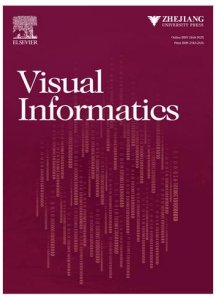
Journal Pre-proof



MDISN: Learning multiscale deformed implicit fields from single images

Yujie Wang, Yixin Zhuang, Yunzhe Liu, Baoquan Chen

PII: S2468-502X(22)00016-X
DOI: https://doi.org/10.1016/j.visinf.2022.03.003
Reference: VISINF 133

To appear in: *Visual Informatics*

Received date : 25 December 2021
Revised date : 21 March 2022
Accepted date : 22 March 2022

Please cite this article as: Y. Wang, Y. Zhuang, Y. Liu et al., MDISN: Learning multiscale deformed implicit fields from single images. *Visual Informatics* (2022), doi: https://doi.org/10.1016/j.visinf.2022.03.003.

Title Page (with Author Details)

MDISN: Learning Multiscale Deformed Implicit Fields from Single Images

Yujie Wang[a,b], Yixin Zhuang[a,*], Yunzhe Liu[a], Baoquan Chen[a,*]

[a]*Peking University*
[b]*Shandong University*

*Corresponding author

MDISN: Learning Multiscale Deformed Implicit Fields from Single Images

Yujie Wang[a,b], Yixin Zhuang[a,*], Yunzhe Liu[a], Baoquan Chen[a,*]

[a]*Peking University*
[b]*Shandong University*

Abstract

We present a multiscale deformed implicit surface network (MDISN) to reconstruct 3D objects from single images by adapting the implicit surface of the target object from coarse to fine to the input image. The basic idea is to optimize the implicit surface according to the change of consecutive feature maps from the input image. And with multi-resolution feature maps, the implicit field is refined progressively, such that lower resolutions outline the main object components, and higher resolutions reveal fine-grained geometric details. To better explore the changes in feature maps, we devise a simple field deformation module that receives two consecutive feature maps to refine the implicit field with finer geometric details. Experimental results on both synthetic and real-world datasets demonstrate the superiority of the proposed method compared to state-of-the-art methods. Pre-trained models and codes will be released for research purposes upon paper acceptance.

Keywords: Single-view 3D Reconstruction, Implicit Neural Representation, Multiscale Deformation

1. Introduction

3D reconstruction from a single RGB image is one of the fundamental problems in computer vision. Recently, many learning-based methods have been proposed and achieved significant progress in single-image shape reconstruction. They generate various shape representations, including point clouds, meshes, voxels, and implicit fields,

*Corresponding author

from which implicit models [1, 2, 3, 4, 5, 6, 7] achieves much better reconstruction quality compared to others.

The implicit fields for image-based 3D reconstruction seek a mapping from a 3D point and the latent vector of image feature to the corresponding value, e.g., signed distance. To better align the generated shapes with 3D objects in the image, some works have specifically addressed pixel-aligned 3D reconstruction [8, 9, 10]. They concerned the reconstructed 3D shapes precisely aligned to the acquired images at the pixel level. To achieve this, they assign each 3D coordinate with a *local image feature* to describe its local property. Local image features are obtained by projecting 3D points to the image plane using a camera pose. With the help of local image features, the generated surfaces can be better aligned to the image, leading to a significant performance improvement.

A local image feature is composed of multiple feature vectors extracted from multiscale image feature maps [8, 9, 10]. Intuitively, at the high resolution of the feature map (that has smaller receptive fields), the feature describes local details of the object, while at the low resolution (larger receptive fields), it only tells the global structure information. And by increasing the resolution of the feature map, the geometric details appear from coarse to fine. Also, a lower resolution shape with simpler geometry and topology is less sensitive to the variation of image appearance and camera poses and can be fit easier with less complex functions. Thus it is natural to construct the surface from a simple structure and iteratively adjust it with increasing surface details. To this end, we propose a new network for implicit surface reconstruction that leverages multiscale image feature maps for progressively surface optimization.

Starting from an initial implicit field, we iteratively refine it by considering the changes of two consecutive feature maps. Figure 1 shows several examples of the feature maps and the coarse-to-fine results, where the feature maps are blurry at low resolution with very little shape information, and at higher resolution, the outlines and details of the shape become clearer. Note that the feature maps are high-dimensional tensors, and we use grey images for better illustration.

To better explore the changes in the feature maps, we develop a field deformation module that transforms the input field conditioning on two consecutive feature maps,

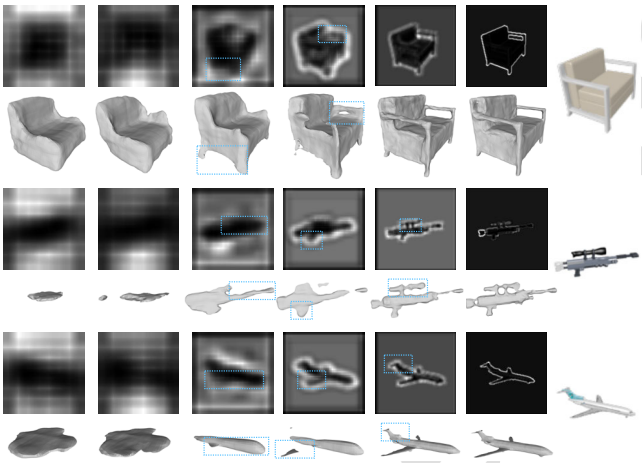


Figure 1: Examples of the reconstructed shapes from single images using our method. The reconstruction is adapted stepwise from coarse to fine. The main components (highlighted by boxes) of the objects can be reconstructed with low-resolution image features, and by increasing the resolution of feature maps, finer geometric details are restored. Note that the refinement of shapes follows the changes in consecutive feature maps.

i.e., the one with a lower resolution corresponds to the input field, and the other with a higher resolution provides more information for refinement upon the lower one. Since there are no ground truth shapes that correspond to the individual scale of the feature maps, we can only apply supervision on the output layer for training the network. Thus there is no supervision applied at each step of deformation. Even though, as shown in Figure 1, the sequential shapes reconstructed change smoothly with increasing details from low resolution to high resolution, especially at higher resolutions. And each reconstructed shape is highly related to the corresponding feature map from the appearance.

The multiscale optimization lends the flexibility to control the size of the network and reconstruct the field from any selected resolution of image feature maps while terminating at any higher resolutions. Moreover, exploring image feature maps and their relations provides an intuitive way for understanding the creation procedure of

3D shapes and how to improve them.

We extensively evaluate our model on both synthetic and real-world 3D shape datasets – the ShapeNet Core dataset [11] and the Pix3D dataset [12], using a combination of standard metrics (e.g., Chamfer Distance, Earth Mover's Distance, and Intersection over Union). The experiments demonstrate that our method can provide state-of-the-art 3D shape reconstruction results from single images compared to previous work. Ablation experiments showcase the effectiveness of the core components of our network.

2. Related Work

We briefly review the extensive research on the task of single-view 3D reconstruction based on Deep Learning. Many methods have been proposed to learn 3D representations, including Points [13, 14, 15], Voxels [16, 17, 18, 19], Meshes [20, 21, 22, 23, 24, 25] and Primitive [26, 27, 28], by the supervisions on 3D groundtruth shape or, even more difficult, the 2D images [29, 30, 31, 32, 33, 14].

For explicit surface representation, AtlasNet [20] represents 3D shapes as a union of multiple surface patches predicted by multiple learned multilayer perceptrons (MLPs). Pixel2Mesh [21] generates shapes by deforming an ellipsoid to the target. Since the ellipsoid is genus-zero and the deformation does not change the edge connections, the reconstruction always has the same topology. 3DN [22] also deforms a template mesh to the target. They train a differentiable mesh sampling operator that moves the sampled points to the target position.

Unlike explicit 3D representations, which have little flexibility in changing shape resolution and topology, implicit functions for 3D objects have shown advantages in representing complicated geometry [1, 2, 8, 9, 34, 35, 36, 37, 10, 5, 38]. It commonly uses an MLP-based neural network to generate implicit fields of 3D objects, as introduced by ImNet [2], OccNet [1] and DeepSDF [3]. The implicit results show great improvement in contrast to the explicit surface representations. Specifically, OccNet [1] generates an implicit volumetric shape by inferring the probability of whether each grid cell is empty or occupied. The shape resolution can be iteratively refined by

upsampling on the cells of interest.

More recently, some works have specifically addressed pixel-aligned 3D reconstruction [8, 9, 10]. In addition to capturing the global shape structure, they introduce a pixel-level alignment between shape and image. DISN [8] extracts a local image feature for each 3D point sampling, and then the local image feature is extended by including the symmetric point of the point sampling in Ladybird [9]. In this work, we explore image feature maps for iteratively surface refinement in the framework of the implicit surface network to achieve better 3D reconstruction from single images.

The most related work that also leverages multiscale implicit fields comes from [7]. The work generates multiscale implicit surfaces for shape generation, completion, and super-resolution tasks. Different from it, we design MDISN for image-based shape reconstruction, focusing on building the connection between 2D images and 3D implicit fields. MDISN extends the 2D-to-3D mapping to sequential deformations, achieving better exploration of multiscale image feature maps, and finally leads to significant improvement of the 3D reconstruction.

3. Method

3.1. Overview

Given an RGB image of an object, our goal is to reconstruct the 3D object that precisely aligns with the input image. We represent the shape by the signed distance function and approximate it with an MLP-based neural network. To generate the implicit surface, we sample a set of points from the canonical shape space such that each 3D point sampling $p = (x, y, z) \in \mathbb{R}^3$, and predict the corresponding signed distance s. Then the surface is extracted as the isosurface of $SDF(\cdot) = 0$ by the Marching Cubes algorithm [39].

From a given image I, the prediction of SDF conditions on the global feature of I. We denote the image encoder as m and the signed distance function as f, then SDF is generated as follows,

$$f(p, F_g) = s, s \in \mathbb{R}, \tag{1}$$

from which the F_g is a global image feature.

To generate the pixel-aligned implicit field, we obtain a local image feature to describe individual point samplings. With both global and local image features, we generate SDF as follows,

$$f(p, F_l(a), F_g) = s, s \in \mathbb{R}, \tag{2}$$

where $a = \pi(p)$ is the corresponding image pixel of point sampling p, and $F_l(a) = m(I(a))$ is the local image feature. The local image features are retrieved at the location a of the 2D projection of the 3D point samplings over the camera pose c. We address the details of camera projection in Appendix A.3 (3D-to-2D Camera Projection).

A local image feature $F_l(a)$ is composed of multiple features from the multiresolution feature maps of the image encoder m. Specifically, we use a VGG-style image convolutional network that has six convolutional layers. The resulting feature maps are $l_1, ..., l_6$, from which each feature map has a different resolution. We resize the multiscale feature maps to the same resolution using bilinear interpolation. Then we apply a deformation module h to each pair of consecutive feature maps to transform the implicit field. For simplicity, we initialize the field with a template generator τ instead of using the global image feature F_g that is encoded from image feature maps. The formulation with deformation modules can be written in recurrence as follows,

$$\begin{aligned} s_0(p) &= \tau(p), \\ s_1(p) &= f_1(p, s_0(p), l_1(a)), \\ s_{n+1}(p) &= f_{n+1}(p, s_n(p), l_n(a), l_{n+1}(a)), n = 1, ..., 5 \end{aligned} \tag{3}$$

where τ generates initial implicit field and f_n deforms the fields. $l_n(a) \in F_l(a)$ is the local image feature utilized at the n-th feature map and at the image location a.

As suggested by Ladybird, we utilize two local image features to describe a 3D coordinate. They are extracted from the 2D projections of the 3D coordinate and its self-reflectional symmetric coordinate and concatenated as a joint feature. For a schematic representation of the proposed model, see Figure 2.

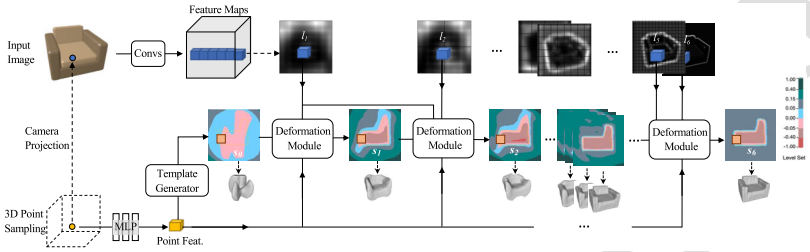


Figure 2: The workflow of our method. Given an input image, our network progressively refines the signed distance fields, from coarse to fine, to reconstruct the underlying 3D object. The field is initialized by a template generator and iteratively refined by a series of deformation modules that follow the changes in the consecutive image feature maps. For better illustration, the 3D tensors of feature maps are shown by gray images.

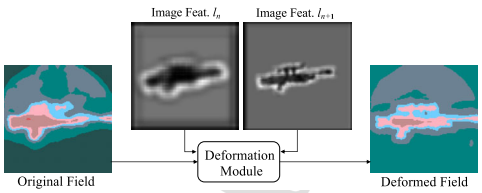


Figure 3: Illustration of implicit field deformation. The deformation is guided by the changes of feature images.

3.2. Implicit Field Deformation

We use a deformation module at the individual scale of the image feature map to gradually refine the generated implicit field towards the target. To better explore the image features and also to emphasize the smoothness of the field deformation, we develop the deformation module to use the changes of the image feature maps, as shown in Figure 3. The consecutive feature images contain different shape information, e.g., image Feat. l_n implies a coarse layout of the underlying object, while in image Feat. l_{n+1} the object components, such as the scope, can be seen more clearly.

In addition to feature maps, we also introduce a state vector into our deformation module to store the contextual information of the sequential deformations. It is similar to the recurrent neural network, which uses a latent code to memorize the historical messages. We then combine the implicit field, image feature maps, and state vector into

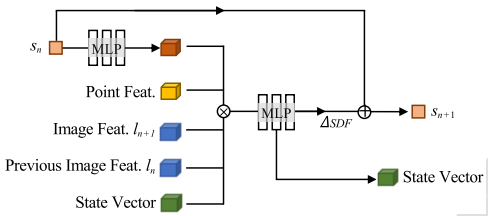


Figure 4: The implicit field deformation module takes as input the signed distance SDF_n of a point sampling, together with its point feature, its corresponding local image features l_n, l_{n+1} and a state vector, and outputs the deformed signed distance SDF_{n+1}. The image feature l_n from the previous stage is also included to control the deformation in accordance with the transition from feature l_n to l_{n+1}. $\otimes$ and $\oplus$ mean concatenate and sum operations.

a unified vector and use it to predict a residual field for modification and an updated state vector. The residual field is added to the input field to obtain the refined field. A schematic illustration of the deformation module can be found in Figure 4.

3.3. Implementation Details

Our network consists of a 3-layer Multi-layer Perceptron (MLP) as a point sampling embedding network, a 3-layer MLP as a template generator, six 3-layer MLP as implicit field deformation modules, and a fully convolutional network of VGG-16 [40] as an image encoder. We use the official VGG pre-trained model to initialize the convolutional modules and the PyTorch default initialization scheme for other modules. Please refer to the Appendix for more implementation details.

Feature Maps. We extract feature maps from multiple layers of a convolutional network. As shown in Figure 5, we use six feature maps from the vgg16 network that have different resolutions.

Loss Function. We develop a loss function for training our network. For each image I from the image collection $\mathcal{I}$, our network generates the underlying 3D object as an implicit field, i.e., for each 3D coordinate, the predicted signed distance $s(p) = s_6(p)$, as shown in Equation 3. We use L_1 norm to compute the distance between the

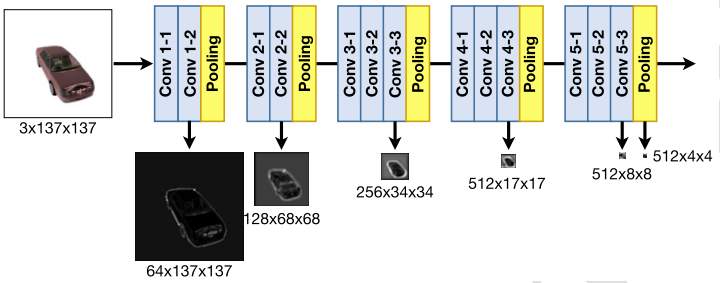


Figure 5: Image feature maps extracted from the VGG-16 convolutional network at multiple layers.

generated implicit fields and the corresponding ground truth fields,

$$L_{SDF} = \sum_{I \in \mathcal{I}} \sum_{p} \omega |s(p) - SDF(p)|, \tag{4}$$

where SDF denotes the ground truth SDF. ω is set to ω_1, if $SDF(p) < \delta$, and ω_2 if not. For all our experiments, the parameters are set to $\omega_1 = 4, \omega_2 = 1$, and $\delta = 0.01$.

4. Experiments

In this section, we compare the qualitative and quantitative performance of MDISN with state-of-the-art methods. We also perform an ablation study for the core component of our network to show the impact of the multiscale reconstruction and the deformation module. Further, we provide more results using the different combinations of feature maps.

Dataset. We use the synthetic dataset, ShapeNet Core [11], and the real dataset, Pix3D [12], for training and evaluation. In testing, we use the estimated camera parameters for evaluation on the ShapeNet dataset. For the Pix3D dataset, following Ladybird[9], we use the ground truth camera parameters and image masks.

Evaluation Metrics. We use the standard metrics, including Chamfer Distance (CD), Earth Mover's Distance (EMD), and Intersection over Union (IOU), to measure the similarity between the generated meshes and the ground truth meshes. CD and EMD

Table 1: Quantitative results on the ShapeNet Core dataset.

Metrics	Methods	plane	bench	cabinet	car	chair	display	lamp	speaker	rifle	sofa	table	phone	watercraft	mean
	AtlasNet	39.2	34.2	20.7	22.0	25.7	36.4	21.3	23.2	45.3	27.9	23.3	42.5	28.1	30.0
	IMNET	55.4	49.5	51.5	**74.5**	52.2	56.2	29.6	52.6	52.3	64.1	45.0	70.9	56.6	54.6
	OccNet	54.7	45.2	**73.2**	73.1	50.2	47.9	37.0	**65.3**	45.8	67.1	**50.6**	70.9	52.1	56.4
IOU(%)↑	DISN	57.5	52.9	52.3	74.3	54.3	56.4	34.7	54.9	59.2	65.9	47.9	72.9	55.9	57.0
	Ladybird	60.0	53.4	50.8	**74.5**	55.3	57.8	36.2	55.6	61.0	**68.5**	48.6	**73.6**	**61.3**	58.2
	Ours$_{cam}$	**60.4**	**54.6**	52.2	**74.5**	**55.6**	**59.4**	**38.2**	55.8	**62.2**	**68.5**	48.6	73.5	60.4	**58.8**
	Ours	68.1	62.5	59.3	80.5	66.4	67.4	54.6	64.2	74.2	74.3	60.0	79.2	68.6	67.6
	AtlasNet	3.39	3.22	3.36	3.72	3.86	3.12	5.29	3.75	3.35	3.14	3.98	3.19	4.39	3.67
	IMNET	2.90	2.80	3.14	2.73	3.01	2.81	5.85	3.80	2.65	2.71	3.39	2.14	2.75	3.13
	OccNet	2.75	2.43	3.05	**2.56**	2.70	2.58	3.96	3.46	2.27	**2.35**	2.83	2.27	2.57	2.75
EMD(x100)↓	DISN	2.67	2.48	3.04	2.67	2.67	2.73	4.38	3.47	2.30	2.62	3.11	**2.06**	2.77	2.84
	Ladybird	2.48	2.29	3.03	2.65	2.60	2.61	4.20	3.32	2.22	2.42	2.82	**2.06**	2.46	2.71
	Ours$_{cam}$	**2.33**	**2.17**	**2.91**	2.70	**2.52**	**2.50**	**3.67**	**3.30**	**2.17**	2.43	**2.81**	2.11	**2.42**	**2.62**
	Ours	1.88	1.94	2.62	2.42	2.20	2.12	2.82	2.87	1.64	2.17	2.53	1.85	2.01	2.24
	AtlasNet	5.98	6.98	13.76	17.04	13.21	7.18	38.21	15.96	**4.59**	8.29	18.08	6.35	15.85	13.19
	IMNET	12.65	15.10	11.39	8.86	11.27	13.77	63.84	21.83	8.73	10.30	17.82	7.06	13.25	16.61
	OccNet	7.70	6.43	9.36	5.26	7.67	**7.54**	26.46	17.30	4.86	6.72	10.57	7.17	9.09	9.70
CD(x1000)↓	DISN	9.96	8.98	10.19	5.39	7.71	10.23	25.76	17.90	5.58	9.16	13.59	6.40	11.91	10.98
	Ladybird	5.85	**6.12**	9.10	**5.13**	7.08	8.23	21.46	14.75	5.53	6.78	**9.97**	**5.06**	6.71	8.60
	Ours$_{cam}$	**5.77**	6.29	**8.78**	5.21	**6.68**	8.13	**15.59**	**14.54**	6.98	**6.96**	10.36	5.36	**6.20**	**8.22**
	Ours	3.74	4.24	7.46	4.17	4.99	5.41	8.15	9.39	5.03	4.88	8.37	4.02	4.27	5.70

perform on point clouds sampled from meshes, while IOU performs on solid voxelization of meshes.

More details on data processing, ablation networks, and training procedures can be found in the Appendix.

4.1. Comparision

We compare our method to state-of-the-art methods, including AtlasNet [20], ImNet [2], OccNet [1], DISN [8], and Ladybird [9]. All methods are trained across all categories. We report two versions of our method. One uses the ground truth camera and is called Ours, and the other uses estimated camera parameters and is called

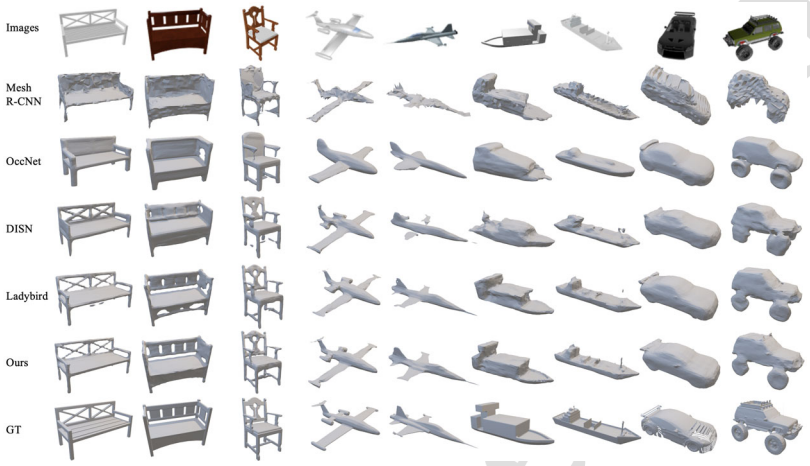


Figure 6: Qualitative Results on the ShapeNet Core dataset.

Ours$_{cam}$.

We report numerical results of the ShapeNet dataset in Table 1. MDISN outperforms other methods on most of the categories and achieves the best average performance. Note that it does not consistently perform best in all categories. Part of the reason is that cross-category training is unstable and sensitive to the diversity and number of samples in individual categories. For large categories, such as Airplane and Chair, the training becomes more stable, and the numerical result of MDISN consistently outperforms others on all the metrics.

To better illustrate the differences between the results generated by the methods, we show some randomly generated shapes in Figure 6. We use the pre-trained models from the Mesh R-CNN, OccNet, and DISN. For Ladybird, we reimplement the network and train it according to their implementation description. In general, all methods can reconstruct the correct 3D objects, and the pixel-aligned methods can better approximate the ground truth shapes, recovering fine-grained details. Although being able to recover geometric details, DISN and Ladybird tend to include more structure noise, and in contrast, our method has less noise and better details.

For real-world images, we present the quantitative evaluation of the Pix3D dataset

Figure 7: Qualitative Results on the Pix3D dataset. Ground truth image masks and camera parameters are used.

Table 2: Quantitative results on the Pix3D dataset.

	IOU(%)↑		**EMD**(x100)↓		**CD**(x1000)↓	
	Ladybird	Ours	Ladybird	Ours	Ladybird	Ours
bed	70.7	**72.4**	2.80	**2.50**	9.84	**7.61**
bookcase	44.3	**48.4**	**2.91**	2.65	10.94	**8.89**
chair	57.3	**59.0**	2.82	**2.53**	14.05	**7.42**
desk	51.2	**60.0**	3.18	**2.72**	18.87	**12.21**
misc	29.8	**45.8**	4.45	**3.63**	36.77	**18.18**
sofa	**86.7**	85.8	**2.02**	**2.02**	**4.56**	4.84
table	56.9	**60.0**	2.96	**2.48**	21.66	**8.97**
tool	41.3	**43.5**	3.70	**3.39**	**7.78**	17.87
wardrobe	87.5	**87.6**	**1.92**	1.96	**4.80**	5.07
mean	58.4	**62.5**	2.97	**2.65**	14.36	**10.12**

in Table 2. Following Ladybird, we use ground truth image masks and camera poses for evaluation. Since there is no official training/test split for the Pix3D dataset, we randomly select 80% of images from the dataset for training and use the remaining images for testing. We train and evaluate Ladybird and MDISN using the same setting. The numerical results show that our method outperforms Ladybird in terms of CD, EMD, and IOU.

In addition to the quantitative results, we also show surface reconstruction results from real-world images in Figure 7. Compared to the synthetic images from the ShapeNet dataset, the real-world images are more diverse in terms of camera views, object sizes, and appearances. The reconstructed shapes from MDISN are much better than Ladybird.

Table 3: Ablation study of our network. Ground truth camera parameters are used for the ablation study.

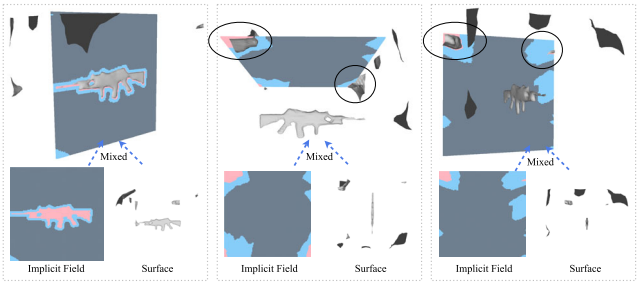
Metrics	Methods	plane	bench	cabinet	car	chair	display	lamp	speaker	rifle	sofa	table	phone	watercraft	mean
IOU(%)↑	Ours$_{no-refine}$	40.3	43.8	57.0	67.5	60.5	55.5	40.7	57.5	31.8	65.2	55.7	51.0	43.9	51.6
	Ours$_{no-deform}$	63.9	51.5	57.9	74.7	63.2	63.0	48.4	61.2	71.8	70.5	56.3	75.3	62.0	63.5
	Ours	68.1	62.5	59.3	80.5	66.4	67.4	54.6	64.2	74.2	74.3	60.0	79.2	68.6	**67.6**
EMD(x100)↓	Ours$_{no-refine}$	14.30	9.31	3.54	10.57	4.41	7.22	8.60	7.01	18.49	7.14	4.16	13.24	14.25	9.40
	Ours$_{no-deform}$	3.12	6.53	2.83	5.26	2.61	3.42	3.77	3.66	2.45	4.11	3.80	3.37	4.55	3.81
	Ours	1.88	1.94	2.62	2.42	2.20	2.12	2.82	2.87	1.64	2.17	2.53	1.85	2.01	**2.24**
CD(x1000)↓	Ours$_{no-refine}$	532.54	297.4	37.09	337.82	87.94	196.00	249.57	162.47	762.36	179.48	64.74	421.25	575.67	300.33
	Ours$_{no-deform}$	49.00	199.14	14.82	93.69	14.30	54.16	24.31	39.00	37.14	79.01	52.66	59.06	96.70	62.54
	Ours	3.74	4.24	7.46	4.17	4.99	5.41	8.15	9.39	5.03	4.88	8.37	4.02	4.27	**5.70**

4.2. Ablation Study

Influence of Core Components. We study two main variants of our network, one using no deformation modules, denoted Ours$_{no-deform}$, and the other considering no coarse-to-fine generation, denoted Ours$_{no-refine}$. In Ours$_{no-deform}$, the output fields from each feature map are accumulated as the final result. In Ours$_{no-refine}$, all feature maps are combined as a union and predict only one field as the final result. See the appendix for details.

From Table 3, we can see that both Ours$_{no-deform}$ and Ours$_{no-refine}$ are unable to work adequately. The multiscale features could fail if having no additional global field as in DISN and Ladybird or multiscale deformed fields as in Ours. To understand this, we show a commonly failed example in Figure 8, where we find that many outliers and noise are created around the 3D object. Those unexpected surface patches lie at the boundaries that increase the Euclidean distance between testing sampling and ground truth samplings, leading to large EMD and CD values. The impact of outlier patches on IOU is smaller since IOU counts spatial occupancy. Thus in Table 3 the IOU makes sense, while EMD and CD do not.

Variants of deformation module. The initial field is generated from a template generator, and deformed by the deformation modules following the changes of consecutive feature maps. The module contains a state vector and a pair of consecutive feature



(A) Left View (B) Bottom View (C) Front View

Figure 8: A failed example generated by the variants of our network that do not use the multiscale deformation modules. As can be seen from various views (A-C), the reconstructed shape contains surface fragments that lie on the boundaries of the canonical 3D shape space. Such results frequently occur from the ablation networks.

Table 4: Ablation study of our deformation module.

Template	Previous Feat.	State	Results		
			IOU	**EMD**	**CD**
			71.4	1.67	2.95
	✓	✓	72.0	1.64	2.58
✓		✓	72.2	1.63	2.35
✓	✓		73.4	**1.59**	1.75
✓	✓	✓	**73.7**	**1.59**	**1.68**

maps. Therefore, we perform the ablation study considering three different factors, the template, the state, and the previous feature map from the feature pairs. We use the category 'Rifle' for the evaluation. From Table 4, we can see that the module performs best when all factors are considered and worst when removing all factors. In particular, the template and the previous feature map are more important than the state vector.

4.3. More Results

While the feature maps dominate the refinement process, the effectiveness of the individual feature maps is uncertain. In the following, we will show the learned templates and the influence of the feature maps.

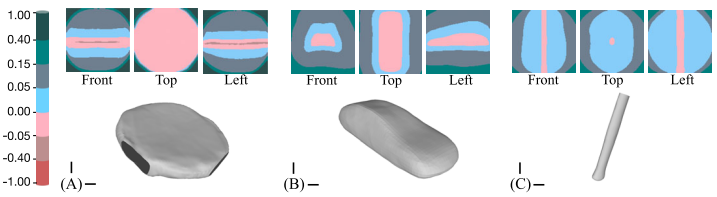


Figure 9: Learned shape templates visualized by implicit fields and 3D surfaces. From (A-C), the templates are learned from all shape categories, the 'Car' category and the 'Lamp' category respectively.

Learned Templates. Three different shape templates, the general template, the 'Car' template, and the 'Lamp' template, are shown in Figure 9. The templates are trained either using all categories (general template) or using a single category ('Car' or 'Lamp' template). From the figures, the 'Car' and 'Lamp' templates have a more intuitive meaning than the general template. In Table 4, using a template field also leads to better performance. Therefore, 3D reconstruction can benefit from a proper initialized implicit field.

Influence of Feature Maps. In Figure 10 we show more results of the multiscale reconstruction. All the objects are from the category 'Rifle' with a similar shape structure. From the figure, we can see that the feature maps of the first two levels cannot provide meaningful information about the shape. Starting from the third level, the global structure of the object becomes notable in feature images and ao as the reconstructed shapes. When given higher resolution features, the reconstructed 3D objects contain

Table 5: The influence of the feature maps. As the resolution increases, the performance improves. At the lowest level, the performance is still reasonable.

Multiscale Feature maps						Results		
l_1	l_2	l_3	l_4	l_5	l_6	**IOU**	**EMD**	**CD**
				✓	✓	60.4	3.41	12.47
			✓	✓	✓	69.2	2.99	7.11
		✓	✓	✓	✓	73.4	2.67	5.58
	✓	✓	✓	✓	✓	79.4	**2.41**	4.24
✓	✓	✓	✓	✓	✓	**80.5**	2.42	**4.17**

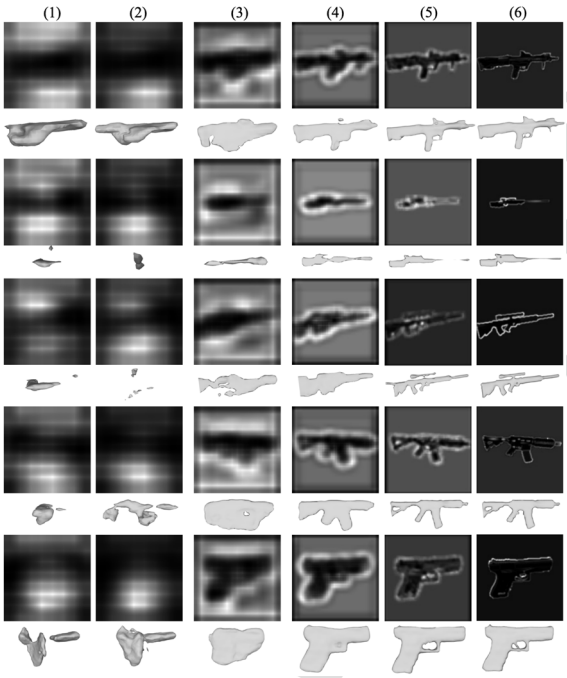


Figure 10: Multiscale Feature-aligned 3D reconstruction. Feature maps (1-6) show different level of shape details, from which we can see (3) is a key level. Before this level there is very little object-level information, e.g. feature maps (1-2) of different objects are very similar. After this level, the outline of the object becomes clearer in both the feature images and the reconstructed shapes.

more surface details and are more aligned with the feature images.

In Table 5 we give qualitative results using different combinations of feature maps. We use the 'Car' category to evaluate multiscale results. The performance continuously improves when the resolution of the feature maps increases from l_1 to l_6. Larger improvement occurs at a lower resolution and becomes minor at higher resolution. This is consistent with the visual results in Figure 10 and Figure 1. More multiscale results from other categories are shown in Figure 11.

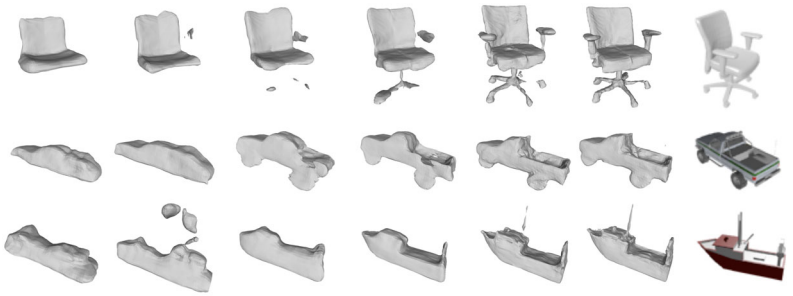


Figure 11: Coarse-to-fine 3D object reconstruction.

4.4. Model Complexity

We show the model size and computation complexity in Table **??**. We feed the networks with a 137x137 image and 2048 points to compute the GLOPs. MDISN has a smaller size but demands more computational resources. Nevertheless, MDISN enables exhausted mining of images features that consequently leads to faster training convergence, as shown in Figure 12.

5. Conclusion and Future Work

We present a multiscale deformed implicit surface network for single-image 3D reconstruction. Our network has a sequence of implicit field deformation modules to adapt the implicit field to the input image iteratively. The deformation module leverage multiscale image feature maps and the change of consecutive feature map for implicit field refinement. Experimental results on synthetic and real-world datasets show that the proposed method performs better than state-of-the-art methods.

Table 6: Model size and complexity.

Model	Model Size (Parameters)	Model Size (MB)	GLOPs (forward pass)
DISN	71.46M	266M	10.22
Ladybird	72.22M	276M	11.76
MDISN	23.5M	90M	24.04

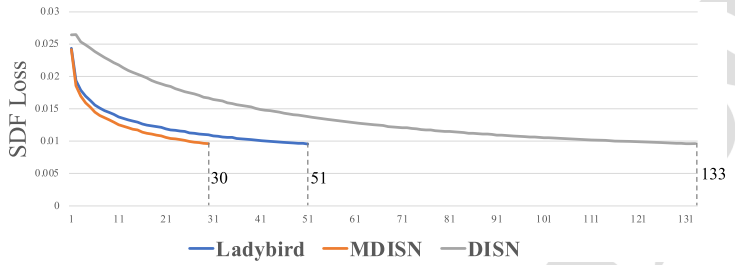


Figure 12: Convergence speeds. The training losses are similar for MDISN, Ladybird, and DISN, at epochs 30, 51, and 133.

Acknowledgements

We thank the anonymous reviewers for their valuable comments. This work was supported in part by National Key R&D Program of China (2018YFB1403901, 2019YFF0302902), NSF China (61902007) and Joint NSFC-ISF Research Grant (62161146002).

References

[1] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, A. Geiger, Occupancy networks: Learning 3D reconstruction in function space, in: Proc. CVPR, 2019.

[2] Z. Chen, H. Zhang, Learning implicit fields for generative shape modeling, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[3] J. J. Park, P. Florence, J. Straub, R. Newcombe, S. Lovegrove, DeepSDF: Learning continuous signed distance functions for shape representation, in: CVPR, 2019.

[4] T. Takikawa, J. Litalien, K. Yin, K. Kreis, C. Loop, D. Nowrouzezahrai, A. Jacobson, M. McGuire, S. Fidler, Neural geometric level of detail: Real-time rendering with implicit 3d shapes, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 11358–11367.

[5] M. Yang, Y. Wen, W. Chen, Y. Chen, K. Jia, Deep optimized priors for 3d shape modeling and reconstruction, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 3269–3278.

[6] Y. Deng, J. Yang, X. Tong, Deformed implicit field: Modeling 3d shapes with learned dense correspondence, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 10286–10296.

[7] Z. Chen, Y. Zhang, K. Genova, S. Fanello, S. Bouaziz, C. Hane, R. Du, C. Keskin, T. Funkhouser, D. Tang, Multiresolution deep implicit functions for 3d shape representation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 13087–13096.

[8] Q. Xu, W. Wang, D. Ceylan, R. Mech, U. Neumann, Disn: Deep implicit surface network for high-quality single-view 3d reconstruction, arXiv preprint arXiv:1905.10711.

[9] Y. Xu, T. Fan, Y. Yuan, G. Singh, Ladybird: Quasi-monte carlo sampling for deep implicit field based 3d reconstruction with symmetry, in: European Conference on Computer Vision, Springer, 2020, pp. 248–263.

[10] S. Saito, Z. Huang, R. Natsume, S. Morishima, A. Kanazawa, H. Li, Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 2304–2314.

[11] A. X. Chang, T. Funkhouser, L. J. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, F. Yu, ShapeNet: An Information-Rich 3D Model Repository (arXiv:1512.03012 [cs.GR]).

[12] X. Sun, J. Wu, X. Zhang, Z. Zhang, C. Zhang, T. Xue, J. B. Tenenbaum, W. T. Freeman, Pix3d: Dataset and methods for single-image 3d shape modeling, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[13] H. Fan, H. Su, L. J. Guibas, A point set generation network for 3d object reconstruction from a single image, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 605–613.

[14] C.-H. Lin, C. Kong, S. Lucey, Learning efficient point cloud generation for dense 3d object reconstruction, in: proceedings of the AAAI Conference on Artificial Intelligence, Vol. 32, 2018.

[15] P. Mandikal, K. Navaneet, M. Agarwal, R. V. Babu, 3d-lmnet: Latent embedding matching for accurate and diverse 3d point cloud reconstruction from a single image, arXiv preprint arXiv:1807.07796.

[16] C. B. Choy, D. Xu, J. Gwak, K. Chen, S. Savarese, 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction, in: European conference on computer vision, Springer, 2016, pp. 628–644.

[17] J. Wu, C. Zhang, X. Zhang, Z. Zhang, W. T. Freeman, J. B. Tenenbaum, Learning shape priors for single-view 3d completion and reconstruction, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 646–662.

[18] H. Xie, H. Yao, X. Sun, S. Zhou, S. Zhang, Pix2vox: Context-aware 3d reconstruction from single and multi-view images, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 2690–2698.

[19] X. Zhang, Z. Zhang, C. Zhang, J. B. Tenenbaum, W. T. Freeman, J. Wu, Learning to reconstruct shapes from unseen classes, arXiv preprint arXiv:1812.11166.

[20] T. Groueix, M. Fisher, V. G. Kim, B. C. Russell, M. Aubry, A papier-mâché approach to learning 3d surface generation, in: Proc. CVPR, 2018, pp. 216–224.

[21] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, Y.-G. Jiang, Pixel2mesh: Generating 3d mesh models from single rgb images, in: ECCV, 2018, pp. 52–67.

[22] W. Wang, D. Ceylan, R. Mech, U. Neumann, 3dn: 3d deformation network, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 1038–1046.

[23] G. Gkioxari, J. Malik, J. Johnson, Mesh r-cnn, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 9785–9795.

[24] J. Pan, X. Han, W. Chen, J. Tang, K. Jia, Deep mesh reconstruction from single rgb images via topology modification networks, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 9964–9973.

[25] Y. Yao, N. Schertler, E. Rosales, H. Rhodin, L. Sigal, A. 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, 2020, pp. 531–540.

[26] C. Niu, J. Li, K. Xu, Im2struct: Recovering 3d shape structure from a single rgb image, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4521–4529.

[27] J. Tang, X. Han, J. Pan, K. Jia, X. Tong, A skeleton-bridged deep learning approach for generating meshes of complex topologies from single rgb images, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 4541–4550.

[28] R. Wu, Y. Zhuang, K. Xu, H. Zhang, B. Chen, Pq-net: A generative part seq2seq network for 3d shapes, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 829–838.

[29] H. Kato, Y. Ushiku, T. Harada, Neural 3d mesh renderer, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 3907–3916.

[30] S. Liu, W. Chen, T. Li, H. Li, Soft rasterizer: Differentiable rendering for unsupervised single-view mesh reconstruction, arXiv preprint arXiv:1901.05567.

[31] L. Liu, J. Gu, K. Zaw Lin, T.-S. Chua, C. Theobalt, Neural sparse voxel fields, Advances in Neural Information Processing Systems 33.

[32] X. Yan, J. Yang, E. Yumer, Y. Guo, H. Lee, Perspective transformer nets: learning single-view 3d object reconstruction without 3d supervision, in: Proceedings of the 30th International Conference on Neural Information Processing Systems, 2016, pp. 1704–1712.

[33] E. Insafutdinov, A. Dosovitskiy, Unsupervised learning of shape and pose with differentiable point clouds, in: Proceedings of the 32nd International Conference on Neural Information Processing Systems, 2018, pp. 2807–2817.

[34] M. Li, H. Zhang, D^2im-net: Learning detail disentangled implicit fields from single images, arXiv preprint arXiv:2012.06650.

[35] M. Niemeyer, L. Mescheder, M. Oechsle, A. Geiger, Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 3504–3515.

[36] S. Liu, Y. Zhang, S. Peng, B. Shi, M. Pollefeys, Z. Cui, Dist: Rendering deep implicit signed distance function with differentiable sphere tracing, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 2019–2028.

[37] Y. Jiang, D. Ji, Z. Han, M. 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, 2020, pp. 1251–1261.

[38] Y. Zhuang, Y. Liu, B. Chen, Neural implicit 3d shapes from single images with spatial patterns, arXiv preprint arXiv:2106.03087.

[39] W. E. Lorensen, H. E. Cline, Marching cubes: A high resolution 3d surface construction algorithm, ACM siggraph computer graphics 21 (4) (1987) 163–169.

[40] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556.

Appendix A. Data Processing

Appendix A.1. Datasets.

The ShapeNet Core dataset [11] includes 13 object categories, and for each object, 24 views are rendered with resolution of 137×137 as in [16]. Pix3D Dataset [12] contains 9 object categories with real-world images and the exact mask images. The number of views and the image resolution varies from different shapes.We process all the shapes and images in the same format for the two datasets. Specifically, all shapes are normalized to the range of [-1,1] and all images are scaled to the resolution of 137×137.

Appendix A.2. 3D Point Sampling.

For each shape, 2048 points are sampled for training. We firstly normalize the shapes to a unified cube with their centers of mass at the origin. Then we uniformly sample 256^3 grid points from the cube and compute th SDF values for all the grid samples. Following the sampling process of Ladybird [9], the 256^3 points are down-sampled with two stages. In the first stage, 32,768 points are randomly sampled from the four SDF ranges [-0.10,-0.03], [-0.03,0.00], [0.00,0.03], and [0.03,0.10], with the same probabilities. In the second stage, 2048 points are uniformly sampled from the 32,768 points using the farthest points sampling strategy.

In testing, 65^3 grid points are sampled are fed to the network, and output the SDF values. The object mesh is extracted as the zero iso-surface of the generated SDF using the Marching Cube algorithm.

Appendix A.3. 3D-to-2D Camera Projection.

The pixel coordinate a of a 3D point sampling p is computed as two stages. Firstly, the point is converted from the world coordinate system to the local camera coordinate system c based on the rigid transformation matrix A^c, such that $p^c = A^c p$. Then in the camera space, point $p^c = (x^c, y^c, z^c)$ is projected to the 2D canvas via perspective transformation, i.e., $\pi(p^c) = (\frac{x^c}{z^c}, \frac{y^c}{z^c})$. The projected pixel whose coordinate lies out of an image will reset to 0 or 136 (the input image resolution is fixed as 137×137 in our experiment).

Appendix B. Network and Training Details

Appendix B.1. Traning Policy.

We implement our method based on the framework of Pytorch. For training on the ShapeNet dataset, we use the Adam optimizer with a learning rate 1e-4, a decay rate of 0.9, and a decay step size of 5 epochs. The network was trained for **20** epochs with batch size 20. For training on the Pix3D dataset, we use the Adam optimizer with a constant learning rate 1e-4, and smaller batch size 5. The network was trained for **50** epochs. For the ShapeNet dataset, at each epoch, we randomly sample a subset of images from each category. Specifically, a maximum number of 36000 images are sampled for each category. The total number of images in an epoch is 411,384 resulting in 20,570 iterations. Our model is trained across all categories.

Appendix B.2. Network Architecture.

We introduce the details of the image encoder m, template generator τ, implicit field deformation module h in our paper.

Appendix B.2.1. Image Encoder.

We use the convolution network of VGG-16 as our image encoder, which generates multi-resolution feature maps. Similar to DISN [8], we reshape the feature maps to the original image size with bilinear interpolation and collect the local image features of a pixel from all scales of feature maps. Specifically, the local feature contains six sub-features from the six feature maps, with the dimension of $\{64, 128, 256, 512, 512, 512\}$ respectively.

Appendix B.2.2. Template Generator.

The template generator maps each point sampling to a SDF value. The point is firstly promoted from R^3 to the dimension of 512 using a multi-layer perceptron (MLP). Then the embedded point feature is passed to another MLP to generate the SDF.

Template Generator g	
Input a 3-dim point $\rightarrow$ output a 1-dim SDF value	
Operation	Output Shape
Conv1D+ReLU	(64,)
Conv1D+ReLU	(256,)
Conv1D+ReLU	(512,)
Conv1D+ReLU	(512,)
Conv1D+ReLU	(256,)
Conv1D	(1,)

Table B.7: Template Generator.

Appendix B.2.3. Deformation Module.

The implit field s is firstly embedded to a 256-dim feature by a 3-layer MLP (64,128,256) before passing to the deformation module. Then the implicit field deformation module combine the feature of s, image feature maps l_n and l_{n+1}, and state vector sv_n into a unified vector and use it to predict a residual field δs and a new state vector sv_{n+1}. Then the residual field is added to the input field to obtain the deformed field, i.e., $s_{n+1} = s_n + \delta s$.

Deformation Module h	
Input a D-dim vector $\rightarrow$ output a 1-dim SDF value and a 256-dim state vector	
Operation	Output Shape
Conv1D+ReLU	(512,)
Conv1D+ReLU	(256,)
Conv1D | Conv1D	(1,) | (256,)
Sum	(1,)

Table B.8: Deformation Module.

Appendix C. Ablation Networks

Two main variants of our network, Ours$_{no-deform}$ and Ours$_{no-refine}$ are shown in Figure C.13.

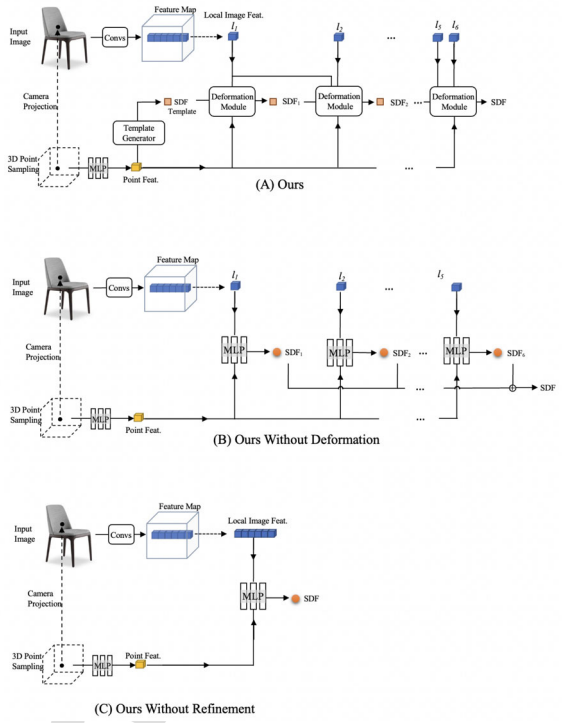


Figure C.13: Method variants, including Ours, Ours$_{no-deform}$ and Ours$_{no-refine}$.

Ethical Approval

This study does not contain any studies with Human or animal subjects performed by any of the authors.

Author Contributions

- Yujie Wang: Methodology, Software, Investigation, Validation, Writing-Original draft preparation.
- Yixin Zhuang: Conceptualization, Software, Visualization, Writing-Original draft preparation.
- Yunzhe Liu: Data Curation, Software, Validation.
- Baoquan Chen: Supervision, Funding acquisition.

Declaration of Interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.