

Reconstructing Surfaces for Sparse Point Clouds with On-Surface Priors

Baorui Ma

School of Software, BNRIst, Tsinghua University

Beijing 100084, P. R. China

mbr18@mails.tsinghua.edu.cn

Yu-Shen Liu*

School of Software, BNRIst, Tsinghua University

Beijing 100084, P. R. China

liuyushen@tsinghua.edu.cn

Zhizhong Han

Department of Computer Science, Wayne State University

Detroit 48201, USA

h312h@wayne.edu

Abstract

It is an important task to reconstruct surfaces from 3D point clouds. Current methods are able to reconstruct surfaces by learning Signed Distance Functions (SDFs) from single point clouds without ground truth signed distances or point normals. However, they require the point clouds to be dense, which dramatically limits their performance in real applications. To resolve this issue, we propose to reconstruct highly accurate surfaces from sparse point clouds with an on-surface prior. We train a neural network to learn SDFs via projecting queries onto the surface represented by the sparse point cloud. Our key idea is to infer signed distances by pushing both the query projections to be on the surface and the projection distance to be the minimum. To achieve this, we train a neural network to capture the on-surface prior to determine whether a point is on a sparse point cloud or not, and then leverage it as a differentiable function to learn SDFs from unseen sparse point cloud. Our method can learn SDFs from a single sparse point cloud without ground truth signed distances or point normals. Our numerical evaluation under widely used benchmarks demonstrates that our method achieves state-of-the-art reconstruction accuracy, especially for sparse point clouds.

1. Introduction

Reconstructing surfaces from 3D point clouds is a vital task in 3D computer vision. It bridges the gap between the data capturing and the surface editing for various downstream applications. It has been studied for decades using geometric approaches [4, 13, 18, 23]. However, these methods require extensive human interaction to set proper parameters for different 3D point clouds, which leads to poor generalization ability. Therefore, the data-driven strategy becomes more promising to resolve this problem.

Recent learning based methods [7, 12, 16, 20, 29] leverage this strategy to learn signed distance functions (SDFs) from 3D point clouds, and further leverage the learned SDFs to reconstruct surfaces using the marching cubes algorithm [18]. One kind of these methods [7, 12, 16] requires supervision including ground truth signed distances or point normals during training, and infers SDFs for unseen 3D point clouds during test. To remove the requirement of the ground truth supervision, another kind of methods [1, 8, 20] can directly learn SDFs from single unseen 3D point cloud with geometric constraints [1, 8] or neural pulling [20]. One key factor that makes these methods successful without the ground truth supervision is that the single point cloud should be dense, which supports to estimate the zero level set [8] or search accurate pulling targets [20]. However, due to the high cost of dense point clouds capturing, the assumption of dense point clouds fails in real applications. Therefore, it is appealing but challenging to learn SDFs from sparse point clouds without ground truth

*The corresponding author is Yu-Shen Liu.

signed distances or normals.

To resolve this issue, we introduce to learn SDFs from single sparse point clouds with an on-surface prior. For a surface represented by a sparse point cloud, we aim to perceive its surrounding signed distance field via projecting an arbitrary query location onto the surface. Our novelty lies in the two constraints that we add on the projections, so that each projection locates on the surface and is the nearest to the query. This leads to two losses to train a neural network to learn SDFs. One loss is provided by the on-surface prior, which determines whether a projection is on the surface represented by the sparse point cloud or not, even if the projection is not a point of the sparse point cloud. While the other loss encourages the projection distance is the minimum to the surface. To achieve this, we train a neural network using a data-driven strategy to capture the on-surface prior from a dataset during training, and leverage the trained network as a differentiable function to learn SDFs for unseen sparse point clouds. For the learning of SDFs, our method does not require ground truth signed distances or point normals, and enables highly accurate surface reconstruction from sparse point clouds. We show our superior performance over the state-of-the-art methods by numerical and visual comparison under the widely used benchmarks. Our contributions are listed below.

- i) We propose a method to learn SDFs from sparse point clouds without ground truth signed distances or point normals.
- ii) We introduce an on-surface prior which can determine the relationship between a point and a sparse point cloud, and further be used to train another network to learn SDFs.
- iii) Our method significantly outperforms the state-of-the-art methods in terms of surface reconstruction accuracy under large-scale benchmarks.

2. Related Work

Reconstructing surfaces from 3D point clouds is a classic research topic. Geometry based methods [4, 13, 18, 23] tried to resolve this problem by analysing the geometry on the shape itself without learning experience from large scale dataset.

Recent learning based methods [5, 7, 11, 12, 16, 17, 22, 29] achieve state-of-the-art results by learning various priors from dataset using deep learning models. Implicit functions, such as SDFs or occupancy fields, are usually learned to represent 3D shapes or scenes, and then the marching cubes algorithm is used to reconstruct the learned implicit functions into surfaces. Some methods [7, 16] require ground truth signed distances or point normals to learn global prior during training. Some other methods [11, 21, 29, 30] learn

occupancy fields as a global prior using the ground truth occupancy supervision. To reveal more detailed geometry, local shape priors are learned as SDFs [5, 12, 17, 32] or occupancy fields [22] with supervision, where point clouds are usually split into different grids [12] or patches [32] as local regions. Moreover, some more interesting methods for surface reconstruction are also proposed, such as a differentiable formulation of poisson solver [24], retrieving parts [28], iso-points [34] or implicit moving least-squares surfaces [16].

Using meshing strategy, surfaces can also be reconstructed by connecting neighboring points using intrinsic-extrinsic metrics [15], Delaunay triangulation of point clouds [19] or connection from an initial meshes [10]. Local regions represented as point clouds can also be reconstructed via fitting using the Wasserstein distance as a measure of approximation [33].

More appealing solutions are to learn SDFs without ground truth signed distances or point normals. Some methods were proposed to achieve this using geometric constraints [1–3, 8, 31, 35] or through neural pulling [20]. However, these methods are limited by the assumption that point clouds are dense points, which makes them not perform well in real applications. Our method falls in this category, but differently, we can learn accurate SDFs from sparse point clouds.

3. Method

Problem Statement. Given a sparse point cloud $G \in \mathbb{R}^{G \times 3}$, we aim to reconstruct its surface. We achieve this by learning SDFs f_θ from G without requiring ground truth signed distances and normals of points on G . f_θ predicts signed distances $s = f_\theta(q, c)$ for arbitrary queries q sampled around G , where c is a condition identifying G . Since our method can learn SDFs f_θ from single point clouds, we will ignore the condition c in the following. With the learned f_θ , we reconstruct the surface of G using the marching cubes algorithm [18].

Overview. Our method is demonstrated in a 2D case in Fig. 1. It is mainly formed by two functions, i.e., a SDF f_θ and an on-surface decision function (ODF) f_ϕ , both of which are learned by deep neural networks parameterized by θ and ϕ , respectively. The SDF f_θ learns the signed distance field around G , with the on-surface prior provided by the ODF f_ϕ . Therefore, the parameters θ in f_θ are learned for single sparse point clouds with fixed parameters ϕ in f_ϕ during test, without the ground truth signed distances or point normals, while we learn ϕ separately during training using a data-driven strategy.

We start from a query q around the sparse point cloud G in Fig. 1 (a). We project q towards G into a projection p in Fig. 1 (b), using the path determined by the signed distance s and the gradient at q from the SDF f_θ . Then, we establish



Figure 1. The demonstration of our method during test. We leverage a data-driven strategy to learn on-surface decision function f_ϕ as an on-surface prior during training. During testing, we learn SDFs f_θ from a sparse point cloud G . (a) Query q is sampled around G . (b) q is projected into a projection p towards G using the path determined by SDFs f_θ . f_ϕ evaluates whether p is on the surface represented by the K nearest neighbors t of p in (c). We run the marching cubes to reconstruct surfaces of f_θ .

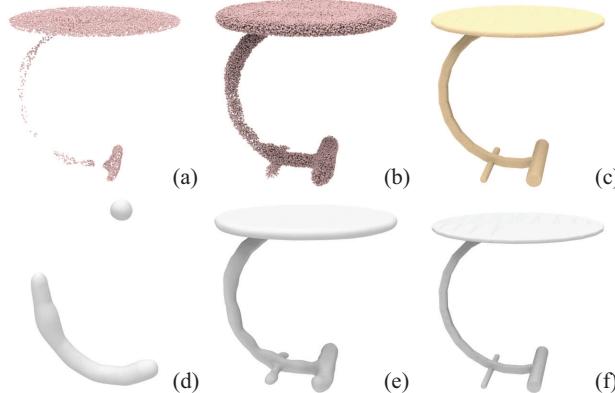


Figure 2. The visual comparison with binary classifier in (a) and unsigned distance function in (b) as the on-surface prior, and the visual comparison with (e) and without (d) geometric regularization. The ground truth shape and sampled points are shown in (f) and (c), respectively.

a local region t on G which is formed by the K nearest neighbors of the projection p in Fig. 1 (c). Finally, the ODF f_ϕ will determine whether the projection p is on the region t or not.

To learn f_θ , we penalize f_θ through the differentiable function f_ϕ , if f_ϕ determines that p is not on the region t , and meanwhile, encourage f_θ to produce the shortest path for the projection.

Query Projection. We project queries q onto sparse point cloud G as an evaluation of f_θ . If the projection path provided by f_θ is correct, there would be no on-surface penalty on f_θ , and vice versa. The projection path for query q can be formed by the signed distance $s = f_\theta(q)$ and the gradient $\nabla f_\theta(q)$. This is also similar to the pulling procedure used in NeuralPull (NP) [20]. The reasonable is that the absolute value of s determines the distance from query q to the surface, and the normalized gradient $d = \nabla f_\theta(q)/\|\nabla f_\theta(q)\|_2$ indicates the direction. Therefore, we can leverage the following equation to project query q to its projection p onto the surface that the zero-level set of f_θ indicates,

$$p = q - sd. \quad (1)$$

On-Surface Prior. The first constraint that we add on the projection p is that p should be located on the surface represented by the sparse point cloud G . This is a difficult problem and the sparseness makes this problem even more

difficult. An intuitive solution is first to establish a local patch t neighboring to p , and fit a quadric surface on t like MPU [23], and finally calculate the distance between p and the quadric surface to make a decision. However, fitting a quadric surface to a point cloud is still challenging, due to the sensitivity to parameter setting and geometry complexity, especially on sparse point clouds.

To resolve this issue, we leverage a data-driven strategy to learn a ODF $f_\phi(p, t)$ using a deep neural network to determine whether the projection p is located on the surface of t or not. We expect $f_\phi(p, t)$ to be class-agnostic and object-agnostic, so we regard t as a local patch rather than a global shape.

Our first trying is to learn $f_\phi(p, t)$ as a binary classifier. The output of f_ϕ indicates the probability of being on the surface. We prepare a training set $T = \{\{p_i, t_i, l_i\}, i \in [1, I]\}$ with ground truth labels l_i indicating whether a point p_i is on a specific point cloud region t_i or not. We leverage the public available benchmarks to obtain the dataset T . We sample p_i around each shape, and sample sparse points on the shape. Among the sampled sparse points, we regard the K nearest neighbors of p_i as t_i . We also label p_i to indicate whether it is sampled from the same surface as the region t_i or not.

However, our preliminary results show that it is very hard to learn a good f_ϕ . As demonstrated in Fig. 2, we learn f_ϕ using one set of points p and evaluate it using another set of points p , where both sets are sampled around the same shape in Fig. 2 (c). We show points that f_ϕ correctly classifies as on-surface points in Fig. 2 (a). The poor result demonstrates that we can not leverage f_ϕ as a binary classifier to capture the on-surface prior.

We further resolve this issue by learning f_ϕ as unsigned distance functions. Using l_i in the training set T as continuous unsigned distances rather than discrete binary labels, we capture a more robust on-surface prior using f_ϕ by minimizing the following loss function,

$$\min_\phi \frac{1}{I} \sum_{i \in [1, I]} \|f_\phi(p_i, t_i) - l_i\|^2. \quad (2)$$

We evaluate the on-surface prior learned with Eq. (2) in Fig. 2 (b). We leverage the same set of points used in Fig. 2 (a) in the evaluation. We use a small unsigned distance threshold to filter out the points that are regarded as

on-surface points by f_ϕ . The smooth surface shown by the on-surface points in Fig. 2 (b) demonstrates that f_ϕ is an eligible on-surface prior.

We obtain poor results by learning f_ϕ as signed distance functions. This is because the sign information among different regions t_i is very complex, which makes it hard to learn the on-surface prior. We will compare this option in experiments later.

Geometric Regularization. One remaining question is that, is the on-surface prior adequate to learn SDFs f_θ for unseen sparse point clouds \mathbf{G} without ground truth signed distances and point normals? To justify this, we learn f_θ by pushing all query projections to arrive on the surface according to the on surface evaluation below, where the learned on-surface prior is represented by the fixed parameters ϕ in ODF f_ϕ ,

$$\min_{\theta} \frac{1}{|Q|} \sum_{q \in Q} |f_\phi(q - sd, Knn(q - sd))|, \quad (3)$$

where Q is a set of queries sampled around the sparse point cloud \mathbf{G} , $|Q|$ is the query number, $Knn(q)$ are K nearest points on \mathbf{G} of q .

We reconstruct surfaces described by the learned f_θ using the marching cubes algorithm. The poor surface in Fig. 2 (d) demonstrates that f_θ can not learn a correct signed distance field. The reason is that the first constraint defined in Eq. (2) only constrains the query projections to be on the surface, while it does not care how the projection path provided by f_θ should be. This results in an inaccurate or even wrong signed distance field.

We resolve this issue by introducing another constraint as a geometric regularization. The geometric regularization encourages the projection path to be the shortest, which matches the definition of signed distances, as defined below,

$$\min_{\theta} \frac{1}{|Q|} \sum_{q \in Q} |f_\theta(q)|. \quad (4)$$

We visualize the effect of the geometric regularization in Fig. 2 (e). Compared to the surface reconstruction without the geometric regularization in Fig. 2 (d), the geometric regularization can infer a more accurate signed distance field, which leads to surface reconstruction with higher fidelity.

Loss Function. Our loss function pushes SDFs f_θ to project queries q onto a surface along the shortest projection path. We learn f_θ from a sparse point cloud \mathbf{G} with the on-surface prior by combining Eq. (3) and Eq. (4) below, where λ is a balance weight,

$$\min_{\theta} \frac{1}{|Q|} \sum_{q \in Q} (|f_\phi(q - sd, Knn(q - sd))| + \lambda |f_\theta(q)|). \quad (5)$$

Implementation. We set $\lambda = 0.4$ to balance the two constraints in Eq. (5). We leverage the same network architectures as NP [20] to learn the function $f_\theta(q)$ and $f_\phi(p, t)$. Additionally, we leverage an MLP with 8 layers in f_ϕ to learn the feature of K nearest neighbors $t = Knn(p)$ of p . In addition, we regard the point p as the origin, and normalize the coordinates of t on the sparse point clouds according to the coordinates of p , such that,

$$t \leftarrow t - p \quad \text{and} \quad p \leftarrow p - p. \quad (6)$$

The purpose of this normalization is to make the on-surface prior learned from various regions on different shapes comparable. Note that we conduct this normalization in both learning $f_\phi(p, t)$ in Eq. (2) and leveraging the learned f_ϕ as a prior in Eq. (3) and Eq. (5).

4. Experiments

4.1. Setup

Dataset. We evaluate our method in surface reconstruction for shapes and scenes. For shapes, we leverage a subset of ShapeNet [6] with the same train and test splitting as [15, 20]. To evaluate our generalization ability, we employ our trained model to produce results under another unseen subset of ShapeNet [6]. For scenes, we report our results under SceneNet [9], 3D Scene [36], and Paris-rue-Madame [27], where the latter two are real scanning datasets.

Details. In surface reconstruction for shapes, we uniformly sample 500 points on each shape as sparse point clouds in both training and test sets. We leverage the training set of each class to form the dataset T . To learn f_ϕ from T , we sample dense points on and around each shape as queries q_i . Each query is paired with its $K = 50$ nearest points in the sparse point cloud, and the $K = 50$ nearest points form a local region t_i . Moreover, we also calculate the unsigned distance l_i for each q_i . With the on-surface prior provided by the learned f_ϕ , we learn SDFs f_θ for each single sparse point cloud in the test set by overfitting the shape without using the condition c .

To reconstruct surfaces for scenes, we leverage the f_ϕ learned from table class in ShapeNet as the on-surface prior to learn f_θ for each single scene. To evaluate our performance under different point densities, we sample different numbers of points as the sparse point clouds. In 3D Scene dataset, we uniformly sample 100, 500, and 1000 points per m^2 to form the sparse point clouds for each scene, while we uniformly sample 20 and 100 points per m^2 for each scene in SceneNet dataset. In Paris-rue-Madame, the point cloud containing $10M$ points was obtained by scanning on a street. We randomly sample $1M$ points as the sparse input.

Metric. We leverage L1 Chamfer Distance (L1CD), L2 Chamfer Distance (L2CD), Normal Consistency (NC), and

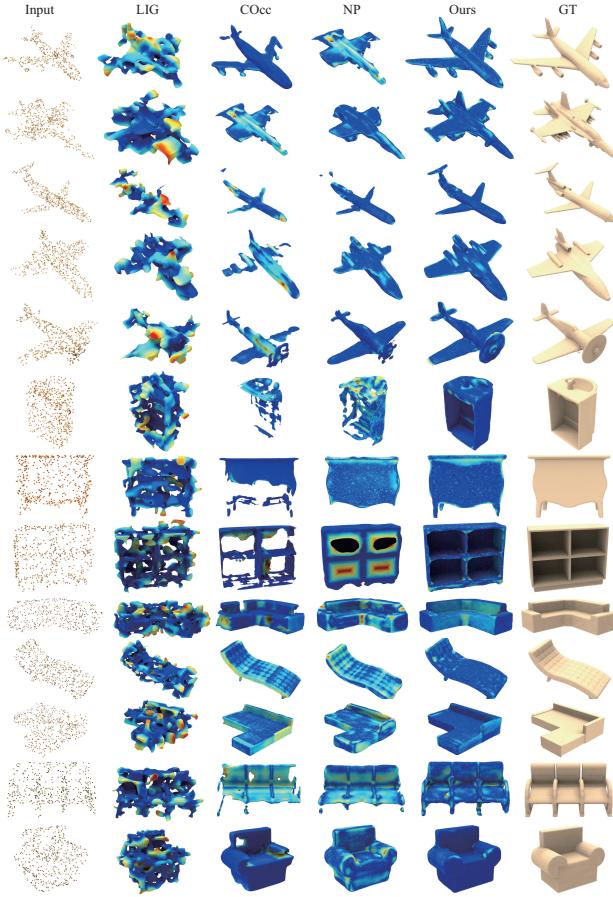


Figure 3. Visual comparison with LIG [12], COcc [29], NP [20] under ShapeNet. The input contains 500 points. Color indicates point to surface errors.

F-Score with a threshold of 0.001 for shapes and 0.025 for scenes. For CD, we sample $100K$ points on both the reconstructed and the ground truth surfaces for single shapes under ShapeNet, while sampling $1M$ points for scenes.

4.2. Surface Reconstruction on Shapes

ShapeNet. We first evaluate our method under ShapeNet. We leverage the pretrained models of COcc [29], LIG [12], and ISO [34] to produce their results on shapes with 500 points, where we also provide COcc and LIG the ground truth point normals. We produce the results of NP [20] by retraining it with the same shapes with 500 points as ours.

We report the numerical comparison under 8 classes in Tab. 1, 2, and 3. We achieve the best results in terms of all the three metrics in all shape classes. We found that current state-of-the-art methods are still struggling to reveal surfaces from sparse point clouds, while our method can handle the sparseness of point clouds well with the on-surface prior. We further demonstrate our advantages using surface reconstruction with error maps in Fig. 3. The visual comparison indicates that the latest methods are dramatically

Class	NP [20]	COcc [29]	LIG [12]	ISO [34]	Ours
airplane	0.082	0.078	0.185	0.776	0.076
cabinet	0.101	0.133	0.188	0.783	0.068
chair	0.163	0.167	0.183	0.822	0.083
display	0.074	0.118	0.183	0.727	0.068
lamp	0.087	0.187	0.207	1.146	0.066
sofa	0.088	0.101	0.178	0.649	0.071
table	0.187	0.187	0.187	0.827	0.053
vessel	0.063	0.089	0.179	0.661	0.057
mean	0.106	0.132	0.186	0.799	0.068

Table 1. L1CD $\times 10$ comparison under ShapeNet.

Class	NP [20]	COcc [29]	LIG [12]	ISO [34]	Ours
airplane	0.863	0.850	0.722	0.638	0.897
cabinet	0.850	0.816	0.657	0.545	0.888
chair	0.840	0.829	0.707	0.610	0.864
display	0.901	0.885	0.677	0.566	0.930
lamp	0.885	0.844	0.733	0.689	0.892
sofa	0.878	0.844	0.681	0.589	0.905
table	0.806	0.845	0.684	0.578	0.907
vessel	0.839	0.799	0.682	0.623	0.866
mean	0.858	0.839	0.693	0.605	0.894

Table 2. NC comparison under ShapeNet.

Class	NP [20]	COcc [29]	LIG [12]	ISO [34]	Ours
airplane	0.977	0.971	0.759	0.422	0.989
cabinet	0.955	0.891	0.793	0.387	0.983
chair	0.914	0.886	0.791	0.354	0.962
display	0.946	0.936	0.793	0.443	0.959
lamp	0.971	0.897	0.739	0.339	0.975
sofa	0.911	0.918	0.815	0.475	0.926
table	0.823	0.816	0.788	0.369	0.836
vessel	0.988	0.963	0.797	0.537	0.989
mean	0.936	0.909	0.784	0.416	0.952

Table 3. F-Score comparison under ShapeNet.

affected by the sparseness of points, which results in poor and incomplete surfaces. While our method is able to reveal surfaces from sparse point clouds in higher accuracy.

To evaluate the generalization ability of our learned on-surface prior, we leverage the f_ϕ learned under table class to reconstruct surfaces for sparse point clouds from 5 unseen classes in Tab. 4. The numerical comparison with IMLS [16] shows that we can leverage the learned class-agnostic and object-agnostic on-surface prior to reconstruct more accurate surfaces for unseen point clouds, which demonstrates our better generalization ability than IMLS. Note that we leverage the pretrained model of IMLS that was trained under 13 shape classes and evaluate it under the same 5 unseen classes with the same input as ours. We further highlight our advantages in visual comparison with IMLS in Fig. 6.

4.3. Surface Reconstruction on Scenes

We leverage the f_ϕ learned from table class in ShapeNet as the on-surface prior for scenes. We report our results by learning f_θ to overfit each scene.

SceneNet. We further evaluate our surface reconstruction performance under SceneNet [9]. With different point densities, we evaluate our surface reconstruction accuracy in different metrics. We leverage the pretrained models of COC-

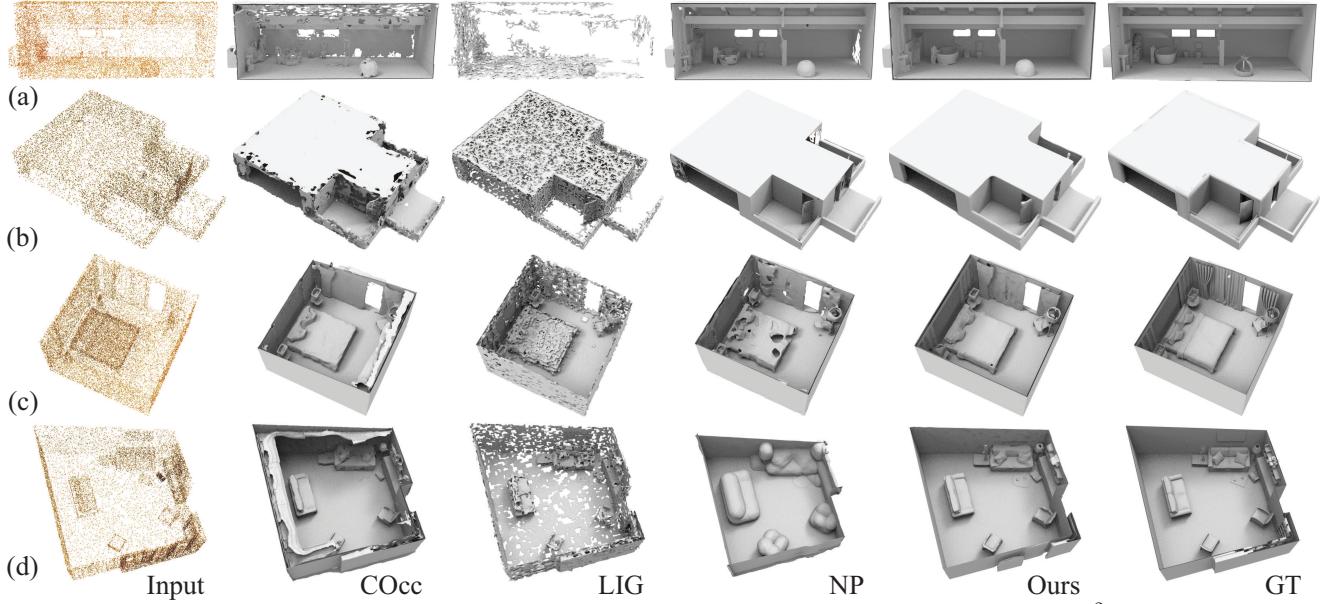


Figure 4. Visual comparison with COcc [29], LIG [12], NP [20] under SceneNet. The input contains $20 \text{ points}/\text{m}^2$ in (a) and (b), and $100 \text{ points}/\text{m}^2$ in (c) and (d). More comparisons can be found in our supplemental materials.

c [29] and LIG [12] to produce their results and retrain NP [20] using the same input point clouds as ours. The numerical comparison in Tab. 5 demonstrates that our method significantly outperforms other methods even there are only 20 points per m^2 in each scene. We further demonstrate our advantages over the state-of-the-art in Fig. 4. Visual comparison indicates that current methods cannot produce smooth and complete surfaces from sparse point clouds, while we show our superior performance over them by producing surfaces with more detailed geometry.

3D Scene. We leverage the same strategy to evaluate our method under 3D scenes dataset. With different numbers of points in the input point clouds, we compare our method with COcc [29], LIG [12] and DeepLS [5]. With the ground truth point normals, we report the results of COcc and LIG using their pretrained models, and we retrain DeepLS by overfitting each point cloud. The numerical comparison in Tab. 6 demonstrates our superior performance in all scene classes with different point densities. We further highlight our performance in the visual comparison in Fig. 5. The visual comparison with 100 points/ m^2 indicates that the latest methods still struggle to reveal surfaces from sparse point clouds while ours can produce more accurate surfaces.

Paris-rue-Madame. Finally, we evaluate our method using a large-scale real scanning. We split the point cloud into multiple sections and leverage each part to train our method or evaluate other methods. We visualize our reconstruction of the entire scene and some partial sections in Fig. 7 and Fig. 8, respectively. The visual comparison with the state-of-the-art demonstrates that our method can reconstruct more accurate surfaces from sparse point clouds.

These plausible results under scenes indicate that our

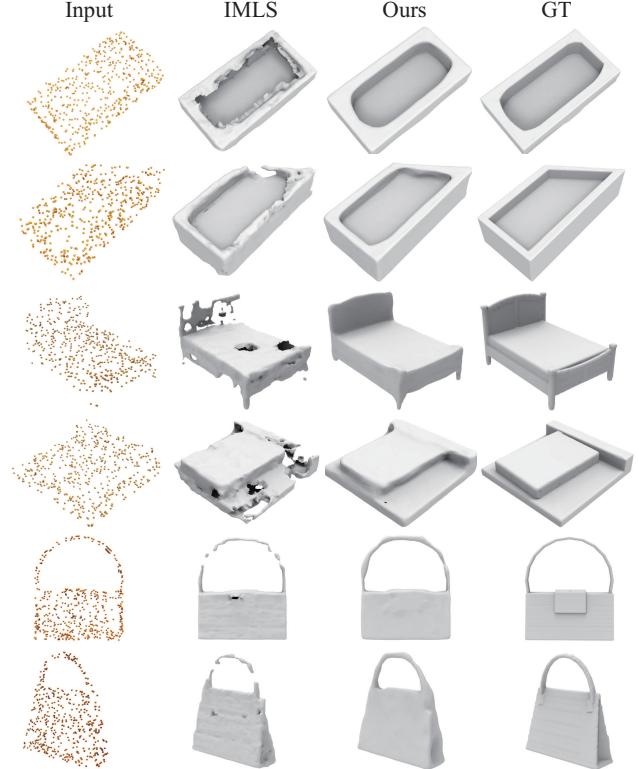


Figure 6. Visual comparison with IMLS [16] in unseen classes under ShapeNet.

method can reveal surfaces for scenes with complex geometry details, and our on-surface prior has remarkable generalization ability.

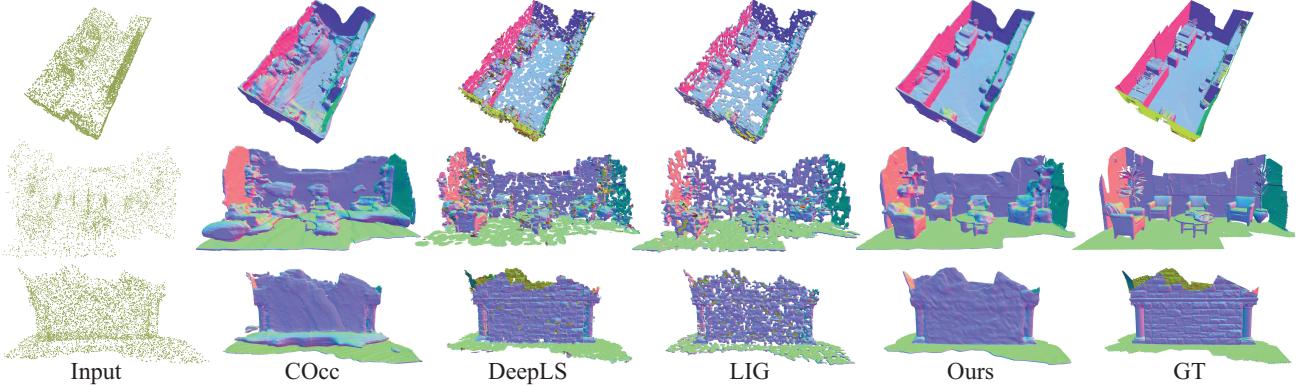


Figure 5. Visual comparison with COcc [29], DeepLS [5], LIG [12] under SceneNet. The input contains 100 points/ m^2 . Color indicates normals.

	bed			bag			bathtub			bottle			pillow		
	L1CD	NC	FScore												
IMLS [16]	0.077	0.838	0.963	0.058	0.926	0.981	0.069	0.889	0.977	0.044	0.954	0.994	0.039	0.959	0.997
Ours	0.069	0.908	0.981	0.054	0.936	0.990	0.056	0.952	0.993	0.043	0.977	0.995	0.039	0.970	0.999

Table 4. Generalization ability evaluation under ShapeNet. L1CD×10.

	SDF	SDF+ f_θ	Binary+ f_θ	UDF+ f_θ
L1CD	0.050	0.049	0.083	0.015
NC	0.827	0.833	0.743	0.928

Table 7. Effect of on-surface prior.

4.4. Ablation Study

We conduct ablation studies to justify the effectiveness of our method. We report numerical comparison in surface reconstruction, using 10 shapes to learn f_ϕ during training and using another 10 shapes to learn f_θ during testing, both of which are from ABC dataset [14]. We sample 500 points on each shape.

Prior. We first justify the effectiveness of our on-surface prior in Tab. 7. We learn a signed distance function (“SDF”) as the on-surface prior, and directly leverage it to reconstruct surfaces without learning f_θ . We found that SDF can not reconstruct plausible surfaces, even with f_θ (“SDF+ f_θ ”). The reason is that the on-surface prior is learned from various small regions without normalizing orientation. This makes the network hard to determine the sign of the distances. We can also learn the prior as a binary classifier, but the result (“Binary+ f_θ ”) gets worse. Our method learns on-surface prior as an unsigned distance function (“UDF”), which achieves the best performance.

Balance Weights. Then, we explore the effect of balance weight λ in Eq. (5). We try different candidates including $\{0, 0.2, 0.4, 0.6\}$, and report the results in Tab. 8. The results of “0” are the worst, which highlights the importance of the geometric regularization in Eq. (4). If the weight is too large (“0.6”), it turns to encourage f_θ to output 0 for any query locations. If the weight is too small (“0.2”), the geometry regularization can not constrain f_θ to predict minimum distances to the surface. We found “0.4” is a proper tradeoff.

λ	0	0.2	0.4	0.6
L1CD	0.050	0.016	0.015	0.023
NC	0.569	0.894	0.928	0.890

Table 8. Effect of balance weight λ .

K	25	50	100	200
L1CD	0.016	0.015	0.016	0.018
NC	0.745	0.928	0.919	0.897

Table 9. Effect of K nearest neighbors.

Normalization	No Trans, No Rot	Trans, No Rot	No Trans, Rot	Trans, Rot
L1CD	0.084	0.015	0.086	0.035
NC	0.766	0.928	0.745	0.843

Table 10. Effect of translation and rotation.

K Nearest Neighbors. We explore the effect of K on the on-surface prior learned by f_ϕ . We try different K to form the nearest region on sparse point clouds for each query, such as $\{25, 50, 100, 200\}$. The numerical comparison in Tab. 9 indicates that K slightly affects the performance, and achieves the best with 50 neighboring points.

Normalization. We explore different strategies to normalize the input of f_ϕ , i.e., p and its K nearest neighbors t , by translating or rotating when learning f_ϕ . We leverage Eq. (6) for translation. For rotation, we align the vector connecting p and its nearest point on t with the positive direction of axis Z. We report the results in Tab. 10. We found that the translation is the key to learning good prior as f_ϕ , while rotation does not help.

Network. We found that the MLP we leveraged to learn features of K nearest neighbors t in f_ϕ performs much better than PointNet [25] and PointNet++ [26], as demonstrated in Tab. 11. We found PointNet and PointNet++ can not understand sparse points well due to the maxpooling while MLP takes full advantage of the point order sorted by the distances to achieve a remarkable performance.

		Livingroom			Bathroom			Bedroom			Kitchen			Office			Mean		
		L1CD	NC	Fscore															
$20/m^2$	LIG [12]	0.032	0.719	0.790	0.030	0.737	0.807	0.029	0.735	0.818	0.029	0.727	0.817	0.033	0.737	0.805	0.030	0.730	0.808
	COcc [29]	0.026	0.895	0.955	0.025	0.862	0.988	0.028	0.823	0.976	0.028	0.849	0.982	0.030	0.829	0.958	0.027	0.852	0.971
	NP [20]	0.068	0.827	0.718	0.072	0.716	0.658	0.044	0.782	0.740	0.069	0.720	0.689	0.066	0.834	0.663	0.037	0.776	0.693
	Ours	0.025	0.904	0.961	0.018	0.924	0.991	0.023	0.919	0.976	0.025	0.911	0.983	0.029	0.851	0.967	0.024	0.902	0.975
$100/m^2$	LIG [12]	0.019	0.922	0.919	0.018	0.930	0.915	0.017	0.918	0.920	0.016	0.920	0.936	0.020	0.910	0.936	0.018	0.920	0.925
	COcc [29]	0.026	0.895	0.979	0.025	0.910	0.979	0.026	0.890	0.980	0.027	0.898	0.981	0.027	0.894	0.985	0.026	0.897	0.981
	NP [20]	0.069	0.883	0.799	0.028	0.907	0.893	0.032	0.890	0.878	0.042	0.896	0.838	0.066	0.866	0.733	0.047	0.888	0.828
	Ours	0.018	0.960	0.985	0.015	0.947	0.984	0.013	0.960	0.983	0.012	0.950	0.985	0.019	0.921	0.990	0.015	0.947	0.985

Table 5. Numerical comparison in surface reconstruction under SceneNet.

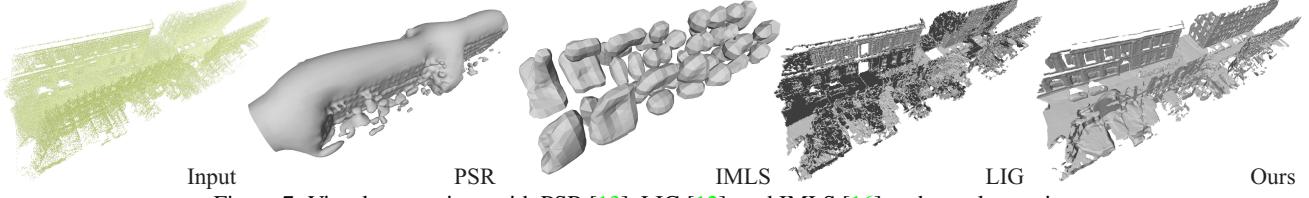


Figure 7. Visual comparison with PSR [13], LIG [12], and IMLS [16] under real scanning.

		Burghers			Lounge			Copyroom			Stonewall			Totempole		
		L2CD	L1CD	NC												
$100/m^2$	COcc [29]	8.904	0.040	0.890	6.979	0.041	0.884	6.78	0.041	0.856	12.22	0.051	0.903	4.412	0.041	0.874
	LIG [12]	3.112	0.044	0.839	9.128	0.054	0.833	4.363	0.039	0.804	5.143	0.046	0.853	9.58	0.062	0.887
	DeepLS [5]	3.111	0.050	0.856	3.894	0.056	0.764	1.498	0.033	0.777	2.427	0.038	0.885	4.214	0.043	0.908
	Ours	0.544	0.018	0.922	0.435	0.013	0.929	0.434	0.017	0.911	0.371	0.016	0.950	3.986	0.040	0.889
$500/m^2$	COcc [29]	26.97	0.081	0.905	9.044	0.046	0.894	10.08	0.046	0.885	17.70	0.063	0.909	2.165	0.024	0.937
	LIG [12]	3.080	0.046	0.840	6.729	0.052	0.831	4.058	0.038	0.810	4.919	0.043	0.878	9.38	0.062	0.887
	DeepLS [5]	0.714	0.020	0.923	10.88	0.077	0.814	0.552	0.015	0.907	0.673	0.018	0.951	21.15	0.122	0.927
	Ours	0.609	0.018	0.930	0.529	0.013	0.926	0.483	0.014	0.908	0.666	0.013	0.955	2.025	0.041	0.954
$1000/m^2$	COcc [29]	27.46	0.079	0.907	9.54	0.046	0.894	10.97	0.045	0.892	20.46	0.069	0.905	2.054	0.021	0.943
	LIG [12]	3.055	0.045	0.835	9.672	0.056	0.833	3.61	0.036	0.810	5.032	0.042	0.879	9.58	0.062	0.887
	DeepLS [5]	0.401	0.017	0.920	6.103	0.053	0.848	0.609	0.021	0.901	0.320	0.015	0.954	0.601	0.017	0.950
	Ours	1.339	0.031	0.929	0.432	0.014	0.934	0.405	0.014	0.914	0.266	0.014	0.957	1.089	0.029	0.954

Table 6. Surface reconstruction for point clouds under 3D Scene. L2CD×1000.

	PointNet [25]	PointNet++ [26]	MLP
L1CD	0.076	0.070	0.015
NC	0.784	0.797	0.928

Table 11. Effect of network.

Density	250	500	1000	2000
L1CD	0.020	0.015	0.013	0.012
NC	0.901	0.928	0.937	0.939

Table 12. Generalization ability of f_ϕ to point densities.

	Noise	0%	0.5%	1%
L1CD	0.015	0.017	0.018	
NC	0.928	0.912	0.908	

Table 13. Generalization ability of f_ϕ to noisy points.



Figure 8. Surface reconstruction from real scanning.

Generalization. We further evaluate the generalization ability of our on-surface prior learned by f_ϕ to different point densities during test. We leverage the f_ϕ trained with $K = 50$ nearest neighbors on shapes represented by 500 points to reconstruct surfaces for shapes with different point densities, such as $\{250, 500, 1000, 2000\}$. Tab. 12 and Fig. 9 (a) demonstrate that f_ϕ generalizes better to higher point densities than lower ones.

Noise. Besides our surface reconstruction under real scanning with noise in Fig. 8 and Fig. 7, we further report the generalization ability of f_ϕ to noise. We leverage the learned f_ϕ to reconstruct surfaces from noisy point clouds with two standard deviations including 0.5% and 1%. The comparison in Tab. 13 and Fig. 9 (b) demonstrates that f_ϕ is able to generalize to different level noise.

Limitation. Although we show our superior performance on sparse point clouds, we can not handle the incomplete point clouds, which is an extreme sparse scenario. It would be a good direction to combine the shape completion prior in the future work.

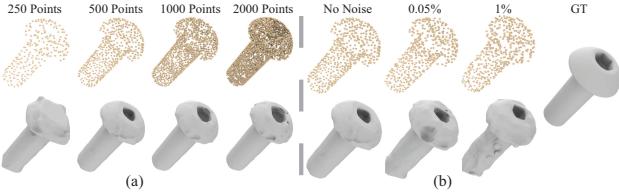


Figure 9. Generalization ability of f_ϕ to (a) point densities and (b) noisy points.

5. Conclusion

We resolve the issue of reconstructing surfaces from sparse point clouds, which surpasses the state-of-the-art methods a lot. We achieve this by learning SDFs via overfitting a sparse point cloud with an on-surface prior. We successfully learn class-agnostic and object-agnostic on-surface prior to reveal surfaces from sparse point clouds in a data-driven way. Our method is able to further leverage the learned on-surface prior with a geometric regularization to learn highly accurate SDFs for unseen sparse point clouds. Our method does not require signed distances or point normals to learn SDFs and the learned on-surface prior demonstrates remarkable generalization ability. Our method outperforms the latest methods in surface reconstruction from sparse point clouds under different benchmarks.

References

- [1] Matan Atzmon and Yaron Lipman. Sal: Sign agnostic learning of shapes from raw data. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. [1](#), [2](#)
- [2] Matan Atzmon and yaron Lipman. SALD: sign agnostic learning with derivatives. In *International Conference on Learning Representations*, 2021. [2](#)
- [3] Yizhak Ben-Shabat, Chamin Hewa Koneputugodage, and Stephen Gould. Digs : Divergence guided shape implicit neural representation for unoriented point clouds. *CoRR*, abs/2106.10811, 2021. [2](#)
- [4] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin. The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 5(4):349–359, 1999. [1](#), [2](#)
- [5] Rohan Chabra, Jan Eric Lenssen, Eddy Ilg, Tanner Schmidt, Julian Straub, Steven Lovegrove, and Richard A. Newcombe. Deep local shapes: Learning local SDF priors for detailed 3D reconstruction. In *European Conference on Computer Vision*, volume 12374, pages 608–625, 2020. [2](#), [6](#), [7](#), [8](#)
- [6] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015. [4](#)
- [7] Philipp Erler, Paul Guerrero, Stefan Ohrhallinger, Niloy J. Mitra, and Michael Wimmer. Points2Surf: Learning implicit surfaces from point clouds. In *European Conference on Computer Vision*, 2020. [1](#), [2](#)
- [8] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. *arXiv*, 2002.10099, 2020. [1](#), [2](#)
- [9] A. Handa, V. Patraucean, V. Badrinarayanan, S. Stent, and R. Cipolla. Understanding realworld indoor scenes with synthetic data. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4077–4085, 2016. [4](#), [5](#)
- [10] Rana Hanocka, Gal Metzger, Raja Giryes, and Daniel Cohen-Or. Point2mesh: A self-prior for deformable meshes. *ACM Trans. Graph.*, 39(4), 2020. [2](#)
- [11] Meng Jia and Matthew Kyan. Learning occupancy function from point clouds for surface reconstruction. *arXiv*, 2010.11378, 2020. [2](#)
- [12] Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, and Thomas Funkhouser. Local implicit grid representations for 3D scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. [1](#), [2](#), [5](#), [6](#), [7](#), [8](#)
- [13] Michael M. Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transactions Graphics*, 32(3):29:1–29:13, 2013. [1](#), [2](#), [8](#)
- [14] Sebastian Koch, Albert Matveev, Zhongshi Jiang, Francis Williams, Alexey Artemov, Evgeny Burnaev, Marc Alexa, Denis Zorin, and Daniele Panozzo. ABC: A big cad model dataset for geometric deep learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2019. [7](#)
- [15] Minghua Liu, Xiaoshuai Zhang, and Hao Su. Meshing point clouds with predicted intrinsic-extrinsic ratio guidance. In *European Conference on Computer vision*, 2020. [2](#), [4](#)
- [16] Shi-Lin Liu, Hao-Xiang Guo, Hao Pan, Pengshuai Wang, Xin Tong, and Yang Liu. Deep implicit moving least-squares functions for 3D reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021. [1](#), [2](#), [5](#), [6](#), [7](#), [8](#)
- [17] S. Lombardi, M. R. Oswald, and M. Pollefeys. Scalable point cloud-based reconstruction with local implicit functions. In *2020 International Conference on 3D Vision (3DV)*, pages 997–1007, 2020. [2](#)
- [18] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. *Computer Graphics*, 21(4):163–169, 1987. [1](#), [2](#)
- [19] Yiming Luo, Zhenxing Mi, and Wenbing Tao. Deepdt: Learning geometry from delaunay triangulation for surface reconstruction. *CoRR*, abs/2101.10353, 2020. [2](#)
- [20] Baorui Ma, Zhizhong Han, Yu-Shen Liu, and Matthias Zwicker. Neural-pull: Learning signed distance functions from point clouds by learning to pull space onto surfaces. In *International Conference on Machine Learning*, 2021. [1](#), [2](#), [3](#), [4](#), [5](#), [6](#), [8](#)
- [21] Julien N. P. Martel, David B. Lindell, Connor Z. Lin, Eric R. Chan, Marco Monteiro, and Gordon Wetzstein. ACORN: adaptive coordinate networks for neural scene representation. *CoRR*, abs/2105.02788, 2021. [2](#)

- [22] Zhenxing Mi, Yiming Luo, and Wenbing Tao. Ssrnet: Scalable 3D surface reconstruction network. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2020. 2
- [23] Yutaka Ohtake, Alexander G. Belyaev, Marc Alexa, Greg Turk, and Hans-Peter Seidel. Multi-level partition of unity implicits. *ACM Transactions on Graphics*, 22(3):463–470, 2003. 1, 2, 3
- [24] Songyou Peng, Chiyu ”Max” Jiang, Yiyi Liao, Michael Niemeyer, Marc Pollefeys, and Andreas Geiger. Shape as points: A differentiable poisson solver. In *Advances in Neural Information Processing Systems*, 2021. 2
- [25] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 7, 8
- [26] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, pages 5105–5114, 2017. 7, 8
- [27] Andrés Serna, Beatriz Marcotegui, François Goulette, and Jean-Emmanuel Deschaud. Paris-rue-madame database - A 3D mobile laser scanner dataset for benchmarking urban detection, segmentation and classification methods. In *International Conference on Pattern Recognition Applications and Methods*, pages 819–824, 2014. 4
- [28] Yawar Siddiqui, Justus Thies, Fangchang Ma, Qi Shan, Matthias Nießner, and Angela Dai. Retrievalfuse: Neural 3d scene reconstruction with a database. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 2
- [29] Lars Mescheder, Marc Pollefeys, Andreas Geiger, Songyou Peng, Michael Niemeyer. Convolutional occupancy networks. In *European Conference on Computer Vision*, 2020. 1, 2, 5, 6, 7, 8
- [30] Towaki Takikawa, Joey Litalien, Kangxue Yin, Karsten Kreis, Charles Loop, Derek Nowrouzezahrai, Alec Jacobson, Morgan McGuire, and Sanja Fidler. Neural geometric level of detail: Real-time rendering with implicit 3D shapes. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 2
- [31] Jiapeng Tang, Jiabao Lei, Dan Xu, Feiying Ma, Kui Jia, and Lei Zhang. Sa-convolnet: Sign-agnostic optimization of convolutional occupancy networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. 2
- [32] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Carsten Stoll, and Christian Theobalt. PatchNets: Patch-Based Generalizable Deep Implicit 3D Shape Representations. *European Conference on Computer Vision*, 2020. 2
- [33] Francis Williams, Teseo Schneider, Claudio Silva, Denis Zorin, Joan Bruna, and Daniele Panozzo. Deep geometric prior for surface reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2
- [34] Wang Yifan, Shihao Wu, Cengiz Oztireli, and Olga Sorkine-Hornung. Iso-points: Optimizing neural implicit surfaces with hybrid representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 374–383, 2021. 2, 5
- [35] Wenbin Zhao, Jiabao Lei, Yuxin Wen, Jianguo Zhang, and Kui Jia. Sign-agnostic implicit learning of surface self-similarities for shape modeling and reconstruction from raw point clouds. *CoRR*, abs/2012.07498, 2020. 2
- [36] Qian-Yi Zhou and Vladlen Koltun. Dense scene reconstruction with points of interest. *ACM Transactions on Graphics*, 32(4):112:1–112:8, 2013. 4