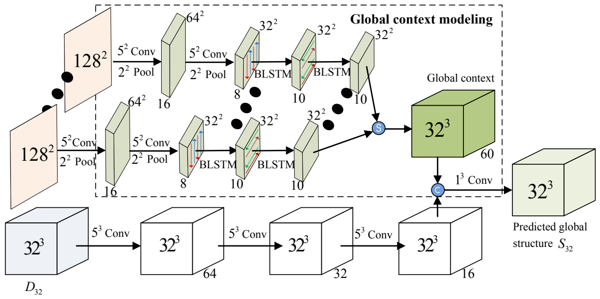

Figure 2: The inputs for global structure inference consist of projected depth images with size 128^2 and down-sampling voxelized point cloud D_{32} with resolution 32^3. “S” stands for the feature stack operation. “C” stands for the concatenate operation.

4. Network Architecture

The incomplete point cloud is represented as a 256^3 volumetric distance field (Section 5), denoted as D_{256}. Our deep neural network is composed of two sub-networks. One infers underlying global structure from a down-sampled version (32^3) of D_{256}. The down-sampled field is denoted as D_{32} and the inferred result is denoted as S_{32}. Another sub-network infers high-resolution local geometry within 32^3 volumetric patches (denoted as P_{32}) cropped from D_{256}.

4.1. Global Structure Inference

Although an incomplete point cloud has missing data, most often it still provides adequate information for recognizing the object category and understanding its global structure (i.e. object parts and their spatial layout). Such categorical and structural information provides a global context that can help resolve ambiguities arising in local shape completion. Therefore, there is a need to automatically infer the global structure of the underlying object given an incomplete point cloud of the object.

To this end, we design a novel deep network for global structure inference. This network takes two sets of input data. Since processing D_{256} would be too time- and memory-consuming, the first set of input is D_{32}, the down-sampled distance field of the input point cloud. To compensate for the low resolution of D_{32}, the second set of input data consists of six 128^2 depth images, each obtained as an orthographic depth image of the point cloud over one of the six faces of its bounding box. Inspired by the LSTM-CF model [16] and ReNet [29], each depth image passes through two convolutional layers, each of which is followed by a max-pooling layer. The feature map from the second pooling layer is further fed into the ReNet, which consists of cascaded vertical and horizontal bidirectional LSTM (BLSTM) layers for global context modeling,

$$\begin{aligned} h^v_{i,j} &= \text{BLSTM}(h^v_{i,j-1}, f_{i,j}), \quad \text{for } j = 1, \dots, 32; \\ h^h_{i,j} &= \text{BLSTM}(h^h_{i-1,j}, h^v_{i,j}), \quad \text{for } i = 1, \dots, 32, \end{aligned} \tag{1}$$

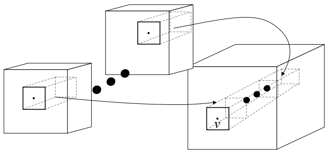

Figure 3: 3D feature map assembles from six 2D feature maps.

where $f_{i,j}$ is the feature map from the second max pooling layer, $h^v_{i,j}$ and $h^h_{i,j}$ are the output maps of the vertical and horizontal BLSTMs, respectively. Afterwards, 2D output maps from the six horizontal BLSTMs are assembled together into a 3D feature map with size 32^3 as follows. A voxel v in the 3D feature map is first projected onto the six faces of its bounding box. The projected location on each face is used to look up the feature vector at that location in the corresponding 2D output map. The six retrieved feature vectors from the six 2D maps are concatenated together as the feature for v (in Fig. 3 for details). In parallel, D_{32} is fed into three 3D convolutional layers with the same resolution and a 3D feature map with size 32^3 is obtained after the third layer. Finally, both 3D features maps from the two parallel branches are concatenated and flow into a 3D convolutional layer with $1 \times 1 \times 1$ kernels for voxel-wise binary prediction.

It is worth mentioning that the occupancy grids are sparse when used for representing voxelized point clouds. This results in highly uneven distributions of two-class data (inside and outside). For instance, the ratio between the inside and outside voxels for the 'chair' category is 25. Meanwhile, precision and recall both play an importance role in shape completion, especially for inside voxels. To address this problem, we add the AUC loss to the conventional cross-entropy loss for classification [5, 7]. According to [5], the AUC of a predictor f is defined as (here f is the final classification layer with softmax activation illustrated in Fig. 2) $\text{AUC}(f) = P(f(t_0) < f(t_1) | t_0 \in D^0, t_1 \in D^1)$, where D^0, D^1 are the samples with groundtruth labels 0 and 1, respectively. Its unbiased estimator, i.e. Wilcoxon-Man-Whitney statistics, is $n_0 n_1 \text{AUC}(f) = \sum_{t_0 \in D^0} \sum_{t_1 \in D^1} I[f(t_0) < f(t_1)]$, where $n_0 = |D^0|, n_1 = |D^1|$, and I is the indicator function. In order to add the noncontinuous AUC loss to the continuous cross-entropy loss and optimize the combined loss through gradient decent, we consider an approximation of the AUC loss by a polynomial with degree k [5], i.e.

$$n_0 n_1 \text{loss}_{\text{auc}} = \sum_{t_0 \in D^0} \sum_{t_1 \in D^1} \sum_{k=0}^{d} \sum_{l=0}^{k} \alpha_{kl} f(t_1)^l f(t_0)^{k-1}$$

where $\alpha_{kl} = c_k C^l_k (-1)^{k-l}$ is a constant. Thus, our global loss function can be formulated as

$$\text{loss}_{\text{global}} = -\frac{1}{N} \sum_i s^*_i \log(s_i) - \lambda_1 \text{loss}_{\text{auc}}, \tag{2}$$

where s_i stands for the predicted probability of a label, s^*_i stands for a ground-truth label, N is the number of voxels and λ_1 is a balancing weight between the cross-entropy loss and the AUC loss.

4.2. Local Geometry Refinement

We further propose a deep neural network for inferring the high-resolution geometry within a local 3D patch P_{32} (each 32^3 patch is a crop from D_{256}) along the boundary of missing regions. Instead of a 3D fully convolutional network, we exploit an encoder-decoder network architecture to achieve this goal. This is because local patches are sampled along the boundary of missing regions, the surface inside a local patch usually suffers from a larger portion of missing data (on average 50%) than the global shape and fully connected layers are better suited for higher-level inference.

As shown in Fig. 4, the first part of our network transforms the input patch into the latent space through a series of 3D convolutional and 3D max pooling layers. This encoding part is followed by two fully connected layers. The decoding part then achieves voxel-wise binary predictions with a series of 3D deconvolutions. In comparison to prior network designs [8, 23, 28], our network has a notable difference, which is the incorporation of global structure guidance. Given S_{32} generated by our global structure inference model, for each input patch P_{32} centered at (x, y, z) in D_{256}, we use S_{32} as guidance at two different places of the pipeline. First, a 8^3 patch centered at $(x/8, y/8, z/8)$ is cropped from S_{32} and passes through a 3D convolutional layer followed by 3D max pooling. The resulting 4^3 patch is concatenated with the 3D feature map at the end of the encoding part. Second, a 4^3 patch centered at $(x/8, y/8, z/8)$ is cropped from S_{32} and directly concatenated with the 4^3 feature map at the beginning of the decoding part. Similar to the loss function of the global structure inference network,

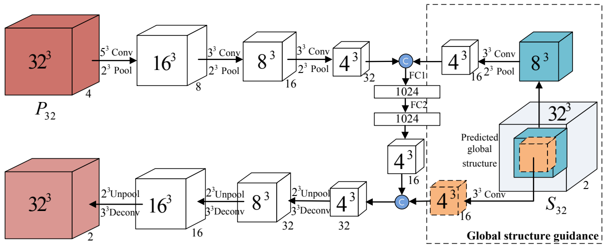

Figure 4: The inputs of local surface refinement network consist of local patches P_{32} with size 32^3 and output S_{32} with size 32^3 from the global structure inference network as global guidance.

the loss of our local geometry refinement network is defined as $\text{loss}_{\text{local}} = -\frac{1}{M}\sum_i p_i^* \log(p_i)$, where p_i is the predicted probability of a local geometry, p_i^* is a ground-truth label, and M is the number of voxels.

4.3. Network Training

Our two deep networks are trained in two phases. In the first phase, the global structure inference network is first trained alone. In the second phase, the local geometry refinement network is trained while the global structure inference network is being fine-tuned. As illustrated in Fig.4, a local patch from S_{32}, which is the output from the global structure inference network, flows into the local geometry refinement network as a global guidance, which is vital for shape completion. On the other side, local patch prediction results can benefit global structure inference as well, e.g., the refined disconnected regions between the side and leg of a chair can give feedback to global structure inference. Thus, due to the interactions between our global and local sub-networks, joint training is performed to improve their performance and robustness. Since the global network has been trained during the first phase, it is further fine-tuned during joint training in the second phase. The loss for such joint training is defined as follows.

$$\text{loss} = \text{loss}_{\text{local}} + \lambda_2 \text{loss}_{\text{global}} + \lambda_3\|\theta\|_2, \qquad (3)$$

where λ_2 is a balancing weight between the local and global loss functions, θ is the parameter vector (L2 norm is adopted for regression terms in our loss).

5. Training Data Generation

Our network has been tested on 6 categories of objects separately. Among them, ‘chairs’, ‘cars’, ‘guitars’, ‘sofas’, and ‘guns’ are from ShapeNet [6] and ‘animals’ were collected by ourselves. For each category from ShapeNet, we select a subset of models by removing repeated ones with very similar structures and thin models that cannot be well voxelized on a 32^3 grid. All animal models were also manually aligned in a coordinate system consistent with ShapeNet. To generate training samples, we simulate object scanning using an RGBD camera and create an incomplete point cloud for each model by fusing multiple partial scans with missing regions. On each created incomplete model, we randomly sample n patches along the boundary of missing regions (n is set to 50 for all object categories). The number of created training models for each category are shown in Table 1. Note that we create multiple scanned models by simulation from each original virtual model. Each point cloud is represented using a volumetric distance field. Note that, to make occupancy grid as dense as possible, different scaling factors are applied to models from different categories before voxelization. This is also the reason why we do not train a single network on all categories.

Table 1: Number of training samples

Category	Chair	Car	Guitar	Gun	Sofa	Animal
# Samples	1000x3	500x5	320x5	300x5	500x5	43x10

Virtual Scanning. We first generate depth maps by placing a virtual camera at 20 distinct viewpoints, which are vertices of a dodecahedron enclosing a 3D model. The camera is oriented towards the centroid of the 3D model. We then randomly select 3-5 viewpoints only to generate partial shapes simulating scans obtained through limited view access. On top of random viewpoint selection, we also randomly add holes and noise to the depth maps to simulate large occlusions or missing data due to specularities and scanner noise. The method in [27] is adopted to create holes. For each depth map, this method is run multiple times and super-pixels are generated at a different level of granularity each time. A set of randomly chosen superpixels are removed at each level of granularity to create holes at multiple scales. The resulting depth maps are then backprojected to the virtual model to form an incomplete point cloud. Note that missing shape regions will also exist due to self occlusions (since depth maps cannot cover the entire object surface).

Colored SDF + Binary Surface. Each point cloud is converted to a signed distance field as in [8]. To enhance the border between points with positive and negative distances, we employ colored SDF (CSDF), which maps negative distances (inside) to colors between cyan and blue and positive distances to colors between yellow and red. As a distance field makes missing parts less prominent, we also take the binary surface (i.e. occupancy grid of input points) as an additional input channel, denoted as BSurf. Thus, a point cloud is converted to a volumetric grid with four channels.

Projected Depth Images. Our global network takes 6 depth images as the second set of input data. These depth images are orthographic views of the point cloud from the 6 faces of the bounding cube. These depth images are enhanced with jet color mapping [10].

Patch Sampling. In general, there would be a large number of patches with similar local geometry if they were chosen randomly (for example, several chair patches would originate from flat or cylindrical surfaces). To avoid class imbalance and increase the diversity of training samples, we perform clustering on all sampled patches and only the cluster centers are chosen as training patches. Here we only use BSurf as the feature during patch clustering.

6. Shape Completion

During testing, given an incomplete point cloud P, as the first step, we apply our global structure inference network to generate a complete but coarse structure. As discussed earlier, starting from the boundary of missing regions, our method iteratively extends the surface into these regions until they are completely filled. In this paper, the method from [3] is used to detect the boundary of missing regions in a point cloud.

During each iteration, local 3D patches with a fixed size of overlap are chosen to cover all points on the boundary of missing regions. Our local geometry refinement network runs on these patches with the guidance from the inferred global structure to produce a voxel-wise probability map for each patch. The probability at each voxel indicates how likely that voxel belongs to the interior of the object represented by the input point cloud. For voxels covered by multiple overlapping patches, we directly average the corresponding probabilities in these patches. Then we transform the

simply deducting the probability values by 0.5. Marching Cubes [17] is then used to extract a partial mesh from the set of chosen patches and a new point set Q is evenly sampled over the partial mesh. We further remove the points in Q that lie very closely to P, and detect new boundary points from the remaining points in Q. Such detected boundary points form the new boundary of missing regions. The above steps are performed repeatedly until new boundary points cannot be found. In our experiments, 5 iterations are sufficient in most cases.

7. Experimental Results

Fig. 5 shows a gallery of results. For each object category, two models with different views are chosen. The incomplete point cloud and the repaired result are placed side by side.

7.1. Implementation

We jointly train the global and local networks for 20 epochs with an Adam optimizer [13]. We include one voxelized point cloud, six depth images associated with the point cloud and 50 local patches sampled from the voxel grid along missing regions in a single mini-batch. The balancing weights are set as follows: $\lambda_1 = 0.2$ and $\lambda_2 = \frac{2}{3}$. The regression weight and learning rate are set to 0.001 and 0.0001, respectively. Considering diverse scales in different object categories, we always pre-train our deep network for a specific object category from scratch using 2400 (800 original models with 3 different virtual scanning per model) chair models. Afterwards, we fine-tune the pre-trained model using training samples from that specific object category. Fine-tuning on each category needs about 4 hours to converge. On average, it takes 400ms to perform a forward pass through the global structure inference and local geometry refinement networks. Once we have the trained global and local networks, it takes around 60s to obtain a completed high-resolution shape. The detailed configuration of our global and local networks is illustrated in Figs. 2 and 4. The implementation is based on the publicly available Lasagne [9] library built on the Theano [2] platform, and network training is performed on a single NVIDIA GeForce GTX 1080.

7.2. Comparisons with existing methods

Let P_{true} be the ground-truth point cloud and $P_{complete}$ be the repaired point cloud. The normalized distance from $P_{complete}$ to P_{true} is used to measure the accuracy of the repaired point cloud. For each point $v \in P_{complete}$, we compute $dist(v, P_{true}) = min\{||v - q||, q \in P_{true}\}$. The average of these distances is finally normalized by the maximum shape diameter (denoted as dm) across all models in a given category. Following [25], we also use "completeness" to evaluate the quality of shape completion. Completeness records the fraction of points in P_{true} that are within distance α_{eval} of any point in $P_{complete}$. In our setting, α_{eval} is set to $0.001 * dm$. The average accuracy and completeness across all models from each object category are reported in Table 2. Specifically, we randomly select 200×3, 100×5, 64×5, 60×5, 100×5 and 8×10 samples as the testing set of chairs, cars, guitars, guns, sofas and animals, respectively.

To evaluate the accuracy and efficiency of the proposed method, we perform comparisons against existing state-of-the-art methods. Poisson surface reconstruction [12] is commonly used to construct 3D models from point clouds. However, it cannot successfully process point clouds with a large percentage of missing data.

Recently, a number of methods [23, 28, 8] attempt to perform shape completion at a coarse resolution (32^3) using 3DCNNs. The completeness and normalized distance of these methods are reported in Table 2, where '3D-EPN-unet' stands for the 3D Encoder-Predictor Network with U-net but without 3D classification in [8]. Note that their model with the 3D classification network is not available.

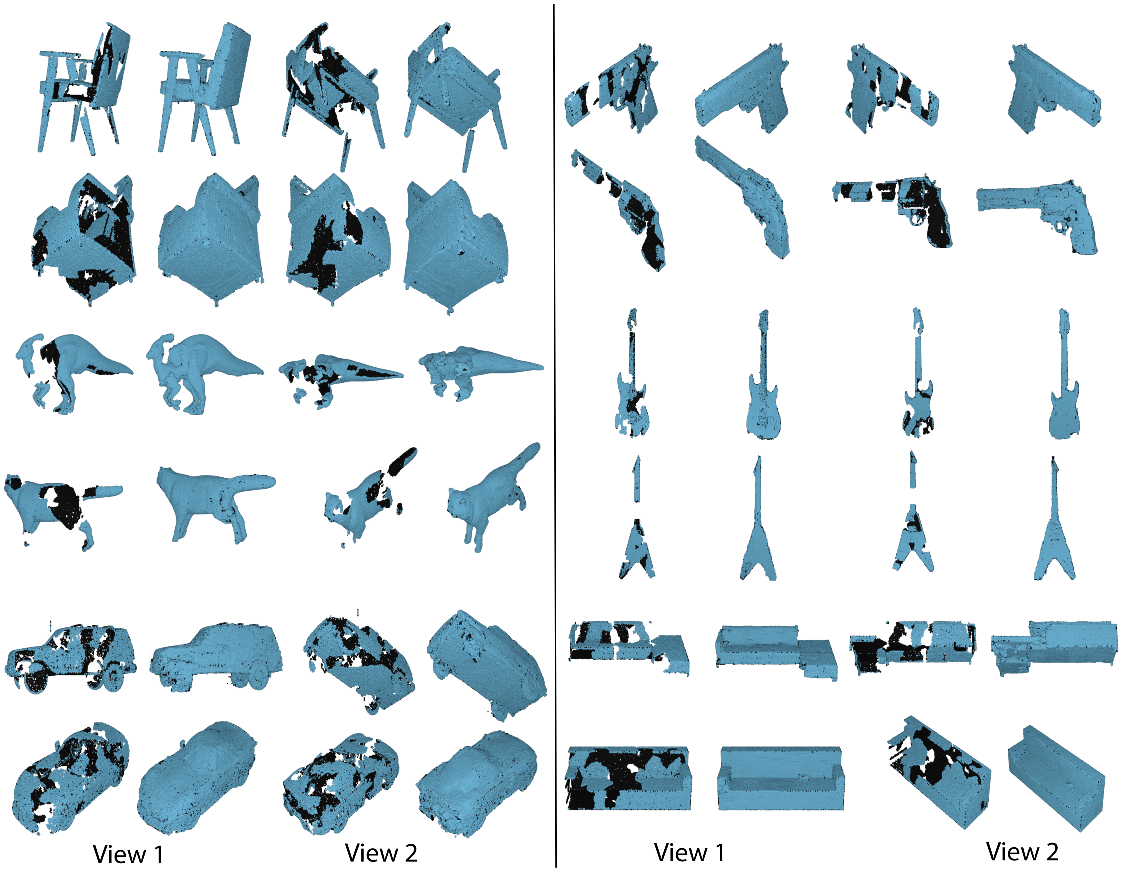

Figure 5: Gallery of final results. There are two models per category. For each model, the input and repaired point clouds are shown side by side from two different views.

'Vconv-dae' stands for the network from [23] and 'Varley *et al.*' stands for the network from [28]. For fair comparison, we retrained these networks using our training data and performed the evaluation on our testing data. Evaluation results on our global network are also reported. They demonstrate that our method outperforms all existing 32^3-level methods even without local geometry inference. To derive an upper bound on the completeness achievable by the methods with 32^3-level outputs, we subsample the distance field of P_{true} at resolution 32^3. We also create a point cloud from that subsampled field (called $P_{downsample}$), which represents a lower bound on the normalized distance that can be achieved by these methods (since these methods introduce additional errors in addition to downsampling). We report the evaluation results on the downsampled point clouds as a baseline for comparison. It can be verified that our results are significantly better than those from existing methods. As an intuitive comparison, Fig. 6 shows the outputs from 'Poisson', '$P_{downsample}$' and our method on two sampled models.

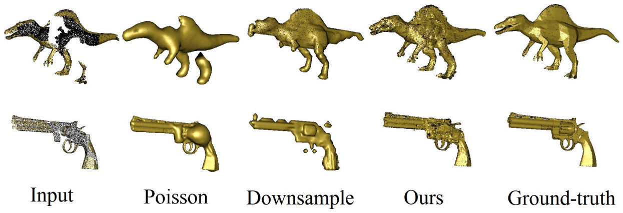

Figure 6: Sampled comparison results with other methods.

The method in [8] also proposes a way to achieve high-resolution completion by perfroming 3D patch synthesis as a post-processing step for shape refinement. In comparison to this approach, an important advantage of our method is that our local geometry refinement network directly utilizes the high-resolution information from the input point cloud, which makes the refined results more accurate. Another type of methods [24, 25] can also complete a shape with large missing regions. However, they require a large database with well-segmented objects. As the models in our datasets have not been segmented into parts, we do not conduct comparisons against such methods since a fair comparison seems out of reach.

Table 2: Performance Comparison. For each category and each method, we show the value of *completeness/normalized dist*.

Category	Input	Varley *et al.*	Vcov-dae	3D-EPN-unet	Our global network	Downsample	Poisson	Our whole network
Chair	71.5%/0	35.8%/0.022	47.7%/0.020	58.5%/0.018	70.1%/0.0108	73.32%/0.0144	87.61%/0.00925	**97.25%/0.00398**
Car	69.1%/0	42.3%/0.014	64.6%/0.013	66.4%/0.0092	81.8%/0.0081	84.35%/0.00756	82.18%/0.0147	**95.88%/0.00312**
Guitar	85.7%/0	45.8%/0.013	56.6%/0.011	62.9%/0.0092	69.7%/0.00626	72.16%/0.00675	88.4%/0.148	**94.35%/0.00248**
Sofa	72.31%/0	18.1%/0.024	58.4%/0.019	62.8%/0.012	77.0%/0.00845	85.15%/0.00615	82.78%/0.027	**95.97%/0.00217**
Gun	62.7%/0	28.5%/0.0165	39.1%/0.0134	49.2%/0.0132	54.3%/0.0091	56.4%/0.0102	77.68%/0.0114	**98.58%/0.00281**
Animal	69.05%/0	35.6%/0.0257	47.8%/0.0229	56.1%/0.019	82.4%/0.01137	85.14%/0.0114	88.88%/0.0567	**95.53%/0.00363**

7.3. Ablation Study

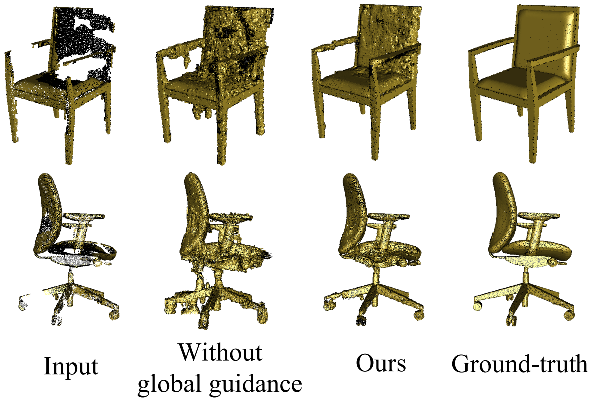

Figure 7: Completion results by using our model with and without global guidance.

To discover the vital elements in the success of our proposed model for shape completion, we conduct an ablation study by removing or replacing individual components in our model trained with 1000×3 chair samples, among which 800×3 samples form the training set and 200×3 samples form the testing set. Specifically, for the global structure inference network, we have tested its performance without the AUC loss, high-resolution depth images, or global context modeling using BLSTM. In addition, we have also tested the global model where the 3DFCN branch only takes CSDF or BSurf as the input to figure out the importance of different input channels. In addition, an encoder-decoder network is used to replace the final 1x1x1 convolutional layer in the global network. For the local geometry refinement network, we have tested its performance by removing the global guidance. The results are presented in Table 3. Because of class imbalance, we use the F1-score of the inside labels as the performance measure of the global network. We directly use classification accuracy to evaluate the performance of the local network since class imbalance is not a concern in this case. Note that the ground truth for the global network is defined on a coarse (32^3) grid while the ground truth for the local network is defined on a high-resolution (256^3) grid.

Table 3: Ablation Study

Network	Component	Performance
Global Structure Inference	w/o AUC loss	0.904
	w/o depth images	0.877
	w/o BLSTM context modeling	0.896
	w/o BSurf channel	0.90
	BSurf channel only	0.836
	Replace 1x1x1conv with encoder-decoder	0.818
	Complete global network	**0.926**
Local Geometry Refinement	Without global guidance	0.912
	With global guidance	**0.961**

During this ablation study, we find that CSDF, BSurf and high-resolution depth images are all necessary for global structure inference as the performance drops to 0.90 if the input to the 3DFCN branch is CSDF only (without the BSurf channel) and the performance drops to 0.877 if the entire 2D branch taking depth images is eliminated. In addition, the most effective components in our network are BLSTM based context modeling and the AUC loss as the performance drops to 0.896 and 0.904, respectively, without either of them. Furthermore, the performance drops to 0.818 if the final 1x1x1 convolutional layer in the global network is replaced with an encoder-decoder network perhaps because the 1x1x1 convolutional layer can better exploit spatial contextual information. For the local geometry refinement network, we find that global guidance is a vital component as the performance of the local network drops to 0.912 without it. This is also verified by Fig. 7, where the final completed point cloud using our model with and without global guidance on two sample models are illustrated.

8. Conclusion

We have presented an effective framework for completing partial shapes through 3D CNNs. Our results show that our method significantly improves the performance of existing state-of-the-art methods. We also believe jointly training global and local networks is a promising direction.

Acknowledgements. The authors would like to thank the reviewers for their constructive comments. Evangelos Kalogerakis acknowledges support from NSF (CHS-1422441, CHS-1617333).

References

[1] C. Barnes, E. Shechtman, A. Finkelstein, and D. Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics*, 28(3):24, 2009.

[2] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. J. Goodfellow, A. Bergeron, N. Bouchard, and Y. Bengio. Theano: new features and speed improvements. 2012.

[3] G. H. Bendels, R. Schnabel, and R. Klein. Detecting holes in point set surfaces. *WSCG 2006*.

[4] M. Berger, A. Tagliasacchi, L. Seversky, P. Alliez, J. Levine, A. Sharf, and C. Silva. State of the art in surface reconstruction from point clouds. In *EUROGRAPHICS star reports*, volume 1, pages 161–185, 2014.

[5] T. Calders and S. Jaroszewicz. Efficient auc optimization for classification. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 42–53. Springer, 2007.

[6] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, et al. Shapenet: An information-rich 3d model repository. *1512.03012*, 2015.

[7] C. Cortes and M. Mohri. Auc optimization vs. error rate minimization. In *NIPS*, volume 9, page 10, 2003.

[8] A. Dai, C. R. Qi, and M. Nießner. Shape completion using 3d-encoder-predictor cnns and shape synthesis. *arXiv preprint arXiv:1612.00101*, 2016.

[9] S. Dieleman, J. Schlüter, and R. et al. Lasagne: First release. *Zenodo: Geneva, Switzerland*, 2015.

[10] A. Eitel, J. T. Springenberg, L. Spinello, M. Riedmiller, and W. Burgard. Multimodal deep learning for robust rgb-d object recognition. In *In Intelligent Robots and Systems 2015*, pages 681–687. IEEE, 2015.

[11] G. Harary, A. Tal, and E. Grinspun. Context-based coherent surface completion. *ACM Transactions on Graphics*, 33(1):5, 2014.

[12] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, volume 7, 2006.

[13] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[14] G. Li, L. Liu, H. Zheng, and N. J. Mitra. Analysis, reconstruction and manipulation using arterial snakes. *ACM Transactions on Graphics*, 2010.

[15] Y. Li, X. Wu, Y. Chrysathou, A. Sharf, D. Cohen-Or, and N. J. Mitra. Globfit: consistently fitting primitives by discovering global relations. In *ACM Transactions on Graphics*, volume 30, page 52. ACM, 2011.

[16] Z. Li, Y. Gan, X. Liang, Y. Yu, H. Cheng, and L. Lin. Lstm-cf: Unifying context modeling and fusion with lstms for rgb-d scene labeling. In *European Conference on Computer Vision*, pages 541–557. Springer, 2016.

[17] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *ACM siggraph computer graphics*, volume 21, pages 163–169. ACM, 1987.

[18] D. T. Nguyen, B.-S. Hua, M.-K. Tran, Q.-H. Pham, and S.-K. Yeung. A field model for repairing 3d shapes. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 5, 2016.

[19] M. Pauly, N. J. Mitra, J. Giesen, M. H. Gross, and L. J. Guibas. Example-based 3d scan completion. In *Symposium on Geometry Processing*, pages 23–32, 2005.

[20] M. Pauly, N. J. Mitra, J. Wallner, H. Pottmann, and L. J. Guibas. Discovering structural regularity in 3d geometry. In *ACM transactions on graphics*, volume 27, page 43. ACM, 2008.

[21] J. Rock, T. Gupta, J. Thorsen, J. Gwak, D. Shin, and D. Hoiem. Completing 3d object shape from one depth image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2484–2493, 2015.

[22] R. Schnabel, P. Degener, and R. Klein. Completion and reconstruction with primitive shapes. In *Computer Graphics Forum*, volume 28, pages 503–512, 2009.

[23] A. Sharma, O. Grau, and M. Fritz. Vconv-dae: Deep volumetric shape learning without object labels. 2016.

[24] C.-H. Shen, H. Fu, K. Chen, and S.-M. Hu. Structure recovery by part assembly. *ACM Transactions on Graphics*, 31(6):180, 2012.

[25] M. Sung, V. G. Kim, R. Angst, and L. Guibas. Data-driven structural priors for shape completion. *ACM Transactions on Graphics*, 34(6):175, 2015.

[26] A. Tagliasacchi, M. Olson, H. Zhang, G. Hamarneh, and D. Cohen-Or. Vase: Volume-aware surface evolution for surface reconstruction from incomplete point clouds. In *Computer Graphics Forum*, volume 30, pages 1563–1571, 2011.

[27] M. Van den Bergh, X. Boix, G. Roig, B. de Capitani, and L. Van Gool. Seeds: Superpixels extracted via energy-driven sampling. In *European conference on computer vision*, pages 13–26. Springer, 2012.

[28] J. Varley, C. DeChant, A. Richardson, A. Nair, J. Ruales, and P. Allen. Shape completion enabled robotic grasping. *arXiv preprint arXiv:1609.08546*, 2016.

[29] F. Visin, K. Kastner, K. Cho, M. Matteucci, A. Courville, and Y. Bengio. Renet: A recurrent neural network based alternative to convolutional networks. *arXiv preprint arXiv:1505.00393*, 2015.

[30] S. Wu, H. Huang, M. Gong, M. Zwicker, and D. Cohen-Or. Deep points consolidation. *ACM Transactions on Graphics*, 34(6):176, 2015.

[31] 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.

[32] Q. Zheng, A. Sharf, G. Wan, Y. Li, N. J. Mitra, D. Cohen-Or, and B. Chen. Non-local scan consolidation for 3d urban scenes. *ACM Transactions on Graphics*, 29(4):94, 2010.