D^{2}IM-Net: Learning Detail Disentangled Implicit Fields from Single Images

Manyi Li Hao Zhang
Simon Fraser University

Abstract

We present the first single-view 3D reconstruction network aimed at recovering geometric details from an input image which encompass both topological shape structures and surface features. Our key idea is to train the network to learn a detail disentangled *reconstruction consisting of two functions, one implicit field representing the coarse 3D shape and the other capturing the details. Given an input image, our network, coined D^{2}IM-Net, encodes it into global and local features which are respectively fed into two decoders. The base decoder uses the global features to reconstruct a coarse implicit field, while the detail decoder reconstructs, from the local features, two displacement maps, defined over the front and back sides of the captured object. The final 3D reconstruction is a fusion between the base shape and the displacement maps, with three losses enforcing the recovery of coarse shape, overall structure, and surface details via a novel* Laplacian *term.*

1. Introduction

Reconstructing 3D shapes from single-view RGB images is the prototypical ill-posed problem in computer vision. Recently, rapid advances in deep learning have propelled the development of data-driven single-view 3D reconstruction methods. In particular, the emergence of *neural implicit models* [5, 27, 22] for 3D shape representation learning has led to much improved reconstruction quality compared to methods designed for voxel grids, meshes, and point clouds. However, while technically the implicit fields could be sampled to an arbitrarily high spatial resolution, state-of-the-art reconstruction methods still are unable to adequately recover fine-level *geometric details*.

Implicit reconstruction networks such as IM-Net [5] and Occupancy Network [22] learn to predict an implicit function, given a feature encoding of the input image, by minimizing a reconstruction loss. These networks generalize well to new images, but only in terms of the coarse shapes; they are not designed to recover geometric details which are often of small scale and do not incur a sufficient penalty on the loss terms. In a more recent work, DISN, Xu et al. [44] account for both global *and local* image features to predict a combined signed distance field (SDF) so as to minimize a *single* reconstruction loss like prior works. Their network can better resolve structural details, such as the slats in the back of a chair, that are well captured by local image features. However, the rest of the details, in particular *surface details*, which are just as important for visual perception (e.g., of depth and material), are still not well recovered.

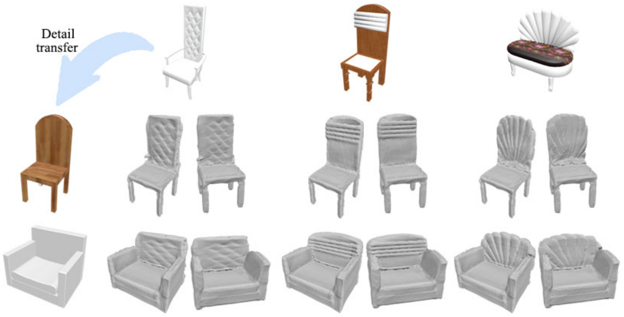


Figure 1. Our network learns to reconstruct a *detail disentangled* 3D representation from single-view images. The disentangled details enable detail transfer and 3D reconstruction (shown in two views) with the transferred details from image to another.

In this paper, we wish to develop an implicit single-view 3D reconstruction network which can recover both topological structures and surface details from an input image. Our key idea is that to best reconstruct the details, we ought to train the network to learn a *detail disentangled* reconstruction consisting of two functions, one representing the coarse 3D shape and one capturing the details. However, the main ensuing challenge is that geometric details are so varied that there is no general and reliable way to define what the details are or what a coarse shape should be. The network must learn the disentangled representations without direct supervision using ground-truth training data.

Figure 2 illustrates the pipeline of our detail disentangled implicit reconstruction network, coined D^{2}IM-Net. Given a single RGB input image, the network encodes it into global and local features which are respectively fed into two decoders. The *base decoder* uses the global features to reconstruct a coarse (i.e., base) implicit field, while the *detail decoder* reconstructs, from the local features, a pair of 2D

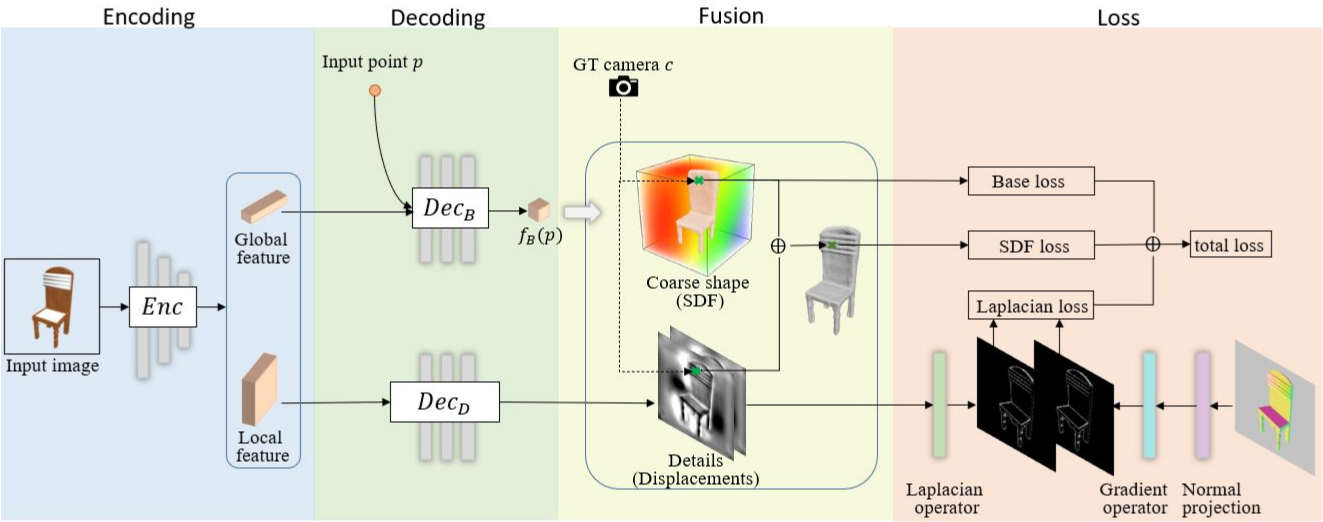
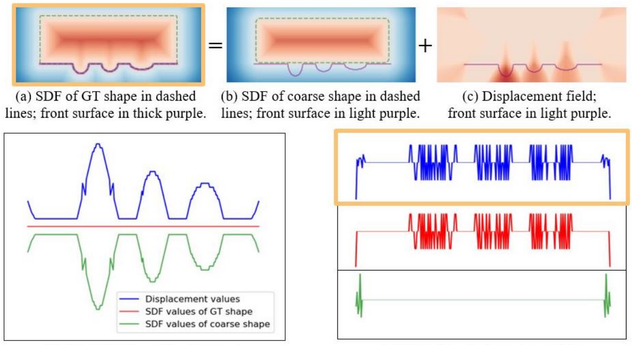


Figure 2. The pipeline of our single-view 3D reconstruction network D^{2}IM-Net consists of three stages. An encoder extracts global and local features from the input image. This is followed by two decoder branches which respectively predict a base or *coarse* shape from global features and two displacement maps (back and front) from local features. The final 3D reconstruction is a fusion between the base shape and the displacement maps, with three losses enforcing recovery of coarse shape, overall structure, and surface details (Laplacian).



(a) SDF of GT shape in dashed lines; front surface in thick purple.

(b) SDF of coarse shape in dashed lines; front surface in light purple.

(c) Displacement field; front surface in light purple.

(d) Plot of SDF and displacement field values along the front surface.

(e) Plot of Laplacian values of the fields shown on the left.

Figure 3. Illustration of a ground-truth (GT) shape+SDF (a) and a disentanglement into a base shape+SDF (b) and a displacement field (c). Bottom row plots SDF, displacement field, and Laplacian values along the *front surface* (purple lines) of the GT shape. We see close resemblance between the Laplacian of the displacement field values and that of the GT SDF: blue vs. red curves in (e). Note that at training, only the GT SDF is known (indicated by orange borders in the figure); all other fields are to be *learned*.

displacement maps, defined over the *front* and *back* sides of the captured object that are visible to the camera.

In the absence of any ground-truth displacement maps for training, or coarse shapes for that matter, we must rely on the original 3D shapes (e.g., from ShapeNet) or their associated SDFs to define the network losses. We first observe that the *Laplacian* of the SDF of a shape near the shape surface is sensitive to local geometry variations[1], i.e., the surface details. Furthermore, this Laplacian function resembles the Laplacian of the front displacement map if the front side of the coarse shape is mostly flat; see Figure 3. Based on these observations, we define a corresponding *Laplacian loss* to optimize the front displacement map.

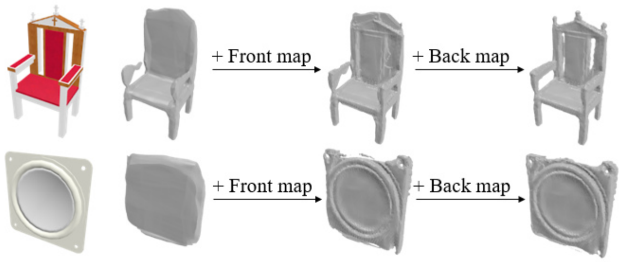


Figure 4. A visualization of 3D shapes reconstructed by the two decoders of D^{2}IM-Net demonstrates detail disentanglement: our network learns to recover surface details via the front displacement map and other details from the back map. The network was trained on ShapeNet across 13 shape categories.

In addition, we define a base loss and an SDF loss, both with respect to the ground-truth SDF, where the SDF loss is computed against a *fusion* between the predicted coarse SDF and the predicted displacement maps, both the front *and the back*. As the back displacement map is not factored into the Laplacian loss, it does not capture surface details. However, with local image features as input, the SDF loss does enforce the back map to help reconstruct the overall shape structure, including topological details. Figure 4 visualizes the disentangled functions our network reconstructs on two examples, where the predicted displacement maps evidently represent shape details, encompassing

[1] The Laplacian of a signed distance function at a point x is proportional to the mean curvature of the isosurface passing through x [8].

both topological structures and surface features, while the base decoder reconstructs the coarse shape.

We train our network on ShapeNet Core [3] across all 13 shape categories and test the network on single-view reconstruction for a variety of 3D objects, including those captured in "images in the wild". We conduct various ablation studies and present both qualitative and quantitative comparisons between D^{2}IM-Net and representative single-view 3D reconstruction methods including IM-Net [5] and DISN [44]. While the focus of our work is on reconstructing shape details, evaluations are conducted on images containing objects with varying degrees of geometric details and using different error metrics applicable to overall shapes and edge revelation. Finally, we develop and demonstrate a novel application of D^{2}IM-Net, where the ability to learn detail functions from images enables detail transfer from an image onto a reconstructed 3D shape; see Figure 1.

2. Related work

Most learning-based methods for 3D reconstruction aim to generalize to novel data [35, 43, 41, 29, 5, 22, 44, 9, 38, 6, 42, 10], while some recent networks are designed to "overfit" to specific inputs [1, 47, 33, 24, 18, 40]. In the latter case, a network is specifically trained to optimize the reconstruction for a given input, typically multi-view images [47, 33, 24, 18] or a point cloud [1]. As expected, such a specialization tends to produce much higher reconstruction quality compared to methods from the first category. However, with a new input, the network needs to be re-trained. Our work belongs to the first category and in this section, we mainly discuss related works in this category for *single-view* 3D reconstruction, or SVR, for short.

Neural implicit models for SVR. Deep neural models for SVR has gained significant improvements with various 3D shape representations, including voxels [6, 35, 41, 29, 42], meshes [38, 9], and structural primitives [26, 48]. Recently, implicit representations [5, 22, 27, 44, 43, 23, 17] have emerged as a desirable alternative due to the advantages they offer at representing continuous surfaces with higher visual quality and flexible topology.

Supervised by the ground-truth (GT) occupancy or SDF, earlier implicit reconstruction methods such as IM-NET [5], OccupancyNetwork [22], and DeepLevelSets [23] predict the scalar value at each 3D point to approximate the GT. Latent features encoded from the input images are fed into an MLP network together with 3D point coordinates to predict their occupancy or signed distances. Littwin and Wolf [17] take the encoded feature vectors as the network weights of the MLP to attain a more accurate reconstruction. Instead of predicting the implicit fields as a whole, PQ-NET [43] separately predicts the SDFs for each structural part of the captured object and then combines them together.

Unsupervised SVR. Along the lines of SVR without 3D supervision, differentiable renderers [13, 37] have been developed to back-propagate the loss computed from the input images. Liu et al. [20] propose a ray-based field probing technique to render the implicit surfaces to 2D silhouettes, with the geometric details erased from the silhouettes. Niemeyer et al. [25] account for both geometry and texture during rendering and make use of rich 2D supervision including RGB, depth, and normal images.

SVR with local image features. What is common about *all* the SVR methods above is that they are trained to reconstruct from *global* image features. As a result, they can successfully reconstruct coarse 3D shapes, but miss most shape details. A recent work by Tatarchenko et al. [36] reveals that such reconstructions could be easily outperformed by simple retrieval baselines, which may suggest that the main role played by the global images features is recognition rather than reconstruction. This naturally leads to the incorporation of local image features for learning 3D shapes [12, 28]. Representative networks along these lines include PIFu [30] and its follow-ups [31, 11], which were designed for detailed human shape reconstruction, but do not perform well on man-made models e.g., from ShapeNet, due to the greater geometric and structural variations.

Most closely related to our work is DISN [44] which accounts for both global and local image features for SVR. Specifically, it predicts the camera parameters to query the local image feature for each point. Global and local features are processed separately with the point coordinates to obtain two predictions, which are combined and optimized against a *single* SDF reconstruction loss. In addition, this loss is weighted to place more emphasis on errors associated with small SDF values. Qualitatively, the resulting reconstruction significantly improves the recovery of shape structures, in particular, topological details, but still unable to reconstruct surface details. In a more recent work, LadyBird, Xu et al. [45] employ farthest point sampling and feature fusion based on reflective symmetries to deal with self-occlusion. However, geometric details are not taken into account.

Compared to DISN [44], our network is specifically designed to learn a detail disentangled implicit shape representation, as contrasted in Figure 4. The key technical difference is that our network defines a dedicated loss for each reconstructed function (the based SDF and two displacement maps) and then sums up the losses, leading to disentanglement, while in DISN, there is only one loss. Specific to the recovery of surface details, we introduce a novel Laplacian loss to learn from GT normal maps.

Laplacian-space processing. The Laplacian operator for image or shape processing captures local variations. There have been neural networks which employ Laplacian pyramids to capture multi-scale image structures for coarse-to-fine image generation [7] and super-resolution [15, 34]. Also, Li et al. [16] develop a Laplacian loss for neural style transfer to preserve detailed image structures. However, it is non-trivial to extend Laplacian losses to the SVR framework, where the predicted shape representation must enable the Laplacian computation, while providing alignment to the GT surface. Applying Laplacian losses to surface meshes, as in Pixel2Mesh [38], is more straightforward, e.g., by means of minimizing the error between the predicted Laplacian coordinates before and after mesh deformation. More recently, in ParseNet, Sharma et al. [32] apply the Laplacian loss on parametric surfaces, aligning the GT and the predicted surfaces via Hungarian matching. For implicit methods, existing works such as SoftRas [19] resort to Laplacian regularization to obtain smooth surfaces, rather than detail recovery. In our work, we define disentangled detail functions as displacement maps, which are aligned with the input images, making it possible to define a proper Laplacian loss for SVR with surface details.

3. Method

Given a single RGB image of a 3D object, our goal is to reconstruct that object with high-quality shape details, in particular, geometry variations over its *surfaces*. The input to our reconstruction network consists of the image as well as a 3D point; the network outputs the *signed distance* from the input point to the target 3D object. Network training is supervised, taking multi-view projections from 3D objects in a shape repository to form the ground-truth data pairs.

Our network learns a *disentangled* signed distance field (SDF) reconstruction corresponding to the coarse shape and the shape details, employing a novel *Laplacian loss* to recover surface details. As shown in Figure 2, our network starts with an encoder using a CNN architecture to extract image features and two decoders to predict the coarse shape and details separately. The coarse shape and details, both in the form of scalar fields, are then fused together to obtain the SDF of the reconstructed 3D object. Finally, we apply Marching Cube [21] to extract the zero level set as the final reconstructed 3D output mesh model.

The main challenges include how to disentangle (Section 3.1) and how to define the Laplacian loss between network predictions and the ground truth (Section 3.3).

3.1. Detail disentanglement formulation

The Laplacian of the SDF of a shape near the shape's surface can help detect rapid local geometry variations [8], i.e., surface details. This motivates the use of Laplacians to help formulate our detail disentanglement under the implicit function setup. Specifically, we disentangle the ground-truth SDF F_{SDF} (i.e., the SDF of the ground-truth shape S) as the sum of a base implicit field, for a coarse shape, and the residual field which models displacements, as shown along the top of Figure 3 and expressed as follows:

$$\begin{aligned} &F_{SDF}(p) = f_B(p) + f_D(p), \\ &f_B : \mathbb{R}^3 \rightarrow \mathbb{R}, f_D : \mathbb{R}^3 \rightarrow \mathbb{R}, \end{aligned} \tag{1}$$

where f_B and f_D denote the base and displacement fields, respectively, which are learned. We follow the convention that capitalization, e.g., F, refers to ground-truth functions, while learned functions are given in lower-case.

We assume that the coarse shape is *smooth* and lies close to the surface S. The smoothness herein implies that the (residual) displacement field contains information about surface details. Such information is connected to F_{SDF} through the Laplacian. Furthermore, near S, the Laplacian of the displacement field f_D would closely approximate the Laplacian of F_{SDF}, if the detail displacements form a height field over a mostly flat surface (on the coarse shape). The latter implies that $\triangle f_B \approx 0$, hence, due to linearity of the Laplacian operator, we have

$$\triangle f_D(p) = \triangle F_{SDF}(p), |dist(p, S)| < \delta. \tag{2}$$

With $|dist(p, S)| < \delta$, only the Laplacian of points near S within a threshold δ need to be sampled during training.

However, for single-view 3D reconstruction, it is difficult to infer occluded geometry in 3D space. Inspired by recent works [46, 31] which treat the front and back surfaces separately, our network predicts *a pair of 2D displacement maps* for the visible front surface and the occluded back surface respectively, instead of a 3D displacement field. The front displacement map recovers details on the visible front surface, by optimizing the Laplacian near that surface against the ground-truth. The back displacement map approximates the residual between the SDF and base distance field to compensate for other details such as topological structures. Putting things together, we have

$$\begin{aligned} &F_{SDF}(p) = \begin{cases} f_B(p) + f_{DF}(u(p)), p \in P_F, \\ f_B(p) + f_{DB}(u(p)), otherwise, \end{cases} \\ &\triangle\, f_{DF}(u(p)) = \triangle F_{SDF}(p), p \in P_F, \\ &f_B : \mathbb{R}^3 \rightarrow \mathbb{R}, f_{DF} : \mathbb{R}^2 \rightarrow \mathbb{R}, f_{DB} : \mathbb{R}^2 \rightarrow \mathbb{R}, \end{aligned} \tag{3}$$

where f_{DF} and f_{DB} are the displacement maps for the front and back surfaces. $u(p)$ is the operation to project the 3D point p to the pixel position on the image. The point set P_F contains the points near the front surface.

The advantages of using 2D displacement maps instead of 3D fields are two fold. First, it enables us to learn the small-scale details with contemporary CNN networks. Second, it aligns the details with the input images to compute the Laplacian loss, which we discuss in Section 3.3.

3.2. Network pipeline: encoder, decoder, fusion

Figure 2 shows the pipeline of our network D^{2}IM-Net. The encoding uses a CNN to extract the global feature vector and local feature map from the input image. The base decoder Dec_B, an MLP, takes the global feature vector with a 3D point coordinate as input, and outputs the base value of this point, i.e., the signed distance from this point to the coarse shape. The detail decoder Dec_D contains the residual convolutional layers with the local feature map as input, and outputs a front displacement map encoding surface details on the visible front surface, and a back map to compensate for the topology details on the back surfaces. The back displacement map is necessary since a pixel outside the object mask should affect all the points along the ray.

The third stage is to fuse the base distance field with two displacement maps. Similar to DISN, we train a separate network to predict the camera parameters to query the displacement values per point on the displacement maps. As in equation (3), the base distance of a point is summed up with its corresponding value queried from the front displacement map, if this point is closer to the visible front surface. Otherwise, we sum the base distance and the corresponding value from the back displacement map. In the implementation, we simply estimate the gradient of the SDF at each point with central difference approximation. If the gradient direction is close to the viewpoint direction and the ground-truth SDF is smaller than a threshold, we classify the point as near the front surface. Note that we use the ground-truth camera parameters and the gradients estimated from ground-truth SDF during training, and the predictions during testing.

3.3. Network losses

Our loss function is formulated as $L = L_B + L_{lap} + L_{sdf}$, where L_B, L_{lap}, and L_{sdf} denote the base loss, Laplacian loss, and SDF loss, respectively. Specifically, L_B is the L2-distance between the predicted base distance field f_B and the ground-truth SDF F_{SDF} over a set of sample points to learn the coarse shape. The SDF loss term L_{sdf} is the L1-distance between the fused implicit field f and the ground-truth SDF F_{SDF}; this term serves as a regularization for the displacement maps. Thus we have,

$$\begin{aligned} L_B &= \frac{1}{M}\sum_{i=1}^{M} \|f_B(p_i) - F_{SDF}(p_i)\|_2^2 \\ L_{sdf} &= \frac{1}{M}\sum_{i=1}^{M} |f(p_i) - F_{SDF}(p_i)| \end{aligned} \quad (4)$$

The Laplacian loss, L_{lap}, aims to minimize the error between the Laplacian of the predicted (front) displacements and the Laplacian of the ground-truth SDF. However, there exists a mismatch between the two Laplacians since our disentangled details are displacement maps defined in 2D while the ground-truth SDFs are defined in 3D.

To solve this problem, we estimate the *2D projection* of the ground-truth Laplacian, i.e., the Laplacian of the ground-truth SDF with respect to pixel positions on the image. This is reasonable since the single-view images are not sensitive to variations along the viewing direction. In addition, this enables us to obtain the ground-truth Laplacian from 2D normal maps, instead of computing it in 3D.

To project a point p in 3D space, we first transform it to $p' = (p'_x, p'_y, p'_z)$ in the camera's viewpoint, and then project it to the pixel position $u(p) = (u_x, u_y)$. The Laplacian of the front displacement map is

$$\triangle f_{DF}(u(p)) = \frac{\partial^2 f_{DF}(u(p))}{\partial (u_x)^2} + \frac{\partial^2 f_{DF}(u(p))}{\partial (u_y)^2}. \quad (5)$$

If p lies on the visible front surface, the ground-truth normal map provides its unit normal vector $N(u(p)) = \frac{\partial F_{SDF}(p)}{\partial p'}$, which equals to the gradient of the SDF with respect to the point coordinates p' in the camera view. With the camera parameters in the projecting operation, we obtain the gradient of the coordinates p' with respect to the pixel position $u(p)$, denoted by $\frac{\partial p'}{\partial u(p)}$. Therefore, we have the projected gradient of the SDF with respect to $u(p)$ as

$$N'(u(p)) = (N(u(p)) \cdot \frac{\partial p'}{\partial u_x}, N(u(p)) \cdot \frac{\partial p'}{\partial u_y}), \quad (6)$$

and the projected Laplacian (the ground-truth Laplacian) is

$$l(u(p)) = N(u(p)) \cdot \frac{\partial p'}{\partial^2 u_x} + N(u(p)) \cdot \frac{\partial p'}{\partial^2 u_y}. \quad (7)$$

Hence, the Laplacian loss is defined as

$$L_{lap} = \frac{1}{|P_F|} \sum_{p_i \in P_F} \|\triangle f_{DF}(u(p_i)) - l(u(p_i))\|_2^2 . \quad (8)$$

Weighted sampling. The loss terms are all defined on a set of sampled points. Unlike previous works, e.g., [5, 44], which randomly sample near object surfaces, we emphasize the importance of small-scale (e.g., thin) structures. Assuming a dense set of point-value pairs for an object, we define the density at each point as the number of points in its neighborhood with a prescribed radius. The interior points only count their neighbor points inside the object, so do the exterior points. During training, we sample an equal number of interior and exterior points with their densities as sampling weights. Such a weighted sampling strategy enables us to have more interior point-value pairs for the thin structures to better recover them during reconstruction.

4. Results, evaluation, and application

All the reconstruction networks are trained (over 13 categories) and tested on the ShapeNet Core dataset [3]. The training set comes from the ground-truth SDFs provided by DISN [44] and their rendered images including single-view

	CD	IoU	ECD-3D	ECD-2D
Baseline	0.0417	0.523	0.0735	3.304
WSamp	0.0340	0.587	0.0624	2.626
NoBack	0.0306	0.589	0.0525	1.802
NoLap	0.0302	0.601	0.0524	1.653
Full	**0.0297**	**0.613**	**0.0503**	**1.456**

Table 1. Quantitative evaluation for ablation study.

images and 2D normal maps. The ground-truth SDFs were randomly sampled on 32,768 points near the object surfaces with their signed distance values. In each iteration during training, we randomly select 2,048 points with our weighted sampling strategy to compute the loss and update the network. For the input images, we use all the views during training but use the one showing most of the details during testing to better evaluate the shape details.

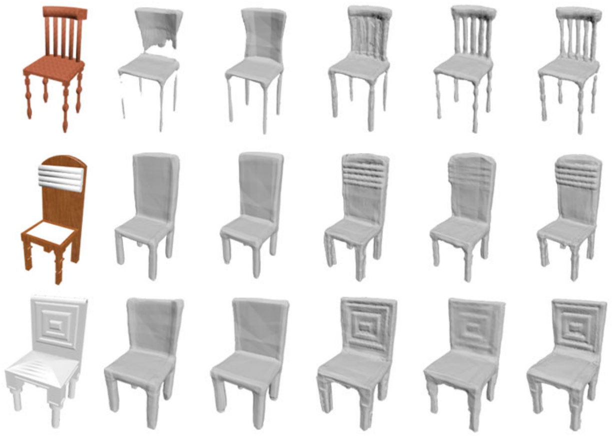
4.1. Evaluation metrics

For all the implicit 3D reconstructions we test, the final meshes are extracted via MarchingCubes in 128^3 resolution. To measure the overall reconstruction quality, we use Chamfer L_1 Distance (CD) [22] with 20K sampled points and Intersection of Union (IoU) in 32^3 resolution. It is worth noting however that despite their popularity, CD and IoU are not the best measures of *visual* reconstruction quality [14]. Also, they do not emphasize on small-scale details.

Since the focus of our work is on detail recovery, we employ the Edge Chamfer Distance (ECD) [4], which is defined as the CD between the edge points on the ground-truth shapes and the reconstructions. The "edgeness" of each point p_i is estimated as $\sigma(p_i) = min_{p_j \in \mathcal{N}_i} |n_i \cdot n_j|$, where $\mathcal{N}_i$ contains neighbors of point p_i, n_i and n_j are the unit normal vectors for points p_i and p_j. From 20K sampled points, we retrieve the nearest 10 neighbors for each point and retain the points with $\sigma(p_i) < 0.8$ to measure the small-scale details. Similarly, we develop a 2D version of the ECD metric, since we recover details observed from images. ECD-2D is defined as the CD between the edge pixels on the corresponding renderings. We apply the Canny edge detector [2] on the rendered 224×224 normal map of the reconstructed objects to obtain the edge pixels. The original ECD and its 2D version are denoted as ECD-3D and ECD-2D in our quantitative evaluation.

4.2. Ablation study

We conduct an ablation study to show how each component of D^{2}IM-Net contributes to detailed single-view 3D reconstruction. For the study, the networks are trained on the chair category from ShapeNet with the ground-truth camera parameters assumed given. The network options are:



(a) Input (b) Baseline (c) WSamp (d) NoBack (e) NoLap (f) Full

Figure 5. Qualitative results for ablation study: reconstructed objects rendered with the same camera parameters as input images.

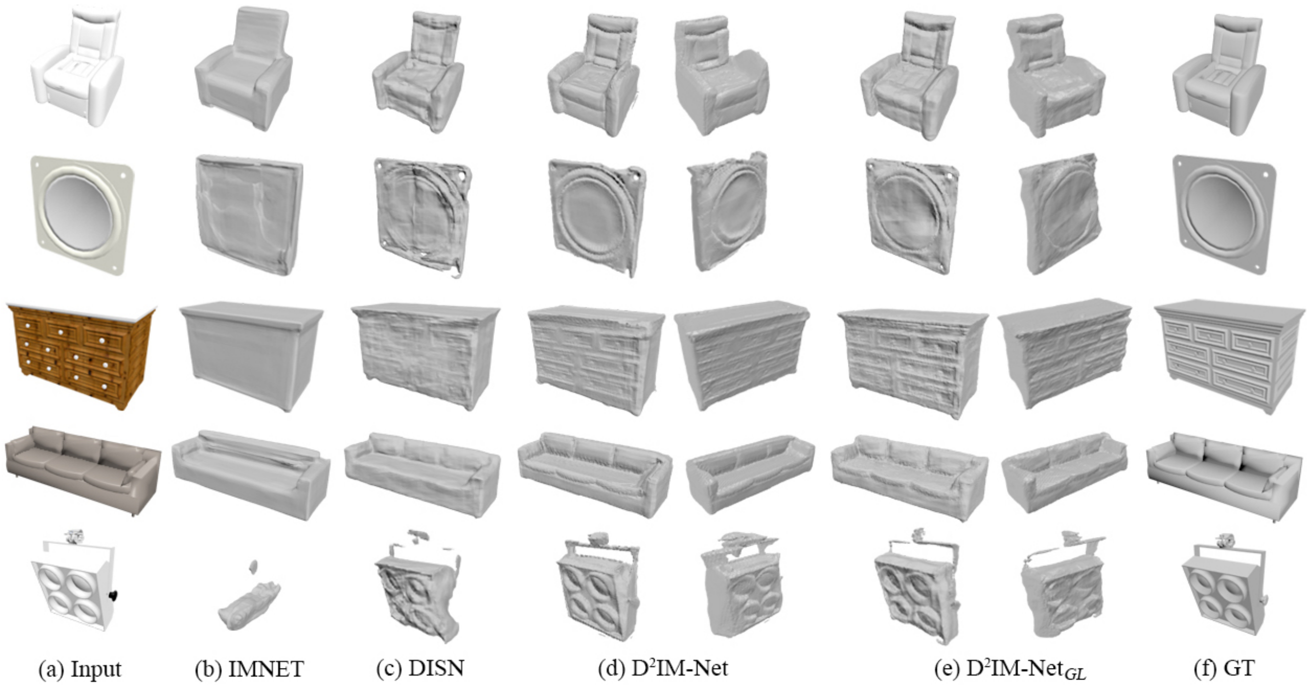
- *Baseline*: no detail decoder from D^{2}IM-Net and trained with uniform sampling and with loss L_{sdf} defined on the output of the base decoder.
- *WSamp*: weighted sampling to train the baseline.
- *NoBack*: no back displacement map prediction from D^{2}IM-Net; the predicted base distance field is fused with only the front displacement map.
- *NoLap*: only removing L_{lap} loss from D^{2}IM-Net; both NoBack and NoLap use weighted sampling.
- *Full*: all-component D^{2}IM-Net as describe in Figure 2.

Figure 5 and Table 1 provide qualitative and quantitative comparison results, respectively. As we can see, weighted sampling helps reconstruct thin volumes, with the detail decoder providing even more improved results on topological structures, while surface details are best recovered with the Laplacian loss (see NoLap vs. Full or NoBack).

4.3. Comparison

In our comparison to the state of the art, we focus on implicit models which have yielded the best reconstruction quality so far. In addition to IMNET [5], which is a baseline corresponding to the base decoder branch of D^{2}IM-Net, we focus on comparing to DISN [44], which is, to the best of our knowledge, the top single-view reconstruction network to date in terms of detail recovery. We also test a slight variant to D^{2}IM-Net, called D^{2}IM-Net$_{GL}$, which takes both global and local features as input to its base decoder. All the methods are trained and tested on the same dataset.

As shown in Figure 6, IMNET generally obtains good coarse reconstruction, but misses most details. DISN does a better job in terms of recovering topological structures and shape boundaries, but typically blurs surface features. Both versions of D^{2}IM-Net visually outperform IMNET



(a) Input (b) IMNET (c) DISN (d) D^{2}IM-Net (e) D^{2}IM-Net$_{GL}$ (f) GT

Figure 6. Qualitative comparison between reconstruction results by IMNET [5], DISN [44], D^{2}IM-Net, and D^{2}IM-Net$_{GL}$.

		plane	bench	box	car	chair	display	lamp	speaker	rifle	sofa	table	phone	boat	Mean
IoU ↑	IMNET	0.5200	0.5133	0.4581	0.7653	0.5411	*0.5185*	0.4168	**0.5194**	0.5643	0.6386	0.5083	0.6701	0.5631	0.5536
	DISN	0.5362	0.5403	0.4615	*0.8105*	*0.5539*	0.4879	0.3791	0.4958	**0.7237**	*0.6520*	**0.5629**	*0.7071*	**0.6566**	0.5821
	D^{2}IM-Net	**0.5584**	**0.5495**	**0.4860**	0.7980	**0.5613**	**0.5272**	*0.4213*	*0.5175*	*0.6813*	**0.6535**	0.5367	**0.7616**	0.6339	**0.5912**
	D^{2}IM-Net$_{GL}$	*0.5553*	*0.5425*	*0.4760*	**0.8114**	0.5441	0.5112	**0.4495**	0.5031	0.6626	0.6437	*0.5475*	0.6966	*0.6381*	*0.5832*
CD ↓	IMNET	0.0426	0.0382	0.0503	0.0437	0.0376	0.0479	*0.0557*	0.0632	0.0329	0.0475	0.0432	0.0317	0.0443	0.0445
	DISN	0.0398	0.0351	0.0412	**0.0308**	*0.0326*	0.0462	0.0770	0.0647	**0.0199**	**0.0366**	*0.0316*	0.0282	**0.0312**	0.0396
	D^{2}IM-Net	**0.0358**	**0.0312**	**0.0385**	0.0348	0.0329	**0.0422**	*0.0557*	**0.0561**	0.0244	0.0391	0.0356	**0.0245**	*0.0339*	*0.0373*
	D^{2}IM-Net$_{GL}$	**0.0358**	*0.0337*	*0.0386*	*0.0313*	**0.0308**	*0.0427*	**0.0549**	*0.0572*	*0.0242*	*0.0375*	**0.0310**	*0.0270*	*0.0339*	**0.0368**
ECD-3D ↓	IMNET	0.0789	0.0685	0.0872	0.0872	0.0661	0.0820	0.0995	0.1080	0.0674	0.0790	0.0710	0.0724	0.0823	0.0807
	DISN	0.0684	0.0573	0.0697	*0.0680*	0.0564	0.0765	0.1127	0.1077	*0.0350*	*0.0606*	*0.0601*	0.0708	0.0583	0.0694
	D^{2}IM-Net	**0.0567**	**0.0477**	**0.0661**	0.0728	*0.0523*	**0.0674**	*0.0918*	**0.0909**	**0.0343**	0.0642	0.0630	**0.0609**	*0.0568*	**0.0634**
	D^{2}IM-Net$_{GL}$	*0.0598*	*0.0516*	*0.0691*	**0.0646**	**0.0504**	*0.0713*	**0.0897**	*0.0973*	0.0357	**0.0602**	**0.0567**	*0.0660*	**0.0534**	*0.0635*
ECD-2D ↓	IMNET	2.532	2.845	4.467	3.344	2.703	3.230	3.361	4.198	3.138	2.979	2.846	2.422	3.046	3.162
	DISN	2.672	2.209	2.250	*2.042*	1.983	3.156	4.863	3.338	*1.353*	2.062	2.065	2.259	*2.003*	2.481
	D^{2}IM-Net	*1.991*	**1.666**	*1.794*	2.072	*1.707*	**1.954**	*3.157*	**2.636**	**1.277**	*2.014*	*1.880*	**1.617**	**1.730**	**1.961**
	D^{2}IM-Net$_{GL}$	**1.982**	*1.774*	**1.739**	**1.767**	**1.584**	*2.675*	**3.009**	*2.715*	1.766	**1.776**	**1.737**	*2.142*	2.269	*2.072*

Table 2. Quantitative comparison results: IoU at 32^3 resolution; CD and ECD-3D on 20K sample points; ECD-2D on 224 × 224 rendered normal maps. Top numbers are in bold and second place is indicated in italic.

and DISN, especially over small-scale, high-frequency details. This is consistent with the quantitative results, provided by ECD-3D and ECD-2D measures, as shown in Table 2. Overall, Table 2 shows that both versions of D^{2}IM-Net also outperform IMNET and DISN quantitatively, in terms of both overall reconstruction quality (CD and IoU) and edge feature recovery (ECD-3D and ECD-2D).

We generally find D^{2}IM-Net to slightly outperform D^{2}IM-Net$_{GL}$ in visual quality (see Figure 6), especially in terms of surface details. This may be due to the redundancy in using latent (local) features in both the base and detail decoders by the latter. D^{2}IM-Net$_{GL}$ appears to perform better on thin structures. Results from Figure 7 support these findings, where we show reconstructions from several online images, with no 3D ground-truth shapes.

4.4. Application: detail transfer and reconstruction

With disentangled coarse shapes and details in the context of 3D reconstruction, enabled by our work, it becomes possible to *transfer* geometric details or features from images to images and then obtain a final 3D outcome.

Detail transfer. Given a pair of single-view images of different objects (e.g., two chairs), our network predicts their disentangled coarse shapes and details, respectively. Detail transfer involves fusing the disentangled *source* details with the *target* coarse shape. In the fusion stage, for each point p, we sum up its base distance $f_B(p)$ from the target image and the queried source detail displacement $f_{DF}(u_S(q))$ or $f_{DB}(u_S(q))$ with a learned 3D correspondence $q = C_{T \rightarrow S}(p)$, where u_S is the projection operation

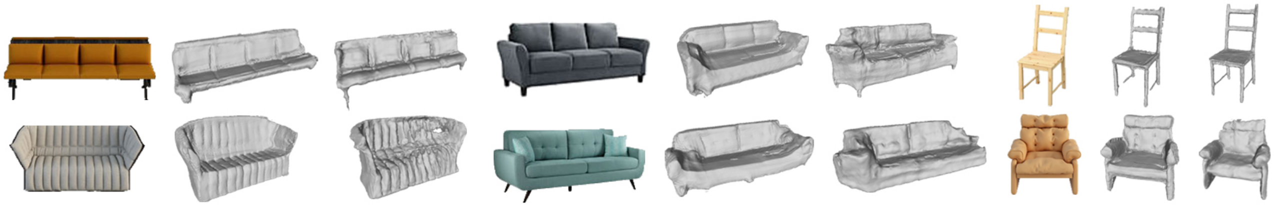


Figure 7. Reconstruction results from single-view images "in the wild" using D^{2}IM-Net (left) and D^{2}IM-Net$_{GL}$ (right).

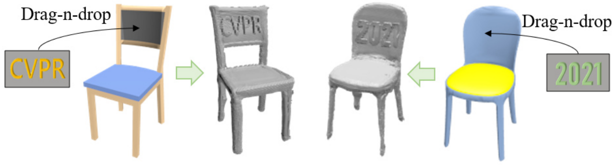


Figure 8. When a logo image, e.g., of "CVPR", is "drag-n-dropped" onto a chair image, we obtain a reconstructed 3D chair model (shown in a view that is different from that of the input image) with surface features resembling the input logo.

with camera parameters predicted from the source image.

Our method allows such a detail transfer for a specified semantic part, by fusing the displacement values from the source image for the points near this part, and displacement values from the target image otherwise. Results in Figure 1 show surface detail transfer from the source chair images (top row) to the target chair images (left column) on the chairs' backs while preserving the coarse shapes.

In the implementation, we use a pre-trained semantic segmentation network [39] on the two coarse shapes to build the correspondence $q = C_{T\to S}(p)$. The corresponding segmented parts imply a point-wise correspondence within the local volumes. Specifically, for each point p, we compute its local coordinates with respect to the frame defined by the target part it belongs to, and then map it back to the world coordinates q based on the frame of the source part. The local frames are origined at the center of the axis-aligned bounding boxes of each part with fixed axes directions.

"Paste-n-reconstruct". Under the same spirit of image-to-image detail transfer but in a slightly different task setting, Figure 8 shows how a small image logo can be drag-n-dropped onto another image, where the logo content is pasted onto the target image and then a 3D shape can be reconstructed with the pasted logo features.

To implement this, the target image (the chair in Figure 8) goes through the D^{2}IM-Net encoder and base decoder to provide the base distance field for the coarse shape. On the other hand, both the target image and the (source) logo image go through the same encoder and detail decoder to predict their displacement maps. With the separately predicted (or pre-defined) camera parameters for each image, we fuse the base distance field and all the displacement maps (only front displacement maps of the logo images) by the projection with their camera parameters. When the foreground masks of the logo images are given, we can crop the foreground displacements for a better visualization.

5. Conclusion, limitation, and future work

We tackle perhaps the "last mile" in single-view 3D reconstruction, i.e., to recover small-scale geometric details, especially surface features. This is a deceptively difficult problem as we seek a network that generalizes to shapes across multiple categories (13 categories in ShapeNet in our experiments), not a method that "overfits" to specific inputs. Note also that we do not rely on symmetry priors or color/material cues. Our key idea is to learn a detail disentangled representation with a dedicated loss for surface details, defined in the Laplacian domain.

One main limitation of our current method is the assumption that the surface details are defined by a height field over a mostly flat surface. One implication of this is that geometric details corresponding to "overhangs" are precluded. Another implication is that, technically, our network would be unable to recover surface details over surfaces that are sufficiently curved. In practice, we have found that our network is able to recover surface details over mildly curved surfaces, as the example at the bottom-left of Figure 7 demonstrates. A second limitation is that our Laplacian loss is defined only on the front surface of the recovered shape. Furthermore, even on the front, we can notice that the reconstructions obtained often look slightly worse when viewed from an different angle as in the input image. Possible remedies to this include more accurate view parameter inference and consideration of symmetry priors [46].

In addition to addressing the above limitations, we are also interested in expanding the use of neural Laplacian domain processing to other shape representations such as voxels, point clouds, and meshes, as well as exploring disentangled learning of geometric details for a variety of other applications including multi-modal detail transfer, 3D superresolution, and generative shape modeling.

Acknowledgements. We thank the anonymous reviewers for their valuable comments, and Wallace Lira for helpful discussions. This work was supported in part by an NSERC grant (No. 611370) and a gift from Adobe.

References

[1] M. Atzmon and Y. Lipman. SAL: Sign agnostic learning of shapes from raw data. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2565–2574, 2020. 3

[2] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986. 6

[3] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015. 3, 5

[4] Zhiqin Chen, Andrea Tagliasacchi, and Hao Zhang. Bsp-net: Generating compact meshes via binary space partitioning. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 6

[5] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5939–5948, 2019. 1, 3, 5, 6, 7

[6] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3D-R2N2: A unified approach for single and multi-view 3d object reconstruction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016. 3

[7] Emily L Denton, Soumith Chintala, Rob Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in neural information processing systems*, pages 1486–1494, 2015. 4

[8] Selim Esedog, Steven Ruuth, Richard Tsai, et al. Diffusion generated motion using signed distance functions. *Journal of Computational Physics*, 229(4):1017–1042, 2010. 2, 4

[9] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan Russell, and Mathieu Aubry. Atlasnet: A papier-mâché approach to learning 3d surface generation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[10] Christian Häne, Shubham Tulsiani, and Jitendra Malik. Hierarchical surface prediction for 3d object reconstruction. In *Proceedings of the International Conference on 3D Vision (3DV)*. 2017. 3

[11] Tong He, John Collomosse, Hailin Jin, and Stefano Soatto. Geo-pifu: Geometry and pixel aligned implicit functions for single-view human reconstruction. *arXiv preprint arXiv:2006.08072*, 2020. 3

[12] Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, and Thomas Funkhouser. Local implicit grid representations for 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6001–6010, 2020. 3

[13] Yue Jiang, Dantong Ji, Zhizhong Han, and Matthias Zwicker. Sdfdiff: Differentiable rendering of signed distance fields for 3d shape optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1251–1261, 2020. 3

[14] Jiongchao Jin, Akshay Gadi Patil, Zhang Xiong, and Hao Zhang. DR-KFS: A differentiable visual similarity metric for 3d shape reconstruction. In *Proc. of ECCV*, 2020. 6

[15] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 624–632, 2017. 4

[16] Shaohua Li, Xinxing Xu, Liqiang Nie, and Tat-Seng Chua. Laplacian-steered neural style transfer. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1716–1724, 2017. 4

[17] Gidi Littwin and Lior Wolf. Deep meta functionals for shape representation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1824–1833, 2019. 3

[18] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *NeurIPS*, 2020. 3

[19] Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2019. 4

[20] Shichen Liu, Shunsuke Saito, Weikai Chen, and Hao Li. Learning to infer implicit surfaces without 3d supervision. In *Advances in Neural Information Processing Systems*, pages 8295–8306, 2019. 3

[21] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM Trans. Graph.*, 21(4):163–169. 4

[22] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019. 1, 3, 6

[23] Mateusz Michalkiewicz, Jhony K Pontes, Dominic Jack, Mahsa Baktashmotlagh, and Anders Eriksson. Deep level sets: Implicit surface representations for 3d shape inference. *arXiv preprint arXiv:1901.06802*, 2019. 3

[24] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 3

[25] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3504–3515, 2020. 3

[26] Chengjie Niu, Jun Li, and Kai Xu. Im2struct: Recovering 3d shape structure from a single rgb image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4521–4529, 2018. 3

[27] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019. 1, 3

[28] Stefan Popov, Pablo Bauszat, and Vittorio Ferrari. Corenet: Coherent 3d scene reconstruction from a single rgb image. In *ECCV*, pages 366–383. Springer, 2020. 3

[29] Stephan R Richter and Stefan Roth. Matryoshka networks: Predicting 3d geometry via nested shape layers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1936–1944, 2018. 3

[30] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2304–2314, 2019. 3

[31] Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization. In *CVPR*, pages 84–93, 2020. 3, 4

[32] Gopal Sharma, Difan Liu, Evangelos Kalogerakis, Subhransu Maji, Siddhartha Chaudhuri, and Radomír Měch. Parsenet: A parametric surface fitting network for 3d point clouds, 2020. 4

[33] Vincent Sitzmann, Julien N.P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *arXiv*, 2020. 3

[34] Yongliang Tang, Weiguo Gong, Xi Chen, and Weihong Li. Deep inception-residual laplacian pyramid networks for accurate single-image super-resolution. *IEEE Transactions on Neural Networks and Learning Systems*, 31(5):1514–1528, 2019. 4

[35] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2088–2096, 2017. 3

[36] Maxim Tatarchenko, Stephan R Richter, René Ranftl, Zhuwen Li, Vladlen Koltun, and Thomas Brox. What do single-view 3d reconstruction networks learn? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3405–3414, 2019. 3

[37] Shubham Tulsiani, Tinghui Zhou, Alexei A Efros, and Jitendra Malik. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2626–2634, 2017. 3

[38] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 52–67, 2018. 3, 4

[39] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (TOG)*, 2019. 8

[40] Francis Williams, Teseo Schneider, Claudio Silva, Denis Zorin, Joan Bruna, and Daniele Panozzo. Deep geometric prior for surface reconstruction. In *CVPR*, pages 10130–10139, 2019. 3

[41] Jiajun Wu, Yifan Wang, Tianfan Xue, Xingyuan Sun, Bill Freeman, and Josh Tenenbaum. Marrnet: 3d shape reconstruction via 2.5 d sketches. In *Advances in Neural Information Processing Systems*, pages 540–550, 2017. 3

[42] Jiajun Wu, Chengkai Zhang, Xiuming Zhang, Zhoutong Zhang, William T Freeman, and Joshua B Tenenbaum. Learning shape priors for single-view 3d completion and reconstruction. In *Proceedings of the European Conference on Computer Vision*, pages 646–662, 2018. 3

[43] Rundi Wu, Yixin Zhuang, Kai Xu, Hao Zhang, and Baoquan Chen. Pq-net: A generative part seq2seq network for 3d shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 829–838, 2020. 3

[44] Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. In *Advances in Neural Information Processing Systems*, pages 492–502, 2019. 1, 3, 5, 6, 7

[45] Yifan Xu, Tianqi Fan, Yi Yuan, and Gurprit Singh. Ladybird: Quasi-monte carlo sampling for deep implicit field based 3d reconstruction with symmetry. *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 3

[46] Yuan Yao, Nico Schertler, Enrique Rosales, Helge Rhodin, Leonid Sigal, and Alla Sheffer. Front2back: Single view 3d shape reconstruction via front to back prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 531–540, 2020. 4, 8

[47] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Ronen Basri, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. *NeurIPS*, 2020. 3

[48] Chuhang Zou, Ersin Yumer, Jimei Yang, Duygu Ceylan, and Derek Hoiem. 3d-prnn: Generating shape primitives with recurrent neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 900–909, 2017. 3