PatchRD: Detail-Preserving Shape Completion by Learning Patch Retrieval and Deformation

Bo Sun[1(✉)], Vladimir G. Kim[2], Noam Aigerman[2], Qixing Huang[1], and Siddhartha Chaudhuri[2,3]

[1] UT Austin, Austin, USA
`bosun0711@gmail.com`
[2] Adobe Research, San Jose, USA
[3] IIT Bombay, Mumbai, India

Abstract. This paper introduces a data-driven shape completion approach that focuses on completing geometric details of missing regions of 3D shapes. We observe that existing generative methods lack the training data and representation capacity to synthesize plausible, fine-grained details with complex geometry and topology. Our key insight is to copy and deform patches from the partial input to complete missing regions. This enables us to preserve the style of local geometric features, even if it drastically differs from the training data. Our fully automatic approach proceeds in two stages. First, we learn to retrieve candidate patches from the input shape. Second, we select and deform some of the retrieved candidates to seamlessly blend them into the complete shape. This method combines the advantages of the two most common completion methods: similarity-based single-instance completion, and completion by learning a shape space. We leverage repeating patterns by retrieving patches from the partial input, and learn global structural priors by using a neural network to guide the retrieval and deformation steps. Experimental results show our approach considerably outperforms baselines across multiple datasets and shape categories. Code and data are available at https://github.com/GitBoSun/PatchRD.

1 Introduction

Completing geometric objects is a fundamental problem in visual computing with a wide range of applications. For example, when scanning complex geometric objects, it is always difficult to scan every point of the underlying object [33]. The scanned geometry usually contains various levels of holes and missing geometries, making it critical to develop high-quality geometry completion techniques [1,10,13,18,24,61,68]. Geometry completion is also used in interactive

| Input | PatchRD (Ours) | Generative Model [71] | Input | PatchRD (Ours) | Generative Model [71] |
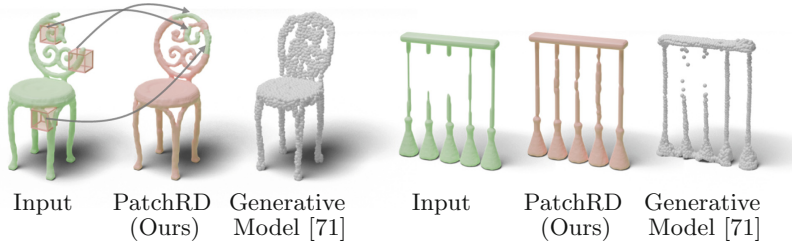
Fig. 1. We propose PatchRD, a non-parametric shape completion method based on patch retrieval and deformation. Compared with the parametric generation methods, our method is able to recover complex geometric details as well as keeping the global shape smoothness.

shape modeling [7], as a way to suggest additional content to add to a partial 3D object/scene. Geometry completion is challenging, particularly when the missing regions contain non-trivial geometric content.

Early geometry completion techniques focus on hole filling [1,10,13,18,24, 61,68]. These techniques rely on the assumption that the missing regions are simple surface patches and can be filled by smoothly extending hole regions. Filling regions with complex shapes rely on data priors. Existing approaches fall into two categories. The first category extracts similar regions from the input shape. The hypothesis is that a physical 3D object naturally exhibits repeating content due to symmetries and texture. While early works use user-specified rules to retrieve and fuse similar patches, recent works have studied using a deep network to automatically complete a single image or shape [20,21,62]. The goal of these approaches is to use different layers of the neural network (e.g., a convolutional neural network) to automatically extract repeating patterns. However, these approaches are most suitable when the repeating patterns are prevalent within the partial input. They cannot infer correlations between the missing surface and the observed surface (Fig. 1).

Another category [12,17,43,46,59,71,73,78] consists of data-driven techniques, which implicitly learn a parametric shape space model. Given an incomplete shape, they find the best reconstruction using the underlying generative model to generate the complete shape. This methodology has enjoyed success for specific categories of models such as faces [4,48,74,80] and human body shapes [1,26,30,38,45], but they generally cannot recover shape details due to limited training data and difficulty in synthesizing geometric styles that exhibit large topological and geometrical variations.

This paper introduces a shape completion approach that combines the strengths of the two categories of approaches described above. Although it remains difficult to capture the space of geometric details, existing approaches can learn high-level compositional rules such as spatial correlations of geometric primitives and parts among both the observed and missing regions. We propose

to leverage this property to guide similar region retrieval and fusion on a given shape for geometry completion.

Specifically, given an input incomplete shape, the proposed approach first predicts a coarse completion using an off-the-shelf method. The coarse completion does not necessarily capture the shape details but it provides guidance on locations of the missing patches. For each coarse voxel patch, we learn a shape distance function to retrieve top-k detailed shape patches in the input shape. The final stage of our approach learns a deformation for each retrieved patch and a blending function to integrate the retrieved patches into a continuous surface. The deformation prioritizes the compatibility scores between adjacent patches. The blending functions optimize the contribution of each patch and ensure surface smoothness.
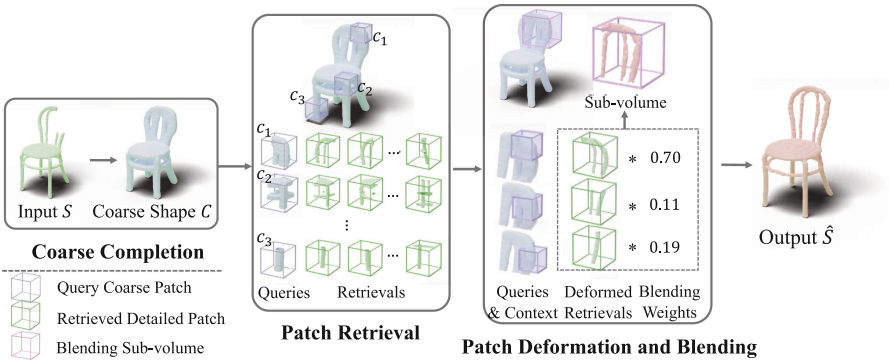
Fig. 2. Approach pipeline. Given an incomplete shape S, we first predict a coarse shape C with the rough structure and no details. For each patch on C, K detailed patch candidates are retrieved from the input shape. Then we predict deformations and blending weights for all retrieved candidates. Finally, the output shape $\hat{S}$ is computed by summing up the deformed patches and their blending weights.

Experimental results on the ShapeNet dataset [6] show that our approach outperforms existing shape completion techniques for reconstructing shape details both qualitatively and quantitatively.

In summary, our contributions are:

– We propose a non-parametric shape completion method based on patch retrieval and deformation.
– Our method preserves local shape details while enforcing global consistency.
– Our method achieves state-of-the-art shape completion results compared with various baselines.

2 Related Work

Shape Completion. Shape completion is a crucial and long-studied task in geometry processing. Non-data-driven works [27,28,40,54] address hole-filling in

a purely geometric manner. Without any high-level prior on the resulting shape, they target filling holes in a "smooth-as-possible" manner with membranes. To complete more complex shapes, several works [29,34,36,44,50,55] rely on data-driven methods to get the structure priors or part references. Similarly to our method, [34,44,50] retrieve some candidate models from a database, then perform a non-rigid surface alignment to deform the retrieved shape to fit the input. However, our approach operates at the patch level and can reconstruct shapes that are topologically different from those in the training data. With the development of deep learning, neural networks can be used to learn a shape prior from a large dataset and then complete shapes. Voxel-based methods [12,69,70] are good at capturing rough structures, but are limited to low resolution by the cubic scaling of voxel counts. Our framework is especially designed to circumvent these resolution limitations. Alternatively, point cloud completion [42,59,67,71–73,77,78] has become a popular venue as well. [25,66,73] use coarse-to-fine structures to densify the output point cloud and refine local regions. NSFA [79] and HRSC [19] used a two stage method to infer global structures and refine local geometries. SnowflakeNet [71] modeled the progressive generation hierarchically and arranged the points in locally structured patterns. As point clouds are sparse and unstructured, it is difficult to recover fine-grained shape details. 3D-EPN [12] and our method both use coarse-to-fine and patch-based pipelines. However, their method only retrieves shapes from the training set and directly copies the nearest patches based on low-level concatenation of distance fields. Our method retrieves patch-level details from the input and jointly learns deformation and blending, which enables our method to handle complex details as well as maintain global coherence.

Patch-Based Image In-Painting. In the 2D domain, many works utilize detailed patches to get high-resolution image inpainting results. Traditional methods [2,14,22,32] often synthesize textures for missing areas or expanding the current images. PatchMatch [2] proposed an image editing method by efficiently searching and replacing local patches. SceneComp [22] patched up holes in images by finding similar image regions in a large database. Recently, with the power of neural networks, more methods [37,47,49,60,75,76] use patch-guided generation to get finer details. [37,47,60] modeled images to scene graphs or semantic layouts and retrieve image patches for each graph/layout component. [49,75,76] add transformers [65], pixel flow and patch blending to get better generation results respectively. Our method leverages many insights from the 2D domain, however these cannot be directly transferred to 3D, for two reasons: i) the signals are inherently different, as 2D pixels are spatially-dense and continuous, while voxels are sparse and effectively binary; ii) the number of voxels in a domain scales cubically with resolution, as opposed to the quadratic scaling of pixels. This significantly limits the performance of various algorithms. The novel pipeline proposed herein is tailor-made to address these challenges.

3D Shape Detailization. Adding or preserving details on 3D shapes is an important yet challenging problem in 3D synthesis. Details can be added to a given surface via a reference 3D texture [23,56,81]. More relevant to use various

geometric representations to synthesize geometric details [5,8,9,16,35]. DLS [5] and LDIF [16] divide a shape to different local regions and reconstruct local implicit surfaces. D2IM-Net [35] disentangles shape structure and surface details. DECOR-GAN [9] trained a patch-GAN to transfer details from one shape to another. In our case, we focus on the task of partial-to-full reconstruction, and use detailization as a submodule during the process.

3D Generation by Retrieval. Instead of synthesizing shapes from scratch with a statistical model, it is often effective to simply retrieve the nearest shape from a database [15,31,57,58]. This produces high-quality results at the cost of generalization. Deformation-aware retrieval techniques [39,51,63,64] improve the representation power from a limited database. Our method also combines deformation with retrieval, but our retrieval is at the level of local patches from the input shape itself. RetrievalFuse [52] retrieves patches from a database for scene reconstruction. An attention-based mechanism is then used to regenerate the scene, guided by the patches. In contrast, we directly copy and deform retrieved patches to fit the output, preserving their original details and fidelity.

3   Overview

Our framework receives an incomplete or a partial shape S as input and completes it into a full *detailed* shape $\hat{S}$. Our main observation is that local shape details often repeat and are consistent across different regions of the shape, up to an approximately rigid deformation. Thus, our approach extracts local regions, which we call *patches*, from the given incomplete shape S, and uses them to complete and output a full complete shape. In order to analyze and synthesize topologically diverse data using convolutional architectures, we represent shapes and patches as voxel grids with occupancy values, at a resolution of s_{shape} cells.

The key challenges facing us are choosing patches from the partial input, and devising a method to deform and blend them into a seamless, complete detailed output. This naturally leads to a three-stage pipeline: (i) complete the partial input to get a coarse complete structure C to guide detail completion; (ii) for each completed coarse patch in C, retrieve candidate detailed patches from the input shape S; (iii) deform and blend the retrieved detailed patches to output the complete detailed shape $\hat{S}$ (see Fig. 2). Following is an overview of the process; we elaborate on each step in the following sections.

Coarse Completion. We generate a full coarse shape C from the partial input S using a simple 3D-CNN architecture. Our goal is to leverage advances in 3D shape completion, which can provide coarse approximations of the underlying ground truth, but does not accurately reconstruct local geometric details.

Patch Retrieval (Sect. 4). We train another neural network to retrieve k candidate detailed patches from S for each coarse patch in C. Namely, we learn geometric similarity, defined by a rigid-transformation-invariant distance d, between the coarse and detailed patches.

Deformation and Blending of Patches (Sect. 5). Given k candidate patches, we use a third neural network to predict rigid transformations and blending weights for each candidate patch, which together define a deformation and blending of patches for globally-consistent, plausible shape completion.

4   Patch Retrieval

The input to this stage is the partial detailed shape S, and a coarse and completed version of the shape, C. The goal of this step is to retrieve a set of patch candidates that can be deformed and stitched to get a fully detailed shape. A *patch* is a cube-shaped sub-region extracted from a shape, composed of s_{patch}^3 voxels. Our patch sampling process $\mathcal{P}$ takes a shape as input and outputs a collection of patches, where coarse patches $\mathcal{P}(C)$ serve as queries and detailed patches from the partial input $\mathcal{P}(S)$ as sources.
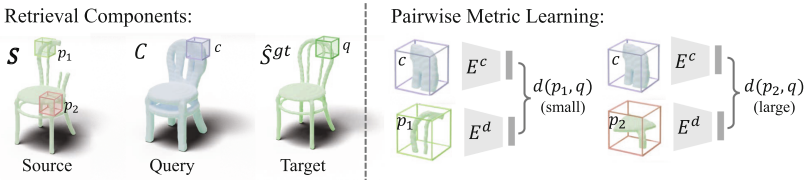
Fig. 3. Retrieval learning. We learn a feature mapping to predict geometric distances between the query coarse patches and the sampled detailed patches. We use the geometric distances between the GT detailed patches and the sampled patches as the supervision. Distances for patches that are close up to a rigid transformation are small. Otherwise, distances are large.

In order to decide whether a retrieved detailed patch could be an apt substitution for a true detailed patch, we propose a *geometric distance* metric invariant to rigid deformations (Sect. 4.1). This geometric distance will be used to supervise the neural network used during testing time, which learns similarities between coarse patches $\mathcal{P}(C)$ and their detailed counterparts $\mathcal{P}(S)$ (Sect. 4.2). Finally, we describe how to use this network at inference time to retrieve candidate patches for the full shape (Sect. 4.3).

4.1   Geometric Distance

We define a measure of geometric distance, d, between two *detailed* shape patches (p_1, p_2). This metric should be agnostic to their poses, since the pose can be fixed at the deformation stage, hence we define the distance as the minimum over all possible rigid transformations of the patch:

$$d(p_1, p_2) = \min_T \frac{\|T(p_1) - p_2\|_1}{\|T(p_1)\|_1 + \|p_2\|_1} \tag{1}$$

where T is a rigid motion composed with a possible reflection, i.e., $T = (R, \mathbf{t}, f)$, $R \in SO(3)$ is the rotation, $\mathbf{t} \in \mathbb{R}^3$ is the translation, $f \in \{0, 1\}$ denotes if a reflection is applied, and $|| \cdot ||_1$ denotes the L1 norm of positional vectors to patch centers. To practically compute this distance, we run ICP [3], initialized from two transformations (with and without reflection enabled) that align patch centers. While this geometric distance can be computed for detailed patches, at inference time we only have coarse patches. Therefore, we train a network to embed coarse patches into a latent space in which Euclidean distances match the geometric distances of the detailed patches they represent.

4.2   Metric Embedding

We train two neural networks to act as encoders, one for coarse patches and one for detailed patches, E^{c} and E^{d}, respectively. We aim to have the Euclidean distances between their generated codes reflect the distances between the true detailed patches observed during training. Given a coarse patch $c \in \mathcal{P}(C)$ with its true corresponding detailed patch $q \in \mathcal{P}(\hat{S}^{\mathrm{gt}})$, as well as a some other detailed patch $p \in \mathcal{P}(S)$, we define a metric embedding loss:

$$L_{\mathrm{r}} = \sum_{(c,p,q) \in \mathcal{T}} ||\|E^{\mathrm{c}}(c) - E^{\mathrm{d}}(p)\|_2 - d(p, q))\|_2. \tag{2}$$

where $d(p, q)$ is the geometric distance defined in Eq. (1). Our training triplets are composed of true matches and random patches: $\mathcal{T} = \mathcal{T}_{\mathrm{true}} \cup \mathcal{T}_{\mathrm{rnd}}$. Where in both sets c is a random coarse patch, q is the corresponding true detailed patch. We either set $p = q$ for $\mathcal{T}_{\mathrm{true}}$ or randomly sample $p \in \mathcal{P}(S)$ for $\mathcal{T}_{\mathrm{rnd}}$. See Fig. 3 for an illustration.

4.3   Retrieval on a Full Shape

We can now use trained encoder networks at inference time to retrieve detailed patches for each coarse patch. First, we encode all the detailed patches in $\mathcal{P}(S)$ via E^{d}. Similarly, for each non-empty coarse patch $c \in \mathcal{P}(C)$ with lowest corner at location l, we encode it with E^{c} and find the K-nearest-neighbor detailed codes. We store the list of retrieved patches for each location, denoted as $\mathcal{R}_l$.

We sample the coarse patches using a fixed-size (s_{patch}^3) sliding window with a stride γ_{patch}. Note that in the retrieval stage we do not assume that we know which parts of the detailed shape need to be completed. Since our feature learning step observed a lot of positive coarse/fine pairs with the detailed input, we found that the input is naturally reconstructed from the retrieved detailed patches.

5   Deformation and Blending of Patches

The input to this stage is the coarse shape C, partial input S, and the retrieval candidates. The output is the full detailed shape $\hat{S}$, produced by deforming

and blending the retrieved patches. As illustrated by Fig. 2 we first apply a rigid transformation to each retrieved patch and then blend these transformed patches into the final shape. Our guiding principle is the notion of partition-of-unity [41], which blends candidate patches with optimized transformations into a smooth completion. Unlike using fixed weighting functions, we propose to learn the blending weights. These weights serve the role of selecting candidate patches and stitching them smoothly.

Table 1. Shape completion results on the random-crop dataset on 8 ShapeNet categories. We show the L_2 Chamfer distance (CD) ($\times 10^3$) between the output shape and the ground truth 16384 points from PCN dataset [78] (lower is better). Our method reduces the CD drastically compared with the baselines.

	Average	Chair	Plane	Car	Table	Cabinet	Lamp	Boat	Couch
AtlasNet [17]	7.03	6.08	2.32	5.32	5.38	8.46	14.20	6.01	8.47
Conv-ONet [46]	6.42	2.91	2.29	8.60	7.94	12.6	5.82	4.03	7.21
TopNet [59]	6.30	5.94	2.18	4.85	5.63	5.13	15.32	5.60	5.73
3D-GAN [69]	6.00	6.02	1.77	3.46	5.08	7.29	12.23	7.20	4.92
PCN [78]	4.47	3.75	1.45	3.58	3.32	4.82	10.56	4.22	4.03
GRNet [73]	2.69	3.27	1.47	3.15	2.43	3.35	2.54	2.50	2.84
VRCNet [42]	2.63	2.96	1.30	3.25	2.35	2.98	2.86	2.23	3.13
SnowflakeNet [71]	2.06	2.45	**0.72**	2.55	2.15	2.76	2.17	1.33	2.35
PatchRD (Ours)	**1.22**	**1.08**	0.98	**1.01**	**1.32**	**1.45**	**1.23**	**0.99**	**1.67**

We observe that learning the blending weights requires some context (our method needs to be aware of at least a few neighboring patches), but does not require understanding the whole shape (coarse shape and retrieved patches already constrain the global structure). Thus, to maximize efficiency and generalizability, we opt to perform deformation and blending at the meso-scale of subvolumes $V \subset S$ with size s_{subv}.

Next, we provide more details on our blending operator (Sect. 5.1) and how to learn it from the data (Sect. 5.2).

5.1   The Deformation and Blending Operator

Given a subvolume V, we first identify $[r_m, m = 1...M]$ an ordered list of M best patches to be considered for blending. These patches are from the retrieved candidates $\mathcal{R}_l$ such that $l \in V$, and sorted according to two criteria: (i) retrieval index, (ii) x, y, z ordering. If more than M such patches exist, we simply take the first M. Each patch r_m is transformed with a rigid motion and possible reflection: T_m, and we have a blending weight for each patch at every point x in our volume: $\omega_m[x]$. The output at voxel x is the weighted sum of the deformed blending candidates:

$$V[x] = \frac{1}{\xi[x]} \sum_{m=1...M} \omega_m[x] \cdot T_m(r_m)[x] \qquad (3)$$

where $\omega_m[x]$ is the blending weight for patch m at voxel x, and $T_m(r_m)$ is the transformed patch (placed in the volume V, and padded with 0), and $\xi[x] = \sum_{m=1..M} \omega_m[x]$ is the normalization factor. At inference time, when we need to reconstruct the entire shape, we sample V over the entire domain $\hat{S}$ (with stride γ_{subv}), and average values in the region of overlap.

5.2   Learning Deformation and Blending

Directly optimizing deformation and blending is prone to being stuck in local optimum. To address this we develop a neural network to predict deformations and blending weights and train it with reconstruction and smoothness losses.

Prediction Network. We train a neural network g to predict deformation and blending weights. The network consists of three convolutional encoders, one for each voxel grid: the coarse shape (with a binary mask for the cropped subvolume V), the partial input, and the tensor of retrieved patches (M channels at resolution of V). We use fully-connected layers to mix the output of convolutional encoders into a bottleneck, which is than decoded into deformation T and blending ω parameters.

Reconstruction Loss. The first loss L_{rec} aims to recover the target shape $\hat{S}^{\mathrm{gt}}$:

$$L_{\mathrm{rec}} = \|V^{\mathrm{gt}} - V\|_2, \qquad (4)$$

where $V^{\mathrm{gt}} \subset \hat{S}^{\mathrm{gt}}$ and $V \subset \hat{S}$ are corresponding true and predicted subvolumes (we sample V randomly for training).

Blending Smoothness Loss. The second loss L_{sm} regularizes patch pairs. Specifically, if two patches have large blending weights for a voxel, then their transformations are forced to be compatible on that voxel:

$$L_{\mathrm{sm}} = \sum_{x \in \mathcal{V}} \sum_{m,n} \|\omega_m[x] \cdot \omega_n[x] \cdot (T_m(r_m)[x] - T_n(r_n)[x])\|$$

where x iterates over the volume and m,n over all retrieved patches. Note that r_m and r_n are only defined on a small region based on where the patch is placed, so this sum only matters in regions where transformed patches $T_m(r_m)$ and $T_n(r_n)$ map to nearby points x accordingly.

Final Loss. The final loss term is

$$L = L_{\mathrm{rec}} + \alpha L_{\mathrm{sm}}. \qquad (5)$$

6 Experiments

We primarily evaluate our method on the detail-preserving shape completion benchmark (Sect. 6.1), and demonstrate that our method outperforms state-of-the-art baselines (Sect. 6.2). We further demonstrate that our method can generalize beyond the benchmark setup, handling real scans, data with large missing areas, and novel categories of shapes (Sect. 6.3). Finally, we run an ablation study (Sect. 6.4) and evaluate sensitivity to the size of the missing area (Sect. 6.5).

6.1 Experimental Setup

Implementation Details. We use the following parameters for all experiments. The sizes of various voxel grids are: $s_{\text{shape}} = 128, s_{\text{patch}} = 18, s_{\text{subv}} = 40$ with strides $\gamma_{\text{patch}} = 4, \gamma_{\text{subv}} = 32$. We sample $|\mathcal{T}_{\text{rnd}}| = 800$ and $|\mathcal{T}_{\text{true}}| = 400$ triplets to train our patch similarity (Sect. 4.2). Our blending operator uses $M = 400$ best retrieved patches (Sect. 5.1). We set $\alpha = 10$ for Eq. 5. To improve performance we also define our blending weights ω_m at a coarser level than V.

Input   3D-GAN   Conv-ONet   VRCNet   Snowflake   PatchRD **(Ours)**   GT

Fig. 4. Qualitative shape completion results on the Random-Crop Dataset. Our results recover more geometric details and keep the shape smooth while other baselines often produce coarse, noisy, or discontinuous results.

Table 2. FID comparison on the chair class (note that we can only apply this metric to volumetric baselines). Our method produces more plausible shapes.

	Conv-ONet [46]	3D-GAN [69]	PatchRD(Ours)
FID	174.72	157.19	**11.89**

In particular, we use windows of size $s_{\text{blend}}^3 = 8^3$ to have constant weight, and compute the blending smoothness term at the boundary of these windows.

Dataset. We use shapes from ShapeNet [6], a public large-scale repository of 3D meshes to create the completion benchmark. We pick eight shape categories selected in prior work PCN [78]. For each category, we use the same subset of training and testing shapes with 80%/20% split as in DECOR-GAN work [9]. For voxel-based methods, we convert each mesh to a 128^3 voxel grid, and for point-based baselines, we use the existing point clouds with 16384 points per mesh [78]. We create synthetic incomplete shapes by cropping (deleting) a random cuboid with 10%–30% volume with respect to the full shape. This randomly cropped dataset is generated to simulate smaller-scale data corruption. We also show results on planar cutting and point scans in Sect. 6.3.

Metrics. To evaluate the quality of the completion, we use the L_2 Chamfer Distance (CD) with respect to the ground truth detailed shape. Since CD does not really evaluate the quality of finer details, we also use Frechet Inception Distance (FID), to evaluate plausibility. FID metric computes the distance of the layer activations from a pre-trained shape classifier. We use 3D VGG16 [53]) trained on ShapeNet and activations of the first fully connected layer.

Baseline Approaches. To the best of our knowledge, we are the first to do the 3D shape completion task on the random-cropped dataset. Considering the task similarity, we compare our method with the other shape completion and reconstruction baselines.

Our baselines span different shape representations: PCN [78], TopNet [59], GRNet [73], VRCNet [42], and SnowFlakeNet [71] are point-based scan completion baselines, 3D-GAN [69] is a voxel-based shape generation method, Conv-ONet [46] is an implicit surfaces-based shape reconstruction methods, and Atlas-Net [17] is an atlas-based shape reconstruction method. We show our method outperforms these baselines both quantitatively and qualitatively.
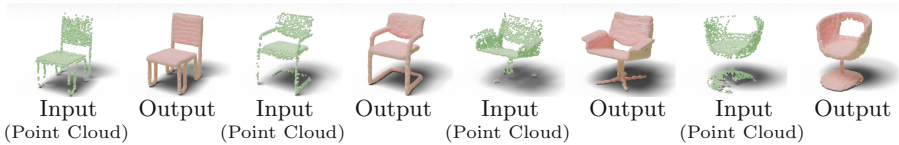
6.2   Shape Completion Results

Table 1 and Table 2 show quantitative comparisons between PatchRD and baselines, demonstrating that our method significantly outperforms all baselines. PatchRD achieved superior performance on all categories except airplanes, which shows that it generalizes well across different classes of shapes.

Specifically, the voxel-based baseline [69] produces coarse shapes where fine details are missing. Point-based baselines [42,71] often have noisy patterns

around on fine structures while our method has clean geometry details. The
implicit surface based method [46] could capture the details but the geometry
is not smooth and the topology is not preserved. Our method keeps the smooth
connection between geometry patches. More results can be found in the sup-
plemental materials. Figure 4 shows qualitative comparisons. We pick four rep-
resentative baselines for this visualization including point-based methods that
performed the best on the benchmark [42, 71] as well as voxel-based [69] and
implicit-based methods [46]. Our results show better shape quality by recover-
ing local details as well as preserving global shape smoothness.

6.3   Other Applications

Real-World Application: Scan Completion. We test our method on real-
world shape completion from scans. We use shapes from ScanNet [11], 3D indoor
scene dataset as input to our method. Objects in ScanNet often have some
missing parts, especially thin structures, due to occlusion and incompleteness
of scan viewpoints. We convert these point clouds to voxel grids and apply our
completion technique trained on ShapeNet, see results in Fig. 5a. Note how our

Input   Output   Input   Output   Input   Output   Input   Output
(Point Cloud)   (Point Cloud)   (Point Cloud)   (Point Cloud)

(a) Shape completion results on real scans for ScanNet objects. Our method completes
the missing areas and fills the uneven areas with detailed and smooth geometries.

Input   Output   Input   Output   Input   Output   Input   Output
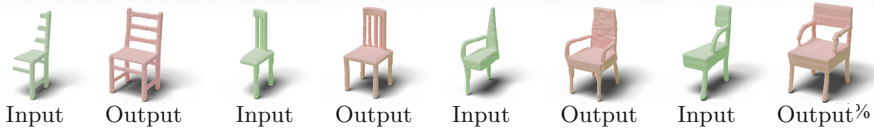
(b) Shape completion results on shapes with large missing areas. Our method recovers
geometric details even when given relatively small regions with reference patterns.

Lamp → Chair   Cabinet → Plane

Input Conv-ONet[46] Snow[71] PatchRD   Input Conv-ONet[46] Snow[71] PatchRD
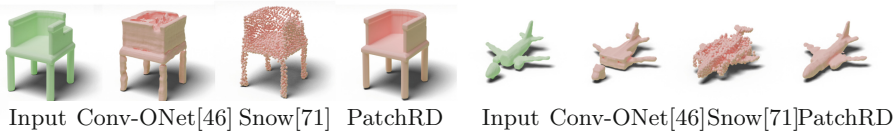
(c) Testing results on novel categories. We show results trained on lamp and cabinet
categories and inferred on lamp and plane categories, respectively. Our method has
better generalization ability.

Fig. 5. More applications on real scans, shapes with large missing areas and novel
categories.

method completes the undersampled areas, while preserving the details of the input and smoothly blending new details to the existing content.

Shapes with Large Missing Areas. We also demonstrate that our method can handle large missing areas (Fig. 5b). In this experiment we cut the shape with a random plane, where in some cases more than half of the shape might be missing. Our method recovers the shape structure and extends the local shape details to the whole shape when only given a small region of reference details.

Completion on Novel Categories. We further evaluate the ability of our method to generalize to novel categories. Note that only the prediction of the complete coarse shape relies on any global categorical priors. Unlike other generative techniques that decode entire shape, our method does not need to learn how to synthesize category-specific details and thus succeeds in detail-preservation as long as the coarse shape is somewhat reasonable. In Fig. 5c we demonstrate the output of different methods when tested on a novel category. Note how our method is most successful in conveying overall shape as well as matching the details of the input.

6.4 Ablation Study

We evaluate the significance of deformation learning, patch blending, and blending smoothness term via an ablation study (see Table 3).

Table 3. Ablation study. In the left figure, we visualize the effect of different components in our experiment. Patch alignment can't get good patch transformation. Results with no blending are subjective to bad retrievals. Results with no smoothing show discontinuity between neighboring patches. Results with all components contain geometric details as well as smoothness. In the right table, We show the reconstruction error CD and shape plausibility score FID on ShapeNet chair class. Results with all components get both better CD and FID.

Input   Patch Alignment   No Blending   No Smoothing   All   GT
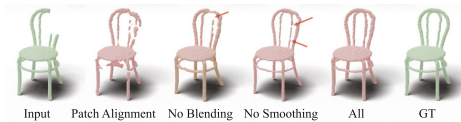
	CD	FID
Patch Alignment	4.90	43.35
No Blending	2.03	30.25
No Smoothing	1.86	27.42
All	**1.43**	**11.89**

No Deformation Learning. We simply use ICP to align the retrieved patch to the query. Table 3 (Patch Alignment) illustrates that this leads to zigzagging artifacts due to patch misalignments.

No Patch Blending. Instead of blending several retrieved patches, we simply place the best retrieved patch at the query location. Table 3 (No Blending) shows that this single patch might not be sufficient, leading to missing regions in the output.
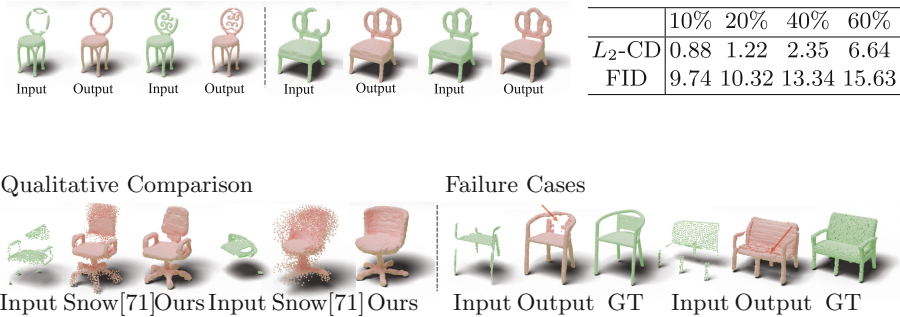
No Blending Smoothness. We set the blending smoothness to $L_{\mathrm{sm}} = 0$ to remove our penalty for misalignments at patch boundaries. Doing so leads to artifacts and discontinuities at patch boundaries (Table 3, No Smoothing).

The quantitative results in Table 3 show that our method with all components performs the best with respect to reconstruction and plausibility metrics.

6.5   Sensitivity to the Size of Missing Regions

The completion task is often ill-posed, and becomes especially ambiguous as the missing region increases. We evaluate the sensitivity of our method to the size of the region by trying to increase the crop size from 10% to 50% of the volume. Table 4 demonstrates that our method can produce plausible completions even under severe crops. However, in some cases it is impossible to reconstruct details that are completely removed. We report quantitative results in Table 4. While both reconstruction and plausibility error increases for larger crops, we observe that plausibility (FID) score does not deteriorate at the same rate.

Table 4. Sensitivity to the size of missing regions. The left figure shows results with different crop ratios. Input geometries and shape contours influence the output shapes. The right table shows the reconstruction error and the shape plausibility with the increase of crop ratios. As the ratio increases, the reconstruction error keeps growing although the output shapes remain fairly plausible.

	10%	20%	40%	60%
L_2-CD	0.88	1.22	2.35	6.64
FID	9.74	10.32	13.34	15.63

Input Output Input Output Input Output Input Output

Qualitative Comparison   Failure Cases

Input Snow[71] Ours Input Snow[71] Ours   Input Output GT   Input Output GT

Fig. 6. Qualitative results on the PCN Dataset [78]. On the left, we show our method is able to produce cleaner and more plausible results than the structure-based baseline. On the right, we show some failure cases where shape details are missing in the input shape.

7   Conclusions, Limitations and Future Work

Conclusions. This paper proposed a novel non-parametric shape completion method that preserves local geometric details and global shape smoothness. Our method recovers details by copying detailed patches from the incomplete shape, and achieves smoothness by a novel patch blending term. Our method obtained state-of-the-art completion results compared with various baselines with different

3D representations. It also achieved high-quality results in real-world 3D scans and shapes with large missing areas.

Limitations. Our method has two limitations: (1) It builds on the assumption that the shape details are present in the partial input shape, which might not hold if large regions are missing in the scan. For completeness, we still evaluate our method in this scenario using the PCN benchmark, which focuses on large-scale structure recovery from very sparse input. In Fig. 6 we show that our method succeeds when there are enough detail references in the input, and fails if the input is too sparse. We also provide quantitative evaluations in supplemental material. These results suggest that our method is better suited for completing denser inputs (e.g., multi-view scans).

In the future, we plan to address this issue by incorporating patches retrieved from other shapes. (2) Our method cannot guarantee to recover the global structure because the retrieval stage is performed at the local patch level. To address this issue, we need to enforce suitable structural priors and develop structure-aware representations. We leave both for future research.

Future Work. Recovering geometric details is a hard but important problem. Our method shows that reusing the detailed patches from an incomplete shape is a promising direction. In the direction of the patch-based shape completion, potential future work includes: (1) Applying patch retrieval and deformation on other 3D representations such as point cloud and implicit surfaces. This can handle the resolution limitation and computation burden caused by the volumetric representation. (2) Unifying parametric synthesis and patch-based non-parametric synthesis to augment geometric details that are not present in the partial input shape.

References

1. Anguelov, D., Srinivasan, P., Koller, D., Thrun, S., Rodgers, J., Davis, J.: SCAPE: shape completion and animation of people. ACM Trans. Graph. (TOG) **24**, 408–416 (2005). https://doi.org/10.1145/1073204.1073207
2. Barnes, C., Shechtman, E., Finkelstein, A., Goldman, D.B.: PatchMatch: a randomized correspondence algorithm for structural image editing. ACM Trans. Graph. (Proc. SIGGRAPH) **28**(3) (2009)
3. Besl, P.J., McKay, N.D.: A method for registration of 3-D shapes. IEEE Trans. Pattern Anal. Mach. Intell. **14**(2), 239–256 (1992). https://doi.org/10.1109/34.121791
4. Blanz, V., Vetter, T.: A morphable model for the synthesis of 3D faces. In: Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1999, pp. 187–194. ACM Press/Addison-Wesley Publishing Co. (1999). https://doi.org/10.1145/311535.311556
5. Chabra, R., et al.: Deep local shapes: learning local SDF priors for detailed 3D reconstruction. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12374, pp. 608–625. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58526-6_36

6. Chang, A.X., et al.: ShapeNet: an information-rich 3D model repository. Technical report. arXiv:1512.03012 [cs.GR], Stanford University—Princeton University—Toyota Technological Institute at Chicago (2015)

7. Chaudhuri, S., Koltun, V.: Data-driven suggestions for creativity support in 3D modeling. ACM Trans. Graph. **29** (2010). https://doi.org/10.1145/1866158.1866205

8. Chen, Z., et al.: Multiresolution deep implicit functions for 3D shape representation. In: ICCV (2021)

9. Chen, Z., Kim, V.G., Fisher, M., Aigerman, N., Zhang, H., Chaudhuri, S.: DECOR-GAN: 3D shape detailization by conditional refinement. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2021)

10. Curless, B., Levoy, M.: A volumetric method for building complex models from range images. In: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1996, pp. 303–312. Association for Computing Machinery, New York (1996). https://doi.org/10.1145/237170.237269

11. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: ScanNet: richly-annotated 3D reconstructions of indoor scenes. In: Proceedings of the Computer Vision and Pattern Recognition (CVPR). IEEE (2017)

12. Dai, A., Qi, C.R., Nießner, M.: Shape completion using 3D-encoder-predictor CNNs and shape synthesis. In: Proceedings of the Computer Vision and Pattern Recognition (CVPR). IEEE (2017)

13. Davis, J., Marschner, S., Garr, M., Levoy, M.: Filling holes in complex surfaces using volumetric diffusion. In: 3DPVT, pp. 428–441 (2002). https://doi.org/10.1109/TDPVT.2002.1024098

14. Efros, A.A., Leung, T.K.: Texture synthesis by non-parametric sampling. In: IEEE International Conference on Computer Vision (ICCV) (1999)

15. Eitz, M., Richter, R., Boubekeur, T., Hildebrand, K., Alexa, M.: Sketch-based shape retrieval. ACM Trans. Graph. **31**(4) (2012). https://doi.org/10.1145/2185520.2185527

16. Genova, K., Cole, F., Sud, A., Sarna, A., Funkhouser, T.: Local deep implicit functions for 3D shape. In: CVPR (2019)

17. Groueix, T., Fisher, M., Kim, V.G., Russell, B., Aubry, M.: AtlasNet: a Papier-Mâché approach to learning 3D surface generation. In: Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)

18. Guo, X., Xiao, J., Wang, Y.: A survey on algorithms of hole filling in 3D surface reconstruction. Vis. Comput. **34**(1), 93–103 (2016). https://doi.org/10.1007/s00371-016-1316-y

19. Han, X., Li, Z., Huang, H., Kalogerakis, E., Yu, Y.: High-resolution shape completion using deep neural networks for global structure and local geometry inference. In: IEEE International Conference on Computer Vision (ICCV) (2017)

20. Hanocka, R., Hertz, A., Fish, N., Giryes, R., Fleishman, S., Cohen-Or, D.: MeshCNN: a network with an edge. ACM Trans. Graph. **38**(4) (2019). https://doi.org/10.1145/3306346.3322959

21. Hanocka, R., Metzer, G., Giryes, R., Cohen-Or, D.: Point2Mesh: a self-prior for deformable meshes. ACM Trans. Graph. **39**(4) (2020). https://doi.org/10.1145/3386569.3392415

22. Hays, J., Efros, A.A.: Scene completion using millions of photographs. ACM Trans. Graph. (SIGGRAPH 2007) **26**(3) (2007)

23. Hertz, A., Hanocka, R., Giryes, R., Cohen-Or, D.: Deep geometric texture synthesis. ACM Trans. Graph. **39**(4) (2020). https://doi.org/10.1145/3386569.3392471

24. Hu, P., Wang, C., Li, B., Liu, M.: Filling holes in triangular meshes in engineering. JSW **7**, 141–148 (2012). https://doi.org/10.4304/jsw.7.1.141-148

25. Huang, Z., Yu, Y., Xu, J., Ni, F., Le, X.: PF-Net: point fractal network for 3D point cloud completion. In: CVPR (2020)

26. Kanazawa, A., Black, M.J., Jacobs, D.W., Malik, J.: End-to-end recovery of human shape and pose. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, 18–22 June 2018, pp. 7122–7131. Computer Vision Foundation/IEEE Computer Society (2018). https://doi.org/10.1109/CVPR.2018.00744. http://openaccess.thecvf.com/content_cvpr_2018/html/Kanazawa_End-to-End_Recovery_of_CVPR_2018_paper.html

27. Kazhdan, M., Bolitho, M., Hoppe, H.: Poisson surface reconstruction. In: Proceedings of the Fourth Eurographics Symposium on Geometry Processing (2005)

28. Kazhdan, M., Hoppe, H.: Screened poisson surface reconstruction. ACM Trans. Graph. (TOG) **32**, 1–13 (2013)

29. Kim, Y.M., Mitra, N.J., Yan, D.M., Guibas, L.: Acquiring 3D indoor environments with variability and repetition. ACM Trans. Graph. (TOG) **31**, 1–11 (2012)

30. Kolotouros, N., Pavlakos, G., Black, M.J., Daniilidis, K.: Learning to reconstruct 3D human pose and shape via model-fitting in the loop. In: 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), 27 October–2 November 2019, pp. 2252–2261. IEEE (2019). https://doi.org/10.1109/ICCV.2019.00234

31. Kuo, W., Angelova, A., Lin, T.Y., Dai, A.: Patch2CAD: patchwise embedding learning for in-the-wild shape retrieval from a single image. In: ICCV (2021)

32. Kwatra, V., Schodl, A., Essa, I., Turk, G., Bobick, A.: Graphcut textures: image and video synthesis using graph cuts. ACM Trans. Graph. SIGGRAPH 2003 **22**(3), 277–286 (2003)

33. Levoy, M., et al.: The digital Michelangelo project: 3D scanning of large statues. In: Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 2000, pp. 131–144. ACM Press/Addison-Wesley Publishing Co. (2000). https://doi.org/10.1145/344779.344849

34. Li, D., Shao, T., Wu, H., Zhou, K.: Shape completion from a single RGBD image. IEEE Trans. Vis. Comput. Graph. **23**, 1809–1822 (2016)

35. Li, M., Zhang, H.: D^{2}IM-Net: learning detail disentangled implicit fields from single images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 10246–10255 (2021)

36. Li, Y., Dai, A., Guibas, L., Nießner, M.: Database-assisted object retrieval for real-time 3D reconstruction. In: Computer Graphics Forum (2015)

37. Li, Y., Ma, T., Bai, Y., Duan, N., Wei, S., Wang, X.: PasteGAN: a semi-parametric method to generate image from scene graph. In: NeurIPS (2019)

38. Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., Black, M.J.: SMPL: a skinned multi-person linear model. ACM Trans. Graph. **34**(6) (2015). https://doi.org/10.1145/2816795.2818013

39. Nan, L., Xie, K., Sharf, A.: A search-classify approach for cluttered indoor scene understanding. ACM Trans. Graph. **31**, 1–10 (2012)

40. Nealen, A., Igarashi, T., Sorkine, O., Alexa, M.: Laplacian mesh optimization. In: Proceedings of the 4th International Conference on Computer Graphics and Interactive Techniques (2006)

41. Ohtake, Y., Belyaev, A., Alexa, M., Turk, G., Seidel, H.P.: Multi-level partition of unity implicits. ACM Trans. Graph. **22**(3), 463–470 (2003). https://doi.org/10.1145/882262.882293

42. Pan, L., et al.: Variational relational point completion network. arXiv preprint arXiv:2104.10154 (2021)
43. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: DeepSDF: learning continuous signed distance functions for shape representation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
44. Pauly, M., Mitra, N.J., Giesen, J., Gross, M.H., Guibas., L.J.: Example-based 3D scan completion. In: Symposium on Geometry Processing (2005)
45. Pavlakos, G., et al.: Expressive body capture: 3D hands, face, and body from a single image. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, 16–20 June 2019, pp. 10975–10985. Computer Vision Foundation/IEEE (2019). https://doi.org/10.1109/CVPR.2019.01123. http://openaccess.thecvf.com/content_CVPR_2019/html/Pavlakos_Expressive_Body_Capture_3D_Hands_Face_and_Body_From_a_CVPR_2019_paper.html
46. Peng, S., Niemeyer, M., Mescheder, L., Pollefeys, M., Geiger, A.: Convolutional occupancy networks. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12348, pp. 523–540. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58580-8_31
47. Qi, X., Chen, Q., Jia, J., Koltun, V.: Semi-parametric image synthesis. In: CVPR (2018)
48. Ranjan, A., Bolkart, T., Sanyal, S., Black, M.J.: Generating 3D faces using convolutional mesh autoencoders. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11207, pp. 725–741. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01219-9_43
49. Ren, Y., Yu, X., Zhang, R., Li, T.H., Liu, S., Li, G.: StructureFlow: image inpainting via structure-aware appearance flow. In: IEEE International Conference on Computer Vision (ICCV) (2019)
50. Rock, J., Gupta, T., Thorsen, J., Gwak, J., Shin, D., Hoiem, D.: Completing 3D object shape from one depth image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
51. Schulz, A., Shamir, A., Baran, I., Levin, D.I.W., Sitthi-Amorn, P., Matusik, W.: Retrieval on parametric shape collections. ACM Trans. Graph. **36**, 1–14 (2017)
52. Siddiqui, Y., Thies, J., Ma, F., Shan, Q., Nießner, M., Dai, A.: RetrievalFuse: neural 3D scene reconstruction with a database. In: ICCV (2021)
53. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR (2015)
54. Sorkine, O., Cohen-Or, D.: Least-squares meshes. In: Shape Modeling Applications (2004)
55. Sung, M., Kim, V.G., Angst, R., Guibas, L.: Data-driven structural priors for shape completion. ACM Trans. Graph. (TOG) **34**, 1–11 (2015)
56. Takayama, K., Schmidt, R., Singh, K., Igarashi, T., Boubekeur, T., Sorkine-Hornung, O.: GeoBrush: interactive mesh geometry cloning. Comput. Graph. Forum (Proc. EUROGRAPHICS 2011) **30**(2), 613–622 (2011)
57. Tangelder, J., Veltkamp, R.: A survey of content based 3D shape retrieval methods. In: Proceedings Shape Modeling Applications, pp. 145–156 (2004). https://doi.org/10.1109/SMI.2004.1314502
58. Tatarchenko, M., Richter, S., Ranftl, R., Li, Z., Koltun, V., Brox, T.: What do single-view 3D reconstruction networks learn? In: CVPR (2019)
59. Tchapmi, L.P., Kosaraju, V., Rezatofighi, S.H., Reid, I., Savarese, S.: TopNet: structural point cloud decoder. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)

60. Tseng, H.-Y., Lee, H.-Y., Jiang, L., Yang, M.-H., Yang, W.: RetrieveGAN: image synthesis via differentiable patch retrieval. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12353, pp. 242–257. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58598-3_15

61. Turk, G., Levoy, M.: Zippered polygon meshes from range images. In: Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1994, pp. 311–318. Association for Computing Machinery, New York (1994). https://doi.org/10.1145/192161.192241

62. Ulyanov, D., Vedaldi, A., Lempitsky, V.S.: Deep image prior. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, 18–22 June 2018, pp. 9446–9454. Computer Vision Foundation/IEEE Computer Society (2018). https://doi.org/10.1109/CVPR.2018.00984. http://openaccess.thecvf.com/content_cvpr_2018/html/Ulyanov_Deep_Image_Prior_CVPR_2018_paper.html

63. Uy, M.A., Huang, J., Sung, M., Birdal, T., Guibas, L.: Deformation-aware 3D model embedding and retrieval. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12352, pp. 397–413. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58571-6_24

64. Uy, M.A., Kim, V.G., Sung, M., Aigerman, N., Chaudhuri, S., Guibas, L.: Joint learning of 3D shape retrieval and deformation. In: CVPR (2021)

65. Vaswani, A., et al.: Attention is all you need. In: Guyon, I., et al. (eds.) Advances in Neural Information Processing Systems, vol. 30. Curran Associates, Inc. (2017). https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

66. Wang, X., Ang, M.H., Jr., Lee, G.H.: Cascaded refinement network for point cloud completion. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)

67. Wang, X., Ang, M.H., Jr., Lee, G.H.: Voxel-based network for shape completion by leveraging edge generation. In: ICCV (2021)

68. Wheeler, M., Sato, Y., Ikeuchi, K.: Consensus surfaces for modeling 3D objects from multiple range images. In: ICCV, pp. 917–924 (1998). https://doi.org/10.1109/ICCV.1998.710826

69. Wu, J., Zhang, C., Xue, T., Freeman, W.T., Tenenbaum, J.B.: Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling. In: Advances in Neural Information Processing Systems, pp. 82–90 (2016)

70. Wu, Z., et al.: 3D ShapeNets: a deep representation for volumetric shape modeling. In: Proceedings of 28th IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015)

71. Xiang, P., et al.: SnowflakeNet: point cloud completion by snowflake point deconvolution with skip-transformer. In: ICCV (2021)

72. Xie, C., Wang, C., Zhang, B., Yang, H., Chen, D., Wen, F.: Style-based point generator with adversarial rendering for point cloud completion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4619–4628 (2021)

73. Xie, H., Yao, H., Zhou, S., Mao, J., Zhang, S., Sun, W.: GRNet: gridding residual network for dense point cloud completion. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12354, pp. 365–381. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58545-7_21

74. Xiong, X., De la Torre, F.: Supervised descent method and its applications to face alignment. In: CVPR, pp. 532–539 (2013). https://doi.org/10.1109/CVPR.2013.75

75. Xu, R., Guo, M., Wang, J., Li, X., Zhou, B., Loy, C.C.: Texture memory-augmented deep patch-based image inpainting. IEEE Trans. Image Process. (TIP) **30**, 9112–9124 (2021)
76. Yang, F., Yang, H., Fu, J., Lu, H., Guo, B.: Learning texture transformer network for image super-resolution. In: CVPR (2020)
77. Yu, X., Rao, Y., Wang, Z., Liu, Z., Lu, J., Zhou, J.: PoinTr: diverse point cloud completion with geometry-aware transformers. In: ICCV (2021)
78. Yuan, W., Khot, T., Held, D., Mertz, C., Hebert, M.: PCN: point completion network. In: 2018 International Conference on 3D Vision (3DV) (2018)
79. Zhang, W., Yan, Q., Xiao, C.: Detail preserved point cloud completion via separated feature aggregation. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12370, pp. 512–528. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58595-2_31
80. Zhao, W., Chellappa, R., Phillips, P.J., Rosenfeld, A.: Face recognition: a literature survey. ACM Comput. Surv. **35**(4), 399–458 (2003). https://doi.org/10.1145/954339.954342
81. Zhou, K., et al.: Mesh quilting for geometric texture synthesis. In: ACM SIGGRAPH 2006 Papers, SIGGRAPH 2006, pp. 690–697 (2006)