Latent Partition Implicit with Surface Codes for 3D Representation

Chao Chen[1], Yu-Shen Liu[1(✉)], and Zhizhong Han[2]

[1] School of Software, BNRist, Tsinghua University, Beijing, China
`chenchao19@mails.tsinghua.edu.cn`, `liuyushen@tsinghua.edu.cn`
[2] Department of Computer Science, Wayne State University, Detroit, USA
`h312h@wayne.edu`

Abstract. Deep implicit functions have shown remarkable shape modeling ability in various 3D computer vision tasks. One drawback is that it is hard for them to represent a 3D shape as multiple parts. Current solutions learn various primitives and blend the primitives directly in the spatial space, which still struggle to approximate the 3D shape accurately. To resolve this problem, we introduce a novel implicit representation to represent a single 3D shape as a set of parts in the latent space, towards both highly accurate and plausibly interpretable shape modeling. Our insight here is that both the part learning and the part blending can be conducted much easier in the latent space than in the spatial space. We name our method *Latent Partition Implicit (LPI)*, because of its ability of casting the global shape modeling into multiple local part modeling, which partitions the global shape unity. LPI represents a shape as Signed Distance Functions (SDFs) using surface codes. Each surface code is a latent code representing a part whose center is on the surface, which enables us to flexibly employ intrinsic attributes of shapes or additional surface properties. Eventually, LPI can reconstruct both the shape and the parts on the shape, both of which are plausible meshes. LPI is a multi-level representation, which can partition a shape into different numbers of parts after training. LPI can be learned without ground truth signed distances, point normals or any supervision for part partition. LPI outperforms the latest methods under the widely used benchmarks in terms of reconstruction accuracy and modeling interpretability. Our code, data and models are available at https://github.com/chenchao15/LPI.

Keywords: Neural implicit representation · Surface codes · Shape reconstruction
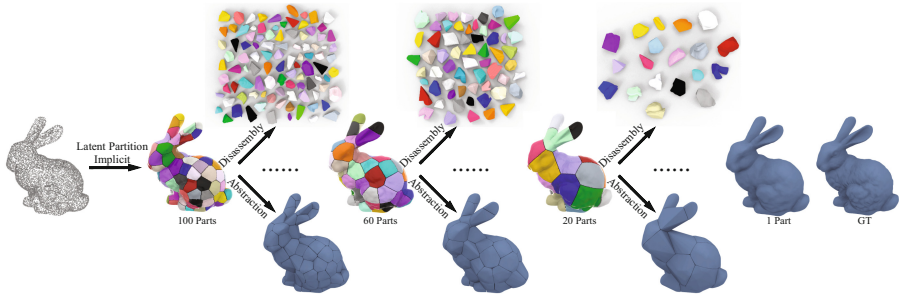
Fig. 1. We introduce *Latent Partition Implicit (LPI)* to represent 3D shapes. We learn LPI from 3D point clouds without requiring ground truth signed distances and point normals. We leverage a surface code to represent a part, and blend all parts with optional surface attributes, such as geodesic distance or segmentation, to reconstruct a surface, both of which are in the latent space. We can represent a shape (mesh) at different levels using different numbers of surface codes, which leads to different numbers of parts (meshes).

1   Introduction

Implicit functions have been a popular representation for 3D objects or scenes. They are able to describe the 3D geometry using the occupancy information tion [11,54] or signed distances [56,66] at arbitrary query locations. We can leverage deep neural networks to learn implicit functions, which we call deep implicit functions. Deep implicit functions usually regard input images or point clouds as conditions to discriminate different object identities in different applications, such as single image reconstruction [12,22,29,43,73,83] or surface reconstruction [22,44,55,89]. Although deep implicit functions have the remarkable ability of geometry modeling, it is difficult for them to decompose shapes into parts. This significantly limits the part-based modeling or editing and makes geometry modeling not interpretable.

Recent methods [16,21,22,82] tried to resolve this problem in 3D spatial space. The key idea behind these methods is to learn to approximate a shape using various primitives, such as convex polytopes [16], 3D Gaussian functions [21,22,82], superquadrics [67] and homeomorphic mappings [69]. However, it is very hard to approximate shapes well by combining primitives together in the spatial space, even a large number of primitives can be used, since both primitive learning and primitive blending in the spatial space are still challenging.

To resolve this issue, we introduce a novel implicit representation to represent a single 3D shape as a set of parts in the latent space, towards both highly accurate and plausibly interpretable shape modeling. Our idea comes from the observation that we can not only learn part geometry but also blend parts together in the latent space, both of which are much easier to be implemented in the latent space than in the spatial space. For a 3D shape, we learn its Signed Distance Functions (SDFs) via blending a set of learnable latent codes in a geometry-aware way. Each latent code represents a part with a center on

the surface, which we call *surface codes*. Our insight of surface codes is that they enable us to flexibly leverage intrinsic attributes of shapes or additional properties of surfaces. Eventually, our learned SDFs can reconstruct not only the shape but also each part on the shape, both of which are plausible meshes. Our method partitions the global shape unity by casting the global shape modeling into multiple local part modeling, as demonstrated in Fig. 1, which achieves important properties such as the reconstruction accuracy and convergence order inherited from the local part modeling. Therefore, we name our method *Latent Partition Implicit (LPI)*. LPI is a multi-level representation for both rigid and non-rigid shapes, which can partition a shape into different numbers of parts after training. LPI can be learned without ground truth signed distances, point normals or any supervision for part partition. Our numerical and visual evaluation shows that LPI outperforms the state-of-the-art methods in terms of reconstruction accuracy and modeling interpretability. Our contributions are listed as follows.

  i) We introduce LPI as a novel implicit representation for 3D shapes. It enables shape modeling and decomposition at the same time, which leads to highly accurate and plausibly interpretable shape modeling.
 ii) We justify the feasibility of learning part geometry and blending parts together in the latent space, which achieves more semantic and efficient parts representation.
iii) Our method significantly outperforms the state-of-the-art methods in terms of reconstruction accuracy and modeling interpretability.

2   Related Work

Deep learning-based 3D shape understanding [26–32,54,58,66,72,85,99] has achieved very promising results in different tasks [10,33,35,36,38,48,57,63,84, 86,87,91].

Learning Global Implicit Functions. Implicit functions have achieved remarkable results in geometry modeling. SDFs or occupancy fields can be learned using ground truth signed distances [7,40,56,66] and binary occupancy labels [11,54]. The learned implicit functions leverage conditions to distinguish shape or scene identities. In different applications, conditions are latent codes obtained from different modalities, such as images for single image reconstruction [12,22,29,43,73,83], learnable latent codes for shape fitting [66], or point clouds for surface reconstruction [17,22,37,44,46,51,52,55,70,89].

With differentiable renderers [3,15,34,38,42,45,47,60,62,64,74,76,77,90,96, 97], we can also learn implicit functions with 2D supervision. This is achieved by minimizing the difference between the images rendered from the learned implicit functions and the ground truth images. Implicit functions achieve great results in geometry and color modeling for complex scenes with neural rendering [58].

Recent methods tried to learn implicit functions from 3D point clouds without ground truth signed distances or binary occupancy labels. Their contributions lied in additional constraints [1,2,4,24,95,98] or the way of leveraging gradients to perceive the surroundings [13,50] to learn signed [1,2,24,50,80,98] or unsigned distance fields [13].

Learning Local Implicit Functions. We can also learn implicit functions in local regions to capture more detailed geometry. The widely used strategy is to split the space occupied by the shape into a voxel grid [8,37,46,53,78,79]. Learnable latent codes located in voxels [8,37,53,78] or vertices of voxels [46,79]. The feature of each query is obtained using bilinear interpolation from these latent codes. Some other methods also used 3D Gaussian functions [21,82] to cover the local regions.
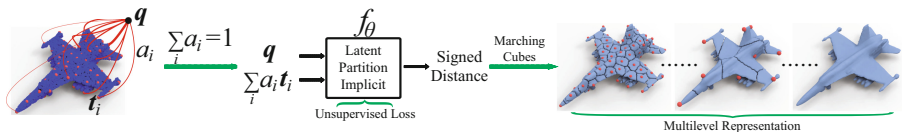

Fig. 2. Demonstration of LPI. We can learn LPI from single point clouds without ground truth signed distances or point normals. LPI can represent shapes using different numbers of parts after training.

The aforementioned methods can approximate the global implicit functions with [7,21,23,40,82,93] or without blending local implicit functions [8,18,37]. Our method shares the idea of approximating global implicit functions by blending local ones, but our novelty lies in that we allow the neural network to adaptively split shapes so that we can blend parts with spatial surface properties or intrinsic attributes in the latent space well.

Shape Decomposition. Decomposing shapes into parts with specific attributes have been extensively studied in computer graphics [19,61,75,92,94]. Recent deep learning based methods tried to resolve this problem by learning primitives using a data-driven strategy [16,20,59,68,69,75,88]. The primitives could be convex polytopes [16,68], 3D Gaussian functions or spheres [22,71]. Instead of primitives learned by these methods, we can learn parts, and blend parts in the latent space rather than spatial space. This results in more accurate approximation and better interpretability for geometry modeling.

3 Method

Problem Statement. We aim to learn LPI as SDFs f_θ for a 3D shape M using a deep neural network parameterized by θ. We can use f_θ to describe multiple shapes with conditions represented as images or point clouds, but we only use

single shapes without conditions to simplify the technical details. Besides this, we also aim to decompose shape M into I parts p_i, each of which is also represented as an SDF, where $i \in [1, I]$. To achieve this, we aim to learn f_θ using the following equation,

$$f_\theta(\boldsymbol{q}, \boldsymbol{a}) = s, \tag{1}$$

where f_θ provides a signed distance s for a query $\boldsymbol{q}$ according to an affinity vector $\boldsymbol{a}$. Each element a_i in $\boldsymbol{a}$ represents its similarity to each part p_i in terms of some metrics, and $\sum_{i \in [1, I]} a_i = 1$. The affinity vector $\boldsymbol{a}$ determines how we can blend local SDFs in the latent space into a global SDF at the query location $\boldsymbol{q}$.

Overview. We demonstrate LPI in Fig. 2. Given a 3D point cloud representing shape M, we first sample I region centers $\boldsymbol{r}_i$ on the point cloud using farthest point sampling (FPS). Each center $\boldsymbol{r}_i$ localizes a part p_i. We represent each part p_i using a latent code $\boldsymbol{t}_i \in \mathbb{R}^{1 \times T}$ in the latent space. The latent codes $\boldsymbol{t}_i$ associated with part centers on the surface are called surface codes. $\boldsymbol{t}_i$ can be used to produce a local SDF to describe the surface of part p_i. We blend all parts together in the latent space by weighting latent codes $\boldsymbol{t}_i$ using an affinity vector $\boldsymbol{a}$ at each query location $\boldsymbol{q}$, which results in a global SDF. We will introduce how to obtain the affinity vector for different kinds of shapes in different applications later. Eventually, we learn the SDFs f_θ which can be further used to reconstruct the surface and decompose the shape into parts at multiple levels.

Surface Codes. It is not new to cast the learning of global SDFs into the learning of multiple local SDFs. The rationale behind this is that local regions are simpler than global shapes, which is easier to learn. A widely used strategy is to regard the space holding 3D shapes as a voxel grid [8,37,78] or a 3D Gaussian mixture model [22,71], and use a latent code to cover each split region, such as voxels or ellipsoids shown in Fig. 3(a) and (b). Different from these methods, we uniformly distribute region centers on the input point cloud, and use latent codes $\boldsymbol{t}_i$ to cover each split region as a Voronoi cell in Fig. 3(c). The benefits we can get are three-folds, which leads our method to be geometry-aware. One benefit is that these centers enable us to perceive the intrinsic attributes of surface, such as geodesic distances, especially for non-rigid shapes, such as humans. This would be helpful to plausibly decompose non-rigid shapes. Another benefit is that Voronoi cells could decompose shapes in a more compact way than voxel grids, especially with the same number of latent codes, which results in more uniformly decomposed parts. This is very helpful to achieve highly accurate shape abstraction without losing too much structure information when abstracting decomposed parts into convex polytopes. Moreover, we can also flexibly leverage additional properties of the surface, such as segmentation, to produce more semantic parts in surface reconstruction. Surface codes $\boldsymbol{t}_i$ are learnable parameters for each shape. They are learned with other network parameters θ.

Blending Regions in the Latent Space. However, the outputs of neighboring local functions are inconsistent. This problem significantly slows down the convergence during training, and also affects the final reconstruction accuracy. Current methods resolve this problem

Fig. 3. Regions covered by voxels and ellipsoids in (a) and (b). Our regions are covered by Voronoi cells in (c). Red points are region centers. (Color figure online)

by blending regions into a global shape in the spatial space [65,82]. They mainly leverage a weight function with a local support to blend the output of multiple local functions, such as local SDFs. This will also bring another issue, that is, the blending requires all neighboring local functions to produce the prediction, which is not affordable when there are lots of local functions. Therefore, DeepLS [8] does not blend local functions but increases the receptive field of each local region to keep region borders consistent.

We resolve this problem by blending local functions in the latent space. Rather than blending outputs in the spatial space, we blend latent codes t_i corresponding to all local SDFs at each query q into a weighted latent code w. w is further concatenated with query q to produce its signed distance. We leverage an affinity vector a at each query q to weight latent codes t_i linearly into w below,

$$w = \sum_{i \in [1,I]} a_i t_i, \tag{2}$$

where a_i represents the affinity between query q and part p_i in terms of some metrics, and we are very flexible to define the affinity metrics, such as Euclidean distance, intrinsic distance, or semantic distance, which achieves different shape decompositions. This flexibility brought by surface codes also differentiates our method with other ones that use bilinear interpolation to blend latent codes on voxel grids [53,79], especially for non-rigid shapes like humans.

Affinity Vector a. One benefit that we can have with surface codes is to obtain geometry awareness. We can encode the geometry by flexibly leveraging different similarity metrics to form the affinity vector a for each query q. We will introduce three metrics
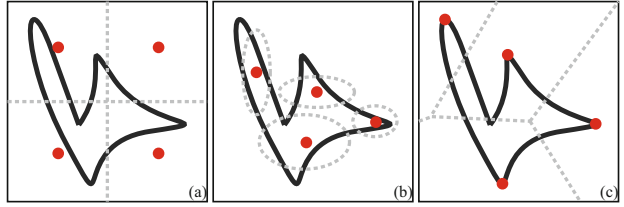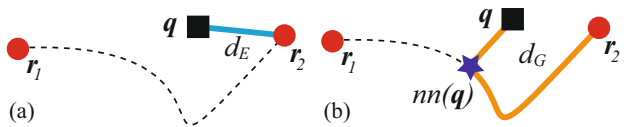

Fig. 4. (a) Euclidean distance d_E between query q and region centers r_i. (b) Intrinsic distance d_G between query q and region centers r_i.

including Euclidean distance d_E, intrinsic distance d_G, and semantic distance d_S to evaluate the distance between query $\boldsymbol{q}$ and each part p_i.

1. Euclidean distance d_E. The most intuitive metric is Euclidean distance, the distance between query $\boldsymbol{q}$ and part p_i can be evaluated using the equation below,

$$d_E(\boldsymbol{q}, p_i) = ||\boldsymbol{q} - \boldsymbol{r}_i||_2, \qquad (3)$$

where $\boldsymbol{r}_i$ is the center of region p_i. This metric indicates that query $\boldsymbol{q}$ is more related to a region p_i if $\boldsymbol{q}$ is nearer to its center $\boldsymbol{r}_i$, as demonstrated in Fig. 4 (a). However, Euclidean distance does not care about the intrinsic property of shapes, especially on non-rigid shapes.

Fig. 5. The effect of Euclidean distance d_E and intrinsic distance d_G

2. Intrinsic distance d_G. To resolve this issue, we also introduce intrinsic distance d_G as a distance metric. The intrinsic distance is formed by Euclidean distance and geodesic distance. The geodesic distance is the distance of the shortest path connecting two points on the surface. So, the intrinsic distance between query $\boldsymbol{q}$ and part p_i can be evaluated using the equation below,

$$d_G(\boldsymbol{q}, p_i) = ||\boldsymbol{q} - nn(\boldsymbol{q}, M)||_2 + G(nn(\boldsymbol{q}, M)), \boldsymbol{r}_i), \qquad (4)$$

where G represents all pair-wise geodesic distances on surface M, $nn(\boldsymbol{q}, M) = \text{argmin}_{\boldsymbol{q}' \in M} ||\boldsymbol{q} - \boldsymbol{q}'||_2$ is the nearest point on M of query $\boldsymbol{q}$. We demonstrate d_G in Fig. 4 (b). We first project query $\boldsymbol{q}$ to surface M by finding the nearest point $nn(\boldsymbol{q}, M)$ of $\boldsymbol{q}$, then we calculate the geodesic distance between $nn(\boldsymbol{q}, M)$ and region center $\boldsymbol{r}_i$. In experiments, we calculate the geodesic distance between each two points on M using heat method [14] in advance.

We highlight the difference between Euclidean distance d_E and intrinsic distance d_G on a non-rigid shape in Fig. 5. Using the same set of six region centers $\boldsymbol{r}_i$, LPI with d_E splits the human body without considering the intrinsic structures in Fig. 5 (a), while LPI with d_G can produce more reasonable splitting using geodesic distances.

For both d_E and d_G, we use the Gaussian function to obtain the similarity a_i between query $\boldsymbol{q}$ and part p_i, and the similarity is normalized by the sum of all similarities to all parts,
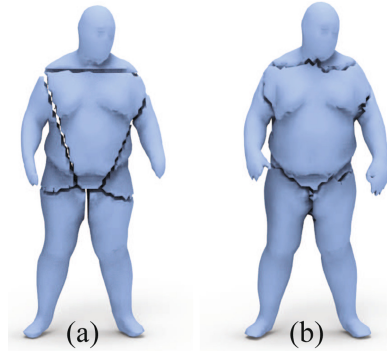
$$a_i = e^{-d(\boldsymbol{q},p_i)/\sigma} / \sum_{i' \in [1,I]} e^{-d(\boldsymbol{q},p_{i'})/\sigma}, \tag{5}$$

where σ is a decay parameter that determines the range in which surface codes get involved, and d could be either d_E or d_G.

3. Semantic distance d_S. With additional properties of the surface, such as segmentation information, we can also obtain the affinity vector using semantic distance d_S. In this case, d_S is an indicator which indicates the segment that query $\boldsymbol{q}$ belongs to. In this way, the affinity vector indicates the semantic similarity between a query and each predefined segment rather than region centers. Assume we were given a sparse point cloud E containing 4 segments in Fig. 6 (a), we find the nearest point $nn(\boldsymbol{q}, E)$ on E for query $\boldsymbol{q}$, and label $\boldsymbol{q}$ using the same segment label of $nn(\boldsymbol{q}, E)$. Then, we set the affinity between $\boldsymbol{q}$ and its segment label to 0.8 while set the affinities to other 3 segments uniformly to keep $\sum_{i \in [1,T]} a_i = 1$. With the segmentation as guidance, LPI can reconstruct the surface of the point cloud in Fig. 2 (a) using 4 semantic parts shown in Fig. 6 (b).

Loss Function. We learn LPI as SDFs $f_\theta(\boldsymbol{q}, \boldsymbol{a})$ without ground truth signed distances or point normals by optimizing network parameters θ and all surface codes $\{\boldsymbol{t}_i\}$. We leverage a loss function that is modified from the pulling loss introduced in [50] to resist the sparseness of input point clouds. The loss function is defined below,
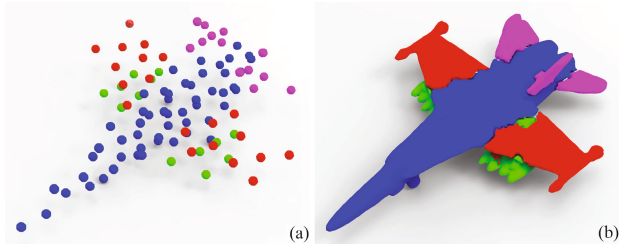

(a)   (b)

Fig. 6. LPI can reconstruct surfaces and semantic parts in (b) according to segmentation.

$$\min_{\theta,\{\boldsymbol{t}_i\}} \left(\sum_{\boldsymbol{x} \in M} \min_{\boldsymbol{y} \in \{\boldsymbol{q}'\}} ||\boldsymbol{x} - \boldsymbol{y}||_2^2 + \sum_{\boldsymbol{y} \in \{\boldsymbol{q}'\}} \min_{\boldsymbol{x} \in M} ||\boldsymbol{x} - \boldsymbol{y}||_2^2 \right), \tag{6}$$

where we obtain the points $\{\boldsymbol{q}'\}$ by projecting each query $\boldsymbol{q}$ using the predicted signed distance $s = f_\theta(\boldsymbol{q}, \boldsymbol{a})$ and the gradient $\nabla f_\theta(\boldsymbol{q}, \boldsymbol{a})$, such that $\boldsymbol{q}' = \boldsymbol{q} - s\nabla f_\theta(\boldsymbol{q}, \boldsymbol{a})/||\nabla f_\theta(\boldsymbol{q}, \boldsymbol{a})||_2$.

Shape and Part Reconstruction. With the learned SDFs f_θ, we reconstruct the shape surface by evaluating f_θ at points on a regular grid and running the marching cubes [49]. To reconstruct each part centered at r_i, we evaluate f_θ at points on the regular grid whose nearest region center is r_i while leveraging an unseen weighted latent code w to predict signed distances at the rest points on the regular grid, and we also leverage marching cubes to reconstruct the surface of each part. After training, we can down sample region centers $\{r_i\}$ to reconstruct the shape with fewer parts in the same way.

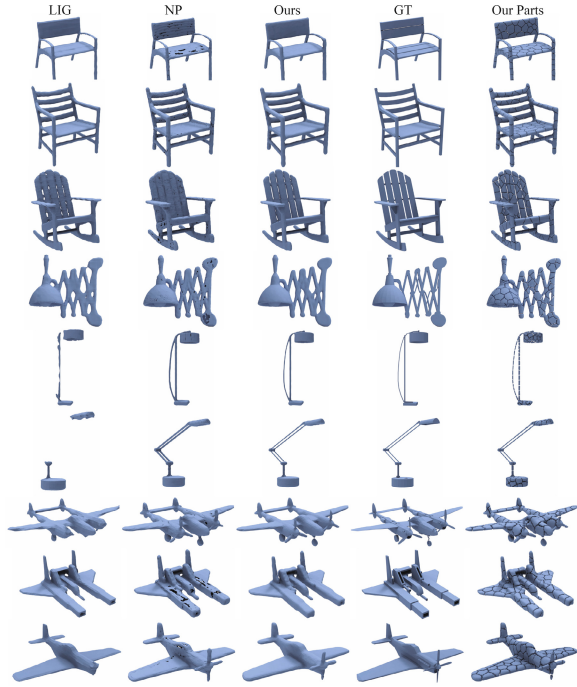**Fig. 7.** Visual comparison with LIG [37] and NP [50] in surface reconstruction under ShapeNet.

4   Experiments

4.1   Setup

Details. For fair comparisons, we leverage the same neural network as NeuralPull (NP) [50] to learn the SDFs f_θ. Moreover, we overfit single shapes in an unsupervised scenario using Eq. (6) without requiring signed distances or point normals, or in a supervised scenario using an MSE loss with signed distance supervision for fair comparisons with others. In an unsupervised scenario, we leverage point clouds as input and sample queries around the point clouds using the same method as NP [50]. In a supervised scenario, we establish a training set containing queries and their ground truth signed distances for each shape. We sample queries and calculate the ground truth signed distances around the shape using the same method as DSDF [66]. All compared methods use the same training set. In both scenarios, we leverage the marching cubes algorithm [49] to reconstruct surfaces using the learned SDFs. We set $\sigma = 1$ in Eq. (5) and the dimension of t_i is $T = 100$.

Table 1. L2CD ($\times 100$) comparison under ShapeNet.

Class	PSR	DMC	BPA	ATLAS	DMC	DSDF	DGP	MeshP	NUD	SALD	NP	Ours
Display	0.273	0.269	0.093	1.094	0.662	0.317	0.293	0.069	0.077	–	0.039	**0.0080**
Lamp	0.227	0.244	0.060	1.988	3.377	0.955	0.167	0.053	0.075	0.071	0.080	**0.0172**
Airplane	0.217	0.171	0.059	1.011	2.205	1.043	0.200	0.049	0.076	0.054	0.008	**0.0060**
Cabinet	0.363	0.373	0.292	1.661	0.766	0.921	0.237	0.112	0.041	–	0.026	**0.0179**
Vessel	0.254	0.228	0.078	0.997	2.487	1.254	0.199	0.061	0.079	–	0.022	**0.0092**
Table	0.383	0.375	0.120	1.311	1.128	0.660	0.333	0.076	0.067	0.066	0.060	**0.0436**
Chair	0.293	0.283	0.099	1.575	1.047	0.483	0.219	0.071	0.063	0.061	0.054	**0.0187**
Sofa	0.276	0.266	0.124	1.307	0.763	0.496	0.174	0.080	0.071	0.058	0.012	**0.0164**
Mean	0.286	0.276	0.116	1.368	1.554	0.766	0.228	0.071	0.069	0.062	0.038	**0.0171**

Datasets. We evaluate our method in three datasets including ShapeNet [9], FAMOUS [17] and D-FAUST [6]. We leverage the same subset of ShapeNet with the same train/test splitting as [44,50]. The subset of ShapeNet contains 8 shape classes. FAMOUS is a dataset released by Points2Surf [17], it contains 22 well-known 3D shapes. D-FAUST is a large-scale dataset containing human meshes. We randomly select 10 shapes and 100 shapes representing different human identities and poses as two sets.

Metrics. We follow MeshingPoint [44] to leverage L2 Chamfer Distance (L2CD), Normal Consistency (NC) [54] and F-score [81] to evaluate our surface reconstruction accuracy under ShapeNet. We sample $100K$ points on the reconstructed and the ground truth surfaces to calculate the L2CD. Following Points2Surf [17], we leverage L2CD to evaluate the reconstruction error between our reconstructed surfaces and the ground truth meshes under FAMOUS dataset. We sample $10K$ points on both reconstructed surfaces and the ground truth meshes to calculate the L2CD. Under D-FAUST dataset, we leverage L1 Chamfer Distance (L1CD), L2CD, NC and IoU to evaluate the reconstruction accuracy. We also sample $10K$ points to calculate CD.

4.2   Surface Reconstruction

ShapeNet. We first evaluate our method in surface reconstruction under ShapeNet. We train our method to reconstruct the surface from each point cloud without using the signed distance supervision. We leverage Euclidean distance d_E to evaluate the distance between a query and each one of $I = 100$ regions to obtain the affinity vector $\boldsymbol{a}$. We use FPS to randomly sample the $I = 100$ region centers on the input point clouds.

We compare our method with the classic surface reconstruction methods and the state-of-the-art data-driven based methods. The compared methods include PSR [39], Ball-Pivoting algorithm (BPA) [5], ATLAS [25], Deep Geometric Prior (DGP) [89], Deep Marching Cube (DMC) [41], DeepSDF (DSDF) [66], MeshP [44], Neural Unsigned Distance (NUD) [13], SALD [2], Local SDF (GRID) [37], IMNET [11] and NeuralPull (NP) [50].

Table 2. Normal consistency comparison under ShapeNet.

Class	PSR	DMC	BPA	ATLAS	DMC	DSDF	MeshP	LIG	IMNET	NP	Ours
Display	0.889	0.842	0.952	0.828	0.882	0.932	0.974	0.926	0.574	0.964	**0.9780**
Lamp	0.876	0.872	0.951	0.593	0.725	0.864	**0.963**	0.882	0.592	0.930	0.9503
Airplane	0.848	0.835	0.926	0.737	0.716	0.872	0.955	0.817	0.550	0.947	**0.9560**
Cabinet	0.880	0.827	0.836	0.682	0.845	0.872	0.957	0.948	0.700	0.930	**0.9576**
Vessel	0.861	0.831	0.917	0.671	0.706	0.841	0.953	0.847	0.574	0.941	**0.9564**
Table	0.833	0.809	0.919	0.783	0.831	0.901	**0.962**	0.936	0.702	0.908	0.9527
Chair	0.850	0.818	0.938	0.638	0.794	0.886	**0.962**	0.920	0.820	0.937	0.9545
Sofa	0.892	0.851	0.940	0.633	0.850	0.906	**0.971**	0.944	0.818	0.951	**0.9713**
Mean	0.866	0.836	0.923	0.695	0.794	0.884	**0.962**	0.903	0.666	0.939	0.9596

Table 3. F-score(μ) comparison under ShapeNet. $\mu = 0.002$.

Class	PSR	DMC	BPA	ATLAS	DMC	DSDF	DGP	MeshP	NUD	LIG	IMNET	NP	Ours
Display	0.468	0.495	0.834	0.071	0.108	0.632	0.417	0.903	0.903	0.551	0.601	0.989	**0.9978**
Lamp	0.455	0.518	0.826	0.029	0.047	0.268	0.405	0.855	0.888	0.624	0.836	0.891	**0.9889**
Airplane	0.415	0.442	0.788	0.070	0.050	0.350	0.249	0.844	0.872	0.564	0.698	0.996	**0.9989**
Cabinet	0.392	0.392	0.553	0.077	0.154	0.573	0.513	0.860	0.950	0.733	0.343	0.980	**0.9849**
Vessel	0.415	0.466	0.789	0.058	0.055	0.323	0.387	0.862	0.883	0.467	0.147	0.985	**0.9955**
Table	0.233	0.287	0.772	0.080	0.095	0.577	0.307	0.880	0.908	0.844	0.425	0.922	**0.9789**
Chair	0.382	0.433	0.802	0.050	0.088	0.447	0.481	0.875	0.913	0.710	0.181	0.954	**0.9897**
Sofa	0.499	0.535	0.786	0.058	0.129	0.577	0.638	0.895	0.945	0.822	0.199	0.968	**0.9946**
Mean	0.407	0.446	0.769	0.062	0.091	0.468	0.425	0.872	0.908	0.664	0.429	0.961	**0.9912**

We report the numerical comparison in terms of L2CD in Table 1, NC in Table 2, F-score with μ in Table 3, F-score with 2μ in Table 4. Our method achieves the best in terms of L2CD and F-score, while obtains the state-of-the-art in terms of NC. We further highlight our advantage in the visual comparison with LIG and NP in Fig. 7. The comparison shows that our method can recon-

Table 4. F-score(2μ) comparison under ShapeNet. $\mu = 0.002$.

Class	PSR	DMC	BPA	ATLAS	DMC	DSDF	DGP	MeshP	NUD	NP	Ours
Display	0.666	0.669	0.929	0.179	0.246	0.787	0.607	0.975	0.944	0.991	**0.9993**
Lamp	0.648	0.681	0.934	0.077	0.113	0.478	0.662	0.951	0.945	0.924	**0.9954**
Airplane	0.619	0.639	0.914	0.179	0.289	0.566	0.515	0.946	0.944	0.997	**0.9998**
Cabinet	0.598	0.591	0.706	0.195	0.128	0.694	0.738	0.946	0.980	0.989	**0.9938**
Vessel	0.633	0.647	0.906	0.153	0.120	0.509	0.648	0.956	0.945	0.990	**0.9985**
Table	0.442	0.462	0.886	0.195	0.221	0.743	0.494	0.963	0.922	0.973	**0.9866**
Chair	0.617	0.615	0.913	0.134	0.345	0.665	0.693	0.964	0.954	0.969	**0.9940**
Sofa	0.725	0.708	0.895	0.153	0.208	0.734	0.834	0.972	0.968	0.974	**0.9982**
Mean	0.618	0.626	0.885	0.158	0.209	0.647	0.649	0.959	0.950	0.976	**0.9957**

struct surfaces with complex geometry in higher accuracy. We also visualize the parts that each surface code covers on the shape, which demonstrates that our method can reconstruct both plausible shapes and parts as meshes.

FAMOUS. We further evaluate our method using non-rigid shapes under FAMOUS dataset. Besides evaluating our surface reconstruction accuracy, we also evaluate our part representation ability. Therefore, we compare our method which also have the ability of representing parts, such as CVX [16] and

Table 5. L2CD, L1CD, and NC comparison under Famous.

Method	L2CD×100	L1CD×100	NC
CVX	0.020	**1.038**	0.906
SIF	0.050	1.600	0.913
Nglod	0.019	1.111	0.934
Ours	**0.018**	**1.038**	**0.950**

SIF [22]. Moreover, we also highlight the advantage of surface codes over the method leveraging voxel grids to cover local regions, such as Nglod [79].

Table 6. L2CD, L1CD, NC and IoU comparison under 10 models of D-FAUST.

Class	L2CD×100	L1CD×100	NC	IoU
NeuralParts	0.008	0.667	0.906	0.695
Ours(Parts)	0.007	0.649	0.886	0.828
Ours(Shape)	**0.004**	**0.555**	**0.952**	**0.837**

For fair comparisons, we train all compared methods to overfit each point cloud in the dataset using the same signed distance supervision. We train our method to regress signed distances by minimizing an MSE loss. We also keep the part number the same in all compared numbers, where we leverage $I = 100$ regions to reconstruct surfaces. We employ 125 latent codes in the 5^3 voxel grid to produce the results of Nglod. We leverage Euclidean distance d_E to calculate the affinity vector to $I = 100$ region centers that are randomly sampled on each point cloud.

We report our numerical comparison in Tab. 5. We achieve the best performance in terms of all metrics. We further highlight our advantage in visual comparison in Fig. 9 (a). We found that our method can reconstruct smoother surfaces with more geometry details than other methods. We also compare the parts that our method reconstructs in Fig. 9 (b). Our parts are more meaningful and expressive than others.
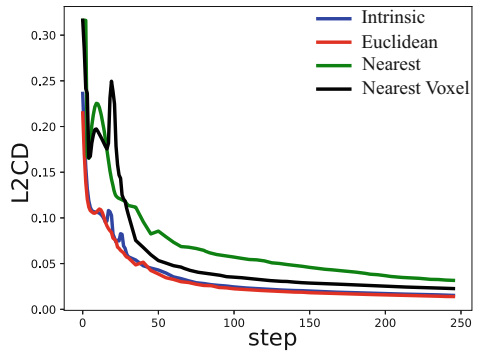

Fig. 8. Effect of surface codes on training.

D-FAUST. Finally, we evaluate our method in surface reconstruction for non-rigid shapes. To evaluate our ability of representing articulated parts, we leverage intrinsic distance d_G to calculate the affinity vector. We compare our method with NeuralParts [69] which is the latest method for representing articulated shapes. Both of our method and NeuralParts represent a human using 6 parts. We annotate $I = 6$ region centers on the input point cloud. For fair comparisons, we leverage the same signed distance supervision to overfit a shape using our method and NeuralParts.

Table 6 reports numerical comparison with NeuralParts. We report our results of global reconstruction and local part approximation, both of which outperform the results of NeuralParts. We further demonstrate our performance in visual comparison in Fig. 10, where our method can reconstruct parts with more geometry details.

Table 7. L2CD, L1CD and NC comparison under 100 models of D-FAUST.

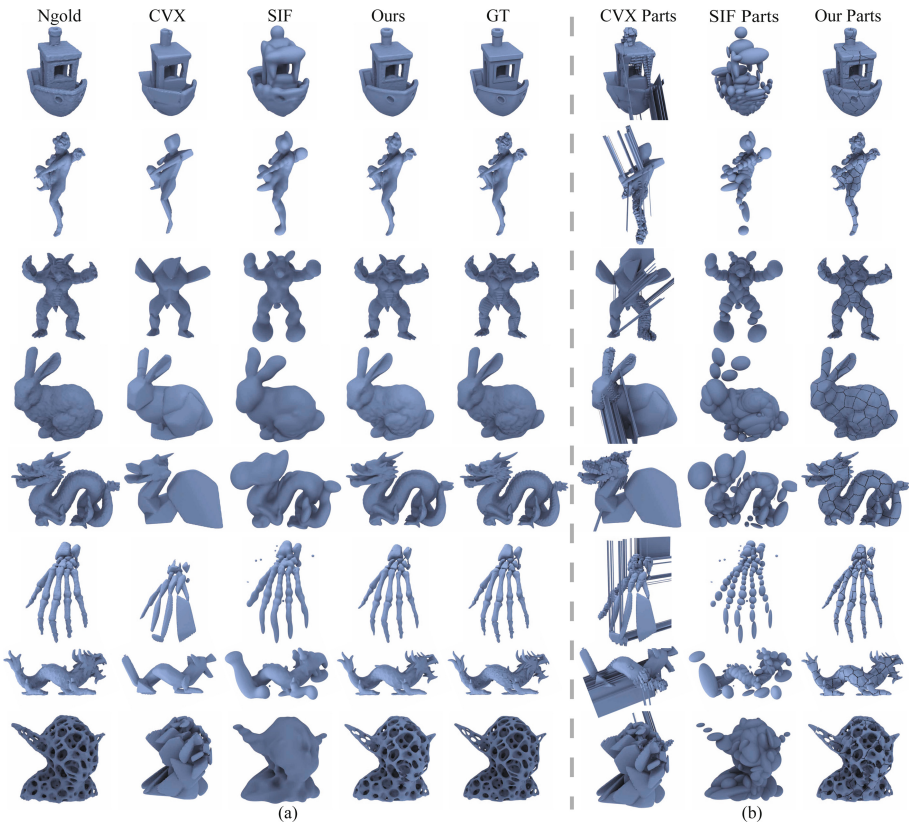| Class	L2CD×100	L1CD×100	NC
DeepLS	0.0065	0.704	0.955
SIF	0.0067	0.681	0.945
Nglod	0.0062	0.718	**0.958**
Ours(Euclidean)	0.0064	0.698	0.955
Ours(Intrinsic)	**0.0059**	**0.611**	0.953

Fig. 9. Visual comparison with Nglod [79], CVX [16], SIF [22] in surface reconstruction under FAMOUS in (a). The part comparison is in (b).

We further highlight the advantages of our surface codes in surface reconstruction for humans. We compare our results obtained with Euclidean distances d_E and intrinsic distances d_G with the methods leveraging latent codes on voxel grids or 3D Gaussian functions, including DeepLS [8], Nglod [79] and SIF [22]. Similarly, we also produce the result of each method using the same set of signed distance supervision. The set contains 100 humans, and we train each method to overfit these humans. We leverage $I = 100$ latent codes to report our results and the results of SIF, while leveraging 125 latent codes to report the results of DeepLS and Nglod.

The numerical comparison is shown in Table 7. Our method achieves the best in terms of CD, and the intrinsic distances d_G work better than Euclidean distances d_E. We visualize our reconstruction in Fig. 11, and our results reveal much smoother surfaces.

4.3   Shape Abstraction

Our method can also represent a shape as an abstraction. We remove the geometry details on a shape by representing each one of its parts as a convex hull.
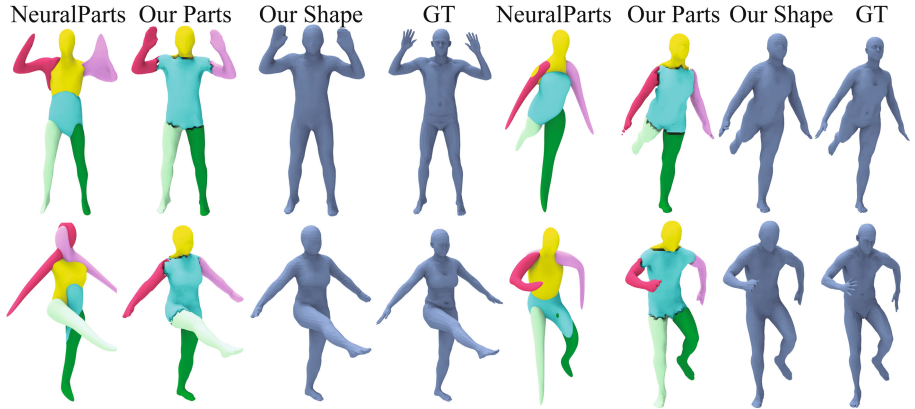

Fig. 10. Visual comparison with NeuralParts [69] under D-FAUST. Color in the same column indicates the same part label.
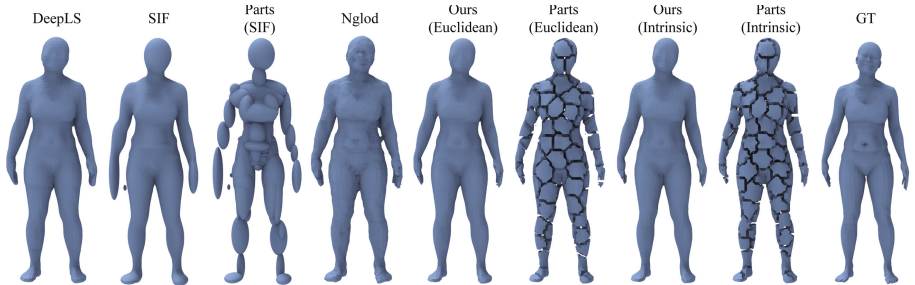

Fig. 11. Visual comparison with DeepLS [8], SIF [22] and Nglod [79] under D-FAUST.

We leverage our learned model in Table 5 to produce the abstractions for shapes in FAMOUS dataset. CVX and SIF also produced results in an overfitting way as ours. Visual comparison in Fig. 13 demonstrates that our method can reveal more complex structures on shapes with the same number of parts.

With semantic distance d_S to obtain affinity vector, our method can also produce shape abstraction with more meaningful parts. Besides the ability of encoding semantic segmentation in Fig. 6, we report our results learned by Eq. (6) with instance segmentation (100 points) in Fig. 12. We first reconstruct the shape and the parts, and abstract the shape using convex hulls of its parts. Due to our ability of using the surface attributes, our method not

Fig. 12. Shape abstraction with instance segmentation. Color in the same row indicates the same part label.

only reconstructs smooth surfaces but also reveals plausible parts. More abstraction results are in our supplemental materials.

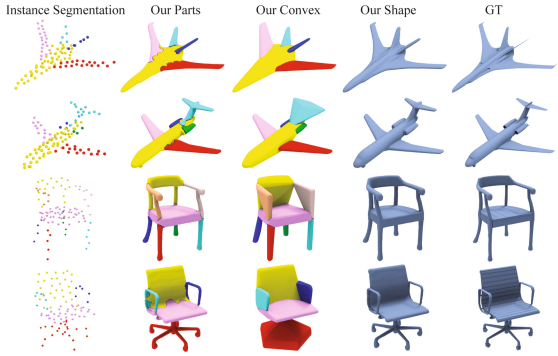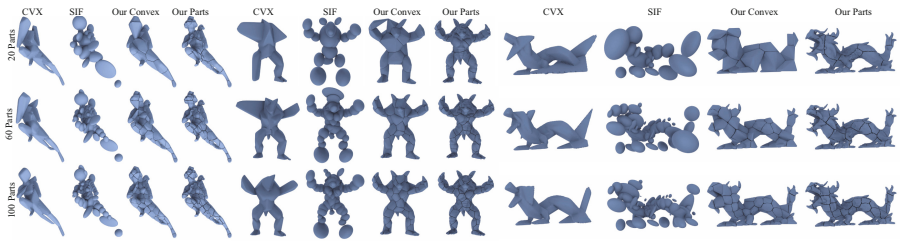Fig. 13. Visual comparison with CVX [16] and SIF [22] in shape abstraction with different numbers of parts.

4.4   Ablation Study

Distance Metrics. We compare the effect of distance metrics on the performance under the 10 humans from D-FAUST. We compare the results with Euclidean distance d_E, intrinsic distance d_G, and no distance encoding. Without distance encoding, we produce the affinity vector as a uniform vector ("Average") or a one-hot vector indicating the nearest surface code ("Nearest"). The numerical comparison in Tab. 8 demonstrates that blending with either Euclidean distances or intrinsic distances achieves better performance and intrinsic distances are the best for non-rigid shape modeling.

Convergence. We compare the effect of surface codes on convergence under the 10 humans from D-FAUST. We compare the average L2CD obtained with Euclidean distance d_E, intrinsic distance d_G, one-hot affinity vector ("Nearest") and latent codes on voxel vertices ("Nearest Voxel"). The comparison in Fig. 8 indicates that blending parts with d_E or

Table 8. Ablation studies under 10 models of D-FAUST.

Affinity	L2CD×100	L1CD×100	NC
Average	0.0058	0.671	**0.952**
Nearest	0.0060	0.671	0.950
Euclidean	0.0057	0.661	**0.952**
Intrinsic	**0.0039**	**0.555**	**0.952**

d_G makes the training converge faster and better than only using the nearest code on both surface or voxel grids.

Surface Code Number. Another advantage of surface codes is that we can represent shapes at multilevel without significant geometry details loss. We reconstruct the Stanford bunny using different numbers of surface codes. Figure 14 indicates that our method still achieves high accuracy even with few surface codes, while CVX [16] requires more parts to approximate the shape well. The reason is that our method represents parts and blends parts in the latent space, which achieves better representation ability.

5   Conclusion

We introduce latent partition implicit to represent 3D shapes. LPI is a multilevel representation, which efficiently represents a shape using different numbers of parts. Our method successfully represents parts using surface codes, and blend parts by weighting surface codes in

Fig. 14. Effect of surface code number.

the latent space to reconstruct surfaces. This leads to highly accurate shape and part modeling. With surface codes, latent partition implicit also enables to flexibly combine additional surface attributes, such as geodesic distance or segmentation. We can learn latent partition implicit from point clouds without requiring ground truth signed distances or point normals. Our method outperforms the state-of-the-art under widely used benchmarks.
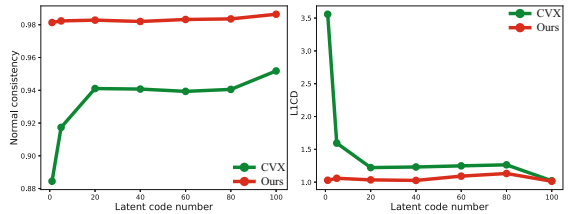
References

1. Atzmon, M., Lipman, Y.: Sal: sign agnostic learning of shapes from raw data. In: IEEE Conference on Computer Vision and Pattern Recognition (2020)
2. Atzmon, M., Lipman, Y.: SALD: sign agnostic learning with derivatives. In: International Conference on Learning Representations (2021)

3. Azinović, D., Martin-Brualla, R., Goldman, D.B., Nießner, M., Thies, J.: Neural rgb-d surface reconstruction. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 6290–6301 (2022)

4. Ben-Shabat, Y., Koneputugodage, C.H., Gould, S.: Digs : divergence guided shape implicit neural representation for unoriented point clouds. CoRR abs/2106.10811 (2021)

5. Bernardini, F., Mittleman, J., Rushmeier, H., Silva, C., Taubin, G.: The ball-pivoting algorithm for surface reconstruction. IEEE Trans. Visual Comput. Graph. **5**(4), 349–359 (1999)

6. Bogo, F., Romero, J., Pons-Moll, G., Black, M.J.: Dynamic FAUST: registering human bodies in motion. In: IEEE Computer Vision and Pattern Recognition (2017)

7. Boulch, A., Marlet, R.: Poco: Point convolution for surface reconstruction. In: IEEE Conference on Computer Vision and Pattern Recognition (2022)

8. Chabra, R., et al.: Deep local shapes: learning local sdf priors for detailed 3d reconstruction. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12374, pp. 608–625. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58526-6_36

9. Chang, A.X., et al.: ShapeNet: an Information-Rich 3D Model Repository. Technical reports. arXiv:1512.03012 [cs.GR], Stanford University – Princeton University – Toyota Technological Institute at Chicago (2015)

10. Chen, C., Han, Z., Liu, Y.S., Zwicker, M.: Unsupervised learning of fine structure generation for 3D point clouds by 2D projection matching. In: IEEE International Conference on Computer Vision (2021)

11. Chen, Z., Zhang, H.: Learning implicit fields for generative shape modeling. In: IEEE Conference on Computer Vision and Pattern Recognition (2019)

12. Chibane, J., Alldieck, T., Pons-Moll, G.: Implicit functions in feature space for 3d shape reconstruction and completion. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 6968–6979 (2020)

13. Chibane, J., Mir, A., Pons-Moll, G.: Neural unsigned distance fields for implicit function learning. arXiv 2010.13938 (2020)

14. Crane, K., Weischedel, C., Wardetzky, M.: The heat method for distance computation. Commun. ACM **60**(11), 90–99 (2017)

15. Darmon, F., Bascle, B., Devaux, J.C., Monasse, P., Aubry, M.: Improving neural implicit surfaces geometry with patch warping. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 6260–6269 (2022)

16. Deng, B., Genova, K., Yazdani, S., Bouaziz, S., Hinton, G.E., Tagliasacchi, A.: Cvxnet: learnable convex decomposition. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 31–41 (2020)

17. Erler, P., Guerrero, P., Ohrhallinger, S., Mitra, N.J., Wimmer, M.: POINTS2SURF learning implicit surfaces from point clouds. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12350, pp. 108–124. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58558-7_7

18. Feng, W., Li, J., Cai, H., Luo, X., Zhang, J.: Neural points: point cloud representation with neural fields for arbitrary upsampling. In: IEEE Conference on Computer Vision and Pattern Recognition (2022)

19. Filoscia, I., Alderighi, T., Giorgi, D., Malomo, L., Callieri, M., Cignoni, P.: Optimizing object decomposition to reduce visual artifacts in 3d printing. Comput. Graph. Forum **39**(2), 423–434 (2020)

20. Gal, R., Bermano, A., Zhang, H., Cohen-Or, D.: MRGAN: multi-rooted 3d shape generation with unsupervised part disentanglement. CoRR abs/2007.12944 (2020)

21. Genova, K., Cole, F., Sud, A., Sarna, A., Funkhouser, T.: Local deep implicit functions for 3d shape. In: IEEE Conference on Computer Vision and Pattern Recognition (2020)
22. Genova, K., Cole, F., Vlasic, D., Sarna, A., Freeman, W.T., Funkhouser, T.: Learning shape templates with structured implicit functions. In: International Conference on Computer Vision (2019)
23. Giebenhain, S., Goldluecke, B.: Air-nets: an attention-based framework for locally conditioned implicit representations. In: 2021 International Conference on 3D Vision. IEEE (2021)
24. Gropp, A., Yariv, L., Haim, N., Atzmon, M., Lipman, Y.: Implicit geometric regularization for learning shapes. arXiv 2002.10099 (2020)
25. Groueix, T., Fisher, M., Kim, V.G., Russell, B.C., Aubry, M.: A papier-mâché approach to learning 3D surface generation. In: IEEE Conference on Computer Vision and Pattern Recognition (2018)
26. Han, Z., Chen, C., Liu, Y.S., Zwicker, M.: Drwr: a differentiable renderer without rendering for unsupervised 3D structure learning from silhouette images. In: International Conference on Machine Learning (2020)
27. Han, Z., Chen, C., Liu, Y.S., Zwicker, M.: ShapeCaptioner: generative caption network for 3D shapes by learning a mapping from parts detected in multiple views to sentences. In: ACM International Conference on Multimedia (2020)
28. Han, Z., et al.: 3D2SeqViews: aggregating sequential views for 3D global feature learning by cnn with hierarchical attention aggregation. IEEE Trans. Image Process. **28**(8), 3986–3999 (2019)
29. Han, Z., Qiao, G., Liu, Y.-S., Zwicker, M.: SeqXY2SeqZ: structure learning for 3d shapes by sequentially predicting 1d occupancy segments from 2d coordinates. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12369, pp. 607–625. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58586-0_36
30. Han, Z., Shang, M., Liu, Y.S., Zwicker, M.: View inter-prediction GAN: unsupervised representation learning for 3D shapes by learning global shape memories to support local view predictions. In: AAAI, pp. 8376–8384 (2019)
31. Han, Z., et al.: SeqViews2SeqLabels: learning 3D global features via aggregating sequential views by rnn with attention. IEEE Trans. Image Process. **28**(2), 672–685 (2019)
32. Han, Z., Shang, M., Wang, X., Liu, Y.S., Zwicker, M.: Y2Seq2Seq: cross-modal representation learning for 3D shape and text by joint reconstruction and prediction of view and word sequences. In: AAAI, pp. 126–133 (2019)
33. Han, Z., Wang, X., Liu, Y.S., Zwicker, M.: Multi-angle point cloud-vae:unsupervised feature learning for 3D point clouds from multiple angles by joint self-reconstruction and half-to-half prediction. In: IEEE International Conference on Computer Vision (2019)
34. Hasselgren, J., Hofmann, N., Munkberg, J.: Shape, light and material decomposition from images using monte carlo rendering and denoising. arXiv abs/2206.03380 (2022)
35. Hu, T., Han, Z., Zwicker, M.: 3D shape completion with multi-view consistent inference. In: AAAI (2020)
36. Jain, A., Mildenhall, B., Barron, J.T., Abbeel, P., Poole, B.: Zero-shot text-guided object generation with dream fields (2022)
37. Jiang, C., Sud, A., Makadia, A., Huang, J., Nießner, M., Funkhouser, T.: Local implicit grid representations for 3D scenes. In: IEEE Conference on Computer Vision and Pattern Recognition (2020)

38. Jiang, Y., Ji, D., Han, Z., Zwicker, M.: SDFDiff: differentiable rendering of signed distance fields for 3D shape optimization. In: IEEE Conference on Computer Vision and Pattern Recognition (2020)

39. Kazhdan, M.M., Hoppe, H.: Screened poisson surface reconstruction. ACM Trans. Graph. **32**(3), 29:1-29:13 (2013)

40. Li, T., Wen, X., Liu, Y.S., Su, H., Han, Z.: Learning deep implicit functions for 3D shapes with dynamic code clouds. In: IEEE Conference on Computer Vision and Pattern Recognition (2022)

41. Liao, Y., Donné, S., Geiger, A.: Deep marching cubes: learning explicit surface representations. In: Conference on Computer Vision and Pattern Recognition (2018)

42. Lin, C.H., Wang, C., Lucey, S.: Sdf-srn: learning signed distance 3d object reconstruction from static images. In: Advances in Neural Information Processing Systems (2020)

43. Littwin, G., Wolf, L.: Deep meta functionals for shape representation. In: IEEE International Conference on Computer Vision (2019)

44. Liu, M., Zhang, X., Su, H.: Meshing point clouds with predicted intrinsic-extrinsic ratio guidance. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12353, pp. 68–84. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58598-3_5

45. Liu, S., Zhang, Y., Peng, S., Shi, B., Pollefeys, M., Cui, Z.: DIST: rendering deep implicit signed distance function with differentiable sphere tracing. In: IEEE Conference on Computer Vision and Pattern Recognition (2020)

46. Liu, S.L., Guo, H.X., Pan, H., Wang, P., Tong, X., Liu, Y.: Deep implicit moving least-squares functions for 3D reconstruction. In: IEEE Conference on Computer Vision and Pattern Recognition (2021)

47. Liu, S., Saito, S., Chen, W., Li, H.: Learning to infer implicit surfaces without 3D supervision. In: Advances in Neural Information Processing Systems (2019)

48. Liu, X., Han, Z., Liu, Y.S., Zwicker, M.: Point2Sequence: learning the shape representation of 3D point clouds with an attention-based sequence to sequence network. In: AAAI, pp. 8778–8785 (2019)

49. Lorensen, W.E., Cline, H.E.: Marching cubes: a high resolution 3D surface construction algorithm. Comput. Graph. **21**(4), 163–169 (1987)

50. Ma, B., Han, Z., Liu, Y.S., Zwicker, M.: Neural-pull: learning signed distance functions from point clouds by learning to pull space onto surfaces. In: International Conference on Machine Learning (2021)

51. Ma, B., Liu, Y.S., Zwicker, M., Han, Z.: Reconstructing surfaces for sparse point clouds with on-surface priors. In: IEEE Conference on Computer Vision and Pattern Recognition (2022)

52. Ma, B., Liu, Y.S., Zwicker, M., Han, Z.: Surface reconstruction from point clouds by learning predictive context priors. In: IEEE Conference on Computer Vision and Pattern Recognition (2022)

53. Martel, J.N.P., Lindell, D.B., Lin, C.Z., Chan, E.R., Monteiro, M., Wetzstein, G.: ACORN: adaptive coordinate networks for neural scene representation. CoRR abs/2105.02788 (2021)

54. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: Learning 3D reconstruction in function space. In: IEEE Conference on Computer Vision and Pattern Recognition (2019)

55. Mi, Z., Luo, Y., Tao, W.: Ssrnet: scalable 3D surface reconstruction network. In: IEEE Conference on Computer Vision and Pattern Recognition (2020)

56. Michalkiewicz, M., Pontes, J.K., Jack, D., Baktashmotlagh, M., Eriksson, A.P.: Deep level sets: implicit surface representations for 3D shape inference. CoRR abs/1901.06802 (2019)
57. Michel, O., Bar-On, R., Liu, R., Benaim, S., Hanocka, R.: Text2mesh: text-driven neural stylization for meshes. In: IEEE Conference on Computer Vision and Pattern Recognition (2022)
58. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: NeRF: representing scenes as neural radiance fields for view synthesis. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12346, pp. 405–421. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58452-8_24
59. Yavartanoo, M., Chung, J., Neshatavar, R., Lee, K.M.: 3dias: 3d shape reconstruction with implicit algebraic surfaces. In: International Conference on Computer Vision (2021)
60. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. arXiv:2201.05989 (2022)
61. Muntoni, A., Livesu, M., Scateni, R., Sheffer, A., Panozzo, D.: Axis-aligned height-field block decomposition of 3d shapes. ACM Trans. Graph. **37**(5), 169:1-169:15 (2018)
62. Niemeyer, M., Mescheder, L., Oechsle, M., Geiger, A.: Differentiable volumetric rendering: Learning implicit 3D representations without 3D supervision. In: IEEE Conference on Computer Vision and Pattern Recognition (2020)
63. Novotny, D., et al.: Keytr: keypoint transporter for 3d reconstruction of deformable objects in videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5595–5604 (2022)
64. Oechsle, M., Peng, S., Geiger, A.: UNISURF: unifying neural implicit surfaces and radiance fields for multi-view reconstruction. CoRR abs/2104.10078 (2021)
65. Ohtake, Y., Belyaev, A.G., Alexa, M., Turk, G., Seidel, H.: Multi-level partition of unity implicits. ACM Trans. Graph. **22**(3), 463–470 (2003)
66. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: DeepSDF: learning continuous signed distance functions for shape representation. In: IEEE Conference on Computer Vision and Pattern Recognition (2019)
67. Paschalidou, D., van Gool, L., Geiger, A.: Learning unsupervised hierarchical part decomposition of 3d objects from a single rgb image. In: Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
68. Paschalidou, D., Gool, L.V., Geiger, A.: Learning unsupervised hierarchical part decomposition of 3d objects from a single RGB image. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1057–1067 (2020)
69. Paschalidou, D., Katharopoulos, A., Geiger, A., Fidler, S.: Neural parts: learning expressive 3d shape abstractions with invertible neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 3204–3215 (2021)
70. Peng, S., Jiang, C.M., Liao, Y., Niemeyer, M., Pollefeys, M., Geiger, A.: Shape as points: a differentiable poisson solver. In: Advances in Neural Information Processing Systems (2021)
71. Rebain, D., Li, K., Sitzmann, V., Yazdani, S., Yi, K.M., Tagliasacchi, A.: Deep medial fields. CoRR abs/2106.03804 (2021)
72. Rückert, D., Franke, L., Stamminger, M.: Adop: approximate differentiable one-pixel point rendering. arXiv:2110.06635 (2021)
73. Saito, S., Huang, Z., Natsume, R., Morishima, S., Kanazawa, A., Li, H.: PIFu: pixel-aligned implicit function for high-resolution clothed human digitization (2019)

74. Fridovich-Keil, S., Yu, A., Tancik, M., Chen, Q., Recht, B., Kanazawa, A.: Plenoxels: radiance fields without neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition (2022)

75. Sharma, G., et al.: Surfit: learning to fit surfaces improves few shot learning on point clouds. CoRR abs/2112.13942 (2021)

76. Sitzmann, V., Martel, J.N., Bergman, A.W., Lindell, D.B., Wetzstein, G.: Implicit neural representations with periodic activation functions. In: Advances in Neural Information Processing Systems (2020)

77. Sitzmann, V., Zollhöfer, M., Wetzstein, G.: Scene representation networks: continuous 3D-structure-aware neural scene representations. In: Advances in Neural Information Processing Systems (2019)

78. Peng, S., Niemeyer, M., Mescheder, L., Pollefeys, M., Geiger, A.: Convolutional occupancy networks. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12348, pp. 523–540. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58580-8_31

79. Takikawa, T., et al.: Neural geometric level of detail: Real-time rendering with implicit 3D shapes. In: IEEE Conference on Computer Vision and Pattern Recognition (2021)

80. Tang, J., Lei, J., Xu, D., Ma, F., Jia, K., Zhang, L.: Sa-convonet: sign-agnostic optimization of convolutional occupancy networks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2021)

81. Tatarchenko, M., Richter, S.R., Ranftl, R., Li, Z., Koltun, V., Brox, T.: What do single-view 3D reconstruction networks learn? In: The IEEE Conference on Computer Vision and Pattern Recognition (2019)

82. Tretschk, E., Tewari, A., Golyanik, V., Zollhöfer, M., Stoll, C., Theobalt, C.: PatchNets: patch-based generalizable deep implicit 3d shape representations. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12361, pp. 293–309. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58517-4_18

83. Wang, W., Xu, Q., Ceylan, D., Mech, R., Neumann, U.: DISN: deep implicit surface network for high-quality single-view 3D reconstruction. In: Advances In Neural Information Processing Systems (2019)

84. Wen, X., Han, Z., Cao, Y.P., Wan, P., Zheng, W., Liu, Y.S.: Cycle4completion: unpaired point cloud completion using cycle transformation with missing region coding. In: IEEE Conference on Computer Vision and Pattern Recognition (2021)

85. Wen, X., Li, T., Han, Z., Liu, Y.S.: Point cloud completion by skip-attention network with hierarchical folding. In: IEEE Conference on Computer Vision and Pattern Recognition (2020)

86. Wen, X., et al.: Pmp-net: point cloud completion by learning multi-step point moving paths. In: IEEE Conference on Computer Vision and Pattern Recognition (2021)

87. Wen, X., Zhou, J., Liu, Y.S., Su, H., Dong, Z., Han, Z.: 3D shape reconstruction from 2D images with disentangled attribute flow. In: IEEE Conference on Computer Vision and Pattern Recognition (2022)

88. Williams, F., Parent-Lévesque, J., Nowrouzezahrai, D., Panozzo, D., Yi, K.M., Tagliasacchi, A.: Voronoinet : general functional approximators with local support. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 1069–1073 (2020)

89. Williams, F., Schneider, T., Silva, C., Zorin, D., Bruna, J., Panozzo, D.: Deep geometric prior for surface reconstruction. In: IEEE Conference on Computer Vision and Pattern Recognition (2019)

90. Wu, Y., Sun, Z.: DFR: differentiable function rendering for learning 3D generation from images. Comput. Graph. Forum **39**(5), 241–252 (2020)
91. Xiang, P., et al.: Snowflakenet: point cloud completion by snowflake point deconvolution with skip-transformer. In: IEEE International Conference on Computer Vision (2021)
92. Yao, C.H., Hung, W.C., Jampani, V., Yang, M.H.: Discovering 3d parts from image collections. In: IEEE International Conference on Computer Vision (2021)
93. Yao, S., Yang, F., Cheng, Y., Mozerov, M.G.: 3d shapes local geometry codes learning with sdf. In: IEEE International Conference on Computer Vision Workshops, pp. 2110–2117 (2021)
94. Yavartanoo, M., Chung, J., Neshatavar, R., Lee, K.M.: 3dias: 3d shape reconstruction with implicit algebraic surfaces. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 12446–12455 (2021)
95. Yifan, W., Wu, S., Oztireli, C., Sorkine-Hornung, O.: Iso-points: optimizing neural implicit surfaces with hybrid representations. CoRR abs/2012.06434 (2020)
96. Yu, Z., Peng, S., Niemeyer, M., Sattler, T., Geiger, A.: Monosdf: exploring monocular geometric cues for neural implicit surface reconstruction. arXiv abs/2022.00665 (2022)
97. Zakharov, S., Kehl, W., Bhargava, A., Gaidon, A.: Autolabeling 3D objects with differentiable rendering of sdf shape priors. In: IEEE Conference on Computer Vision and Pattern Recognition (2020)
98. Zhao, W., Lei, J., Wen, Y., Zhang, J., Jia, K.: Sign-agnostic implicit learning of surface self-similarities for shape modeling and reconstruction from raw point clouds. CoRR abs/2012.07498 (2020)
99. Zhu, Z., Peng, S., et al.: Nice-slam: neural implicit scalable encoding for slam. In: IEEE Conference on Computer Vision and Pattern Recognition (2022)