VoronoiNet

General Functional Approximators with Local Support

Francis Williams
New York University

Jérôme Parent-Lévesque
McGill University

Derek Nowrouzezahrai
McGill University

Daniele Panozzo
New York University

Kwang Moo Yi
University of Victoria

Andrea Tagliasacchi
Google Brain

Abstract

Voronoi diagrams are highly compact representations that are used in various Graphics applications. In this work, we show how to embed a differentiable version of it – via a novel deep architecture – into a generative deep network. By doing so, we achieve a highly compact latent embedding that is able to provide much more detailed reconstructions, both in 2D and 3D, for various shapes. In this tech report, we introduce our representation and present a set of preliminary results comparing it with recently proposed implicit occupancy networks.

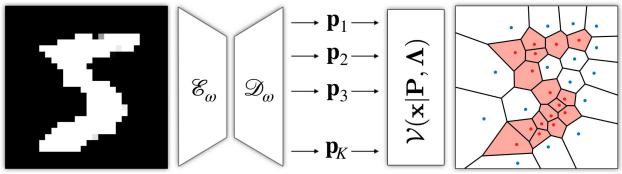

Figure 1. We propose a new differentiable *implicit* representation of solid object based on Voronoi diagrams. An encoder $\mathcal{E}_K$ generate a latent representation, which a decoder $\mathcal{D}_K$ converts into a collection of K sites $\{\mathbf{p}_k\}$. Our layer receives these sites in input, and generate a function that can be evaluated at a point $\mathbf{x}$.

1. Introduction

Choosing a shape representation is a fundamental problem for any geometric task. Especially, with the advent of deep methods for geometry, it defines what operations are possible (e.g. convolution), what choices of architecture can be used (e.g. graph [19] or point networks [21, 22]), and what input modality (e.g. point clouds or images) can be used for training. Naturally, finding a proper differentiable representation for geometry has been of much research interest recently, with much focus on 3D [18, 20, 8, 23, 21, 12]. A wide variety of 3D representations exist in the literature and are used for a variety of tasks from surface reconstruction [13, 3, 14], shape completion [9], predicting shape from images [8], semantic segmentation [21] and many more.

At a high level, geometric representations can be grouped into two: *explicit* representations, where the surface of an object is explicitly represented using for example, meshes [15], parameterized patches [12, 24] or point clouds [21, 22]; and *implicit* methods, where a 3D object is defined by a scalar function in $\mathbb{R}^3$ (for example by defining the surface as a level set of this function) [18, 8, 23, 11, 20, 4]. With deep networks, a recent trend is to use a neural network to represent the scalar function for a shape [8, 18, 20, 24]. Explicit representations have the benefit that they make surface extractions easy – e.g. via Marching Cubes [17] – while the implicit ones are easy to embed into a deep network with simple architectures. Recently, *hybrid* representations [10, 7] have been proposed to combine the best of both.

Of particular relevance to our work is CvxNet [10], which represent shapes as the intersection of a finite number of half spaces. This representation is a universal approximator of convex domains – similar to ours – as well as non-convex ones via composition. However, they are still *implicit* when it comes to modelling overlap. They *train* to make their decompositions non-overlapping through an additional loss term and therefore have no guarantee that it would also be non-overlapping during inference. While this can be of minor importance for reasoning tasks such as shape classification, it is problematic for others such as physical simulation.

Inspired by [1], we propose a novel representation that guarantees non-overlapping convexes. In other words, any network trained with our representation generates non-overlapping convexes *by construction*. We encode geometric information in the form of a point set $\mathbf{P}=\{\mathbf{p}_k\}$, and generates the collection of convexes as the corresponding collection of their *Voronoi cells* $\mathbf{C}=\{\mathbf{c}_k\}$. This representation is hybrid: the position of the seeds is *explicit*, and extracting the surface only requires to compute their *Voronoi*

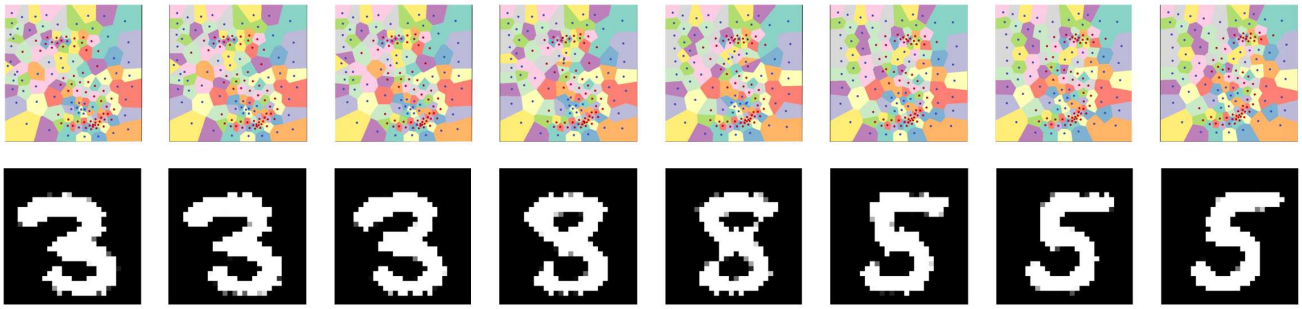

Figure 2. We encode the leftmost (3) and rightmost (5) digits in latent space and then linearly interpolate the corresponding latent codes.

Diagram – a task for which a number of robust and efficient software libraries exist [5]. Note that differently from iso-surface extraction, the Voronoi Diagram is unique and resolution independent – no parameter needs to be selected to compute it. Interestingly for our purposes, it is possible to closely approximate the Voronoi Diagram with a differentiable *implicit* function, which is ideal for training.

2. Method

We follow the trend pioneered by [8] and seek *functional* networks – where the output of our network is a *function* that can be queried at a desired location $\mathbf{x}$. Given the fixed vector $\mathbf{\Lambda}=(\lambda_k \in \{0,1\})_k$, we express this function as the the piecewise constant function over the Voronoi diagram of the point set $\mathbf{P} = \{\mathbf{p}_k \in \mathbb{R}^d\}_k$ where the value of the function at points in the k^{th} cell have value $\boldsymbol{\lambda}_k$:

$$\mathcal{V}(\mathbf{x}|\mathbf{\Lambda},\mathbf{P}) = \mathbf{\Lambda}\left[\arg\min_k \{\|\mathbf{x}-\mathbf{p}_k\|_2\}\right] \tag{1}$$

where we assume that $\Sigma_k \boldsymbol{\lambda}_k = |\mathbf{\Lambda}|/2$ – in other words, we fix half of the sites to represent the "inside" (1) of a shape, and other half to represent the "outside" (0) of a shape.

Given an input $\mathcal{I}$ (e.g. image, point cloud, voxel grid) from a training dataset $\{\mathcal{I}_n\}$, an encoder $\mathcal{E}_\omega$ maps $\mathcal{I}$ to a latent code z which a decoder $\mathcal{D}_\omega$ maps to the collection of Voronoi centers: $\mathbf{P}=\mathcal{D}_\omega(\mathcal{E}_\omega(\mathcal{I}))$. Figure 1 illustrates this architecture visually. The parameters of encoder and decoder are then trained by minimizing a reconstruction loss:

$$\mathcal{L}_{\text{rec}}(\omega) = \sum_n \mathbb{E}_{\mathbf{x}\in[0,1]^D}\left[\|\mathcal{O}_n(\mathbf{x}) - \mathcal{V}(\mathbf{x}|\mathcal{D}_\omega(\mathcal{E}_\omega(\mathcal{I}_n)))\|_2\right] \tag{2}$$

where $\mathcal{O}_n$ is the ground truth occupancy function corresponding to $\mathcal{I}_n$. If we compare our representation to the one provided by ReLU functional networks [18, 8, 20], we differ in a fundamental way: our learnable parameters have *localized* support, while the transition boundaries of an MLP generally have a *global* support.

Regularizers. While the reconstruction loss $\mathcal{L}_{\text{rec}}$ lies at the core of our method, minimizing this loss is ill-posed. In particular, there exist an infinite space of solutions where voronoi cell agrees with the occupancy of the ground truth. To remedy this, we develop a number of regularizers that aid our training process. Notably, these losses do not typically produce pareto-optimal variants of the trained network.

Lemma 1. *Let $|\mathbf{P}| > 6$ be a set of points such that half of $\mathbf{p}_k \in \mathbb{R}^2$ are labelled 1, and let $\mathcal{O}_n = \mathcal{V}(\mathbf{x}|\mathbf{\Lambda},\mathbf{P})$ be the occupancy function of the associated Voronoi diagram. Assume that there are three points labelled 1 so that the triangle they form is contained in $\mathcal{O}_n$. Then, there exist an infinite number of minimizers $(\mathbf{P}^*,\mathbf{\Lambda}^*)$ to (2).*

Proof. Assume without loss of generality that p_1, p_2, p_3 are all labelled 1 and the triangle they form is inside $\mathcal{O}_n$. Then let q be any point inside this triangle. Label q with 1, and define $(\mathbf{P}^*,\mathbf{\Lambda}^*)$ by adding this labeled point to $(\mathbf{P},\mathbf{\Lambda})$. Then $\mathcal{V}(\mathbf{x}|\mathbf{\Lambda}^*,\mathbf{P}^*)$ is a minimizer of (2) for $\mathcal{O}_n$. In fact $\mathcal{V}(\mathbf{x}|\mathbf{\Lambda}^*,\mathbf{P}^*) = \mathcal{O}_n$ since the $(\mathbf{P}^*,\mathbf{\Lambda}^*)$ produces the same function as $(\mathbf{P},\mathbf{\Lambda})$. □

Soft-Voronoi. To differentiate through our Voronoi function, we generalize (1) by replacing the *argmin* with a *soft-argmin*. Given $D_k(\mathbf{x})=\|\mathbf{x}-\mathbf{p}_k\|_2$, we first define a vector $\mathbf{W}$:

$$\mathbf{W}_k(\mathbf{x}|\mathbf{P},\beta) = e^{-\beta D_k(\mathbf{x})}/\Sigma_k e^{-\beta D_k(\mathbf{x})} \tag{3}$$

where $\beta\in\mathbb{R}^+$ is a temperature parameter and then formulate the soft version of (1) as:

$$\mathcal{V}(\mathbf{x}|\mathbf{\Lambda},\mathbf{P},\beta) = \mathbf{\Lambda}\cdot\mathbf{W}(\mathbf{x}|\mathbf{P},\beta) \tag{4}$$

hence the temperature hyper-parameter β controls the soft-argmin approximation to argmin. In all experiments in the paper we set β=10,000.

Bounds loss. We naturally want to prevent our Voronoi sites from drifting far away from the data, which can be enforced in a smooth way via [7]:

$$\mathcal{L}_{\text{bound}}(\omega) = \sum_{\{\mathbf{p}_k\}}\sum_d \text{soft-bound}(\mathbf{p}_k[d]) \tag{5}$$

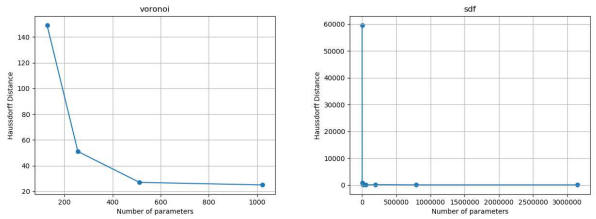

Figure 3. Plot of the number of parameters (x-axis) vs. Hausdorff distance (y-axis) from the ground truth for the overfitted sphere example using (left) Voronoi and (right) OccNet.

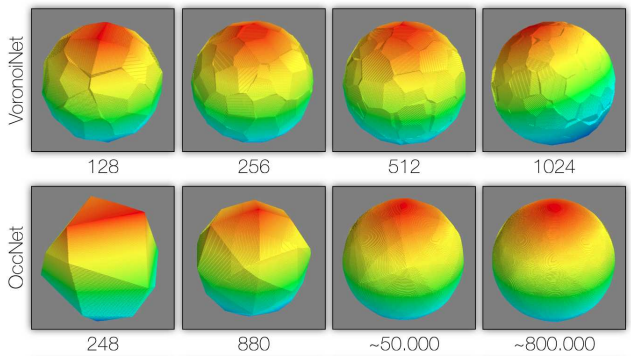

Figure 4. We compare the reconstruction power in terms of neural capacity of our VoronoiNet (top) vs. the one of traditional multi-layer perceptrons used in OccNet [18] (bottom) on a simple 3D sphere – note these are *overfitting* results on a single example.

where $[d]$ extracts the d^{th} dimension and soft-bound$(x) = \max(-x, 0) + \max(x-1, 0)$. We favor this to the use of output layers with bounded ranges as [7] noting how these can suffer of vanishing gradients.

Signed distance Loss. As we prescribe the Voronoi (inside/outside) classes $\mathbf{\Lambda}$ rather than optimizing them, it is clear that if the $\lambda_k{=}1$, then the corresponding $\mathbf{p}_k$ should be *inside*, or in other words $\mathcal{O}(\mathbf{p}_k){=}1$ (and symmetrically for $\lambda_k{=}0$). Hence, we can define a loss that induces strong gradients towards the satisfaction of this property. With $\phi^+(\mathbf{x})$ let us define the distance function to $\mathcal{O}$, and with $\phi^-(\mathbf{x})$ the distance function to its complement space $\bar{\mathcal{O}}(\mathbf{x}){=}1{-}\mathcal{O}(\mathbf{x})$, and then define:

$$\mathcal{L}_{\text{sdf}}(\omega) = \sum_k \boldsymbol{\lambda}_k \phi^+(\mathbf{p}_k) + (1 - \boldsymbol{\lambda}_k)\phi^-(\mathbf{p}_k) \qquad (6)$$

Note that all correct approximations $\mathbf{P}, \mathbf{\Lambda}$ of the ground truth occupancy lie in the null space of this loss. Thus, $\mathcal{L}_{\text{sdf}}$ simply accelerates training and does not prevent the network from finding a global minimum to the problem.

Centroidal Voronoi loss. To remedy the ill posedness (Lemma 1) of the reconstruction loss (2), we add a loss that pushes each Voronoi point towards the centroid of its corresponding cell. A Voronoi diagram whose points lie at the centroid of its cells is known as *centroidal*. Centroidal Voronoi tesselations have cells with roughly equal shape and have been used for many years in graphics to generate high quality tesselations of space [2, 6, 16]. Asking the Voronoi diagram to be as centroidal as possible prevents points from clustering and introduces a unique reconstruction minimum. Given m Voronoi sites $\mathbf{P}$, we augment the sites with $\sqrt{m}$ points on the boundary with 0 (outside), we compute their Delaunay triangulation, and express its corresponding graph Laplacian operator via a sparse matrix $\mathbf{L}$; a CVD-like loss can then be expressed by:

$$\mathcal{L}_{\text{cvd}}(\omega) = \sum_k \|\mathbf{L}\mathbf{p}_k\|_2^2 \qquad (7)$$

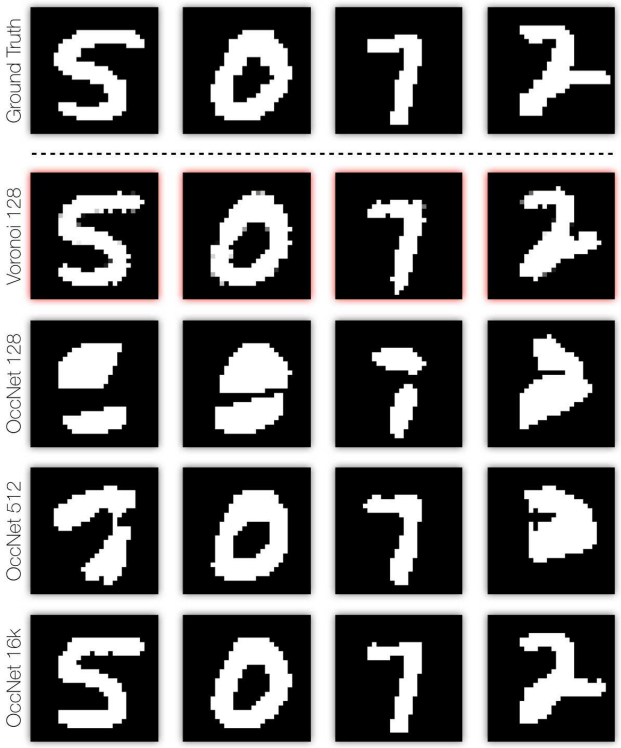

Figure 5. A qualitative comparisons of the representation power of different neural decoders as the number of degrees of freedom is increased.

3. Experiments and Results

Overfitting a Sphere. We start by evaluating the reconstruction power of our network in terms of number of degrees of freedom used for a simple 3D dataset (Figure 4). We compare our method to the state of the art OccNet [18] and DeepSDF [20]. Note that while both OccNet and DeepSDF guarantee C^0 continuity, the number of neurons necessary to generate reconstructions comparable to Voronoi networks in terms of Hausdorff distance to

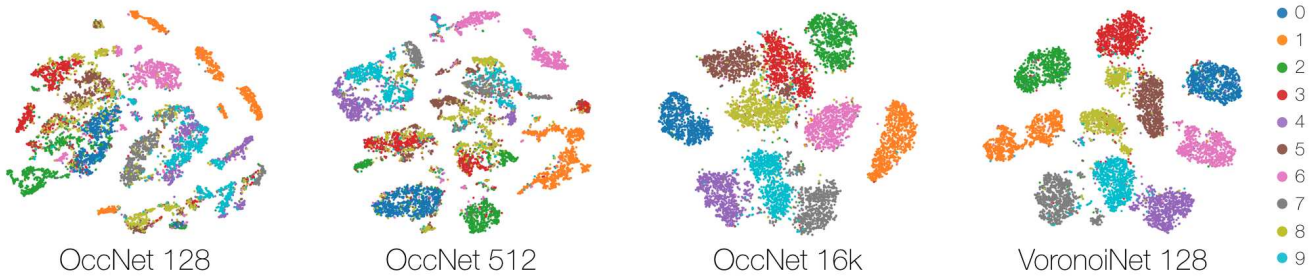

Figure 6. A tSNE embedding of our latent code on the MNIST dataset, where the ground truth class has been used to color their identity.

the ground truth is *three* orders of magnitudes larger than with our approach. Figure 3 plots the number of parameters for the function versus Haussdorff distance from the ground truth for all 3 methods. Figure 4 shows the reconstructions for each method visually.

MNIST. We evaluate our formulation on the MNIST dataset by treating the digits as an occupancy function in the $[0, 1]^2$ domain that needs to be predicted. We compare our method against OccNet [18]. Both methods use a 4 layer fully connected encoder with 1024 neurons per layer. The encoder maps an MNIST digit image to a 16 dimensional latent variable. The decoder for our method is a 3 layer fully connected network with 1024 neurons per layer which maps the latent code to 128 Voronoi cells. The decoder for occnet has one hidden layer with a varying number of neurons. The decoder maps a latent code and point $\mathbf{x}$ to a probability of occupancy.

Embedding space. We start by visualizing the tSNE embedding in Figure 6. Notice that while the method was trained in a self-supervised fashion, the latent space was able to organize the various digits by clearly separating the semantic classes. It is interesting to note how part of the "8" embedding space is wedged between the "3" and the "5", reflecting the geometric similarity between these characters, and the required topological changes to interpolate between them. To show this, we also visualize a path in the embedding space by encoding two digits, and then interpolating their latent codes; see Figure 2. Notice how the topology of the "9" is first converted into the one of a "5", then into a "6" and finally smoothly deformed towards the target configuration.

We conclude our experiments by evaluating (on the test set) the auto-encoding performance on MNIST. Note in this comparison we keep the capacity of the *encoder* portion of our auto-encoder consistent across the various baselines. In particular, we compare our Voronoi decoders to popular implicit pipelines that use a multi-layer perceptron as a (conditional) implicit decoder [18, 20, 8]. Figure 5 shows randomly drawn results illustrate how the Voronoi decoder allows for a significantly more compact representation of occupancy than occnet. Table 1 compares statistics of voronoi reconstructions versus occnet on the test set with varying number of degrees of freedom.

Method	Mean	Std	Med
OccNet 128	83.803001	28.211296	85.692169
OccNet 512	76.165771	28.211296	75.422150
OccNet 16k	52.658348	14.332524	53.036644
Voronoi 128	57.996124	17.018425	58.294270

Table 1. Autoencoder statistics for different methods with different degrees of freedom. Note how Voronoi with 128 cells is comparable to OccNet with with 4 orders of magnitude more parameters.

4. Conclusion and Future Work

We introduced a new differentiable layer for solid geometry representation leveraging the Voronoi diagram. Similarly to [18, 20, 8], we expect our solution to scale to the modeling of 3D objects with minor modifications. The challenge will be the identification of a random sampling tailored to evaluate the expectation of $\mathcal{L}_{\text{rec}}$. While CvxNet [10] introduced the idea of hybrid representation learning, where training is performed in the implicit domain, and inference in the explicit domain (i.e. generates meshes), our network can infer discrete geometry as the crust separating the inside/outside Voronoi cells, removing the need for the iso-surfacing post-processing (e.g. marching cubes [17]).

Our work is early in its stage. As future work, we plan to apply our method to higher dimensional data, to produce meshing of volume and not only surfaces, to analyze the benefit it brings in physical simulations.

References

[1] Narendra Ahuja, Byong An, and Bruce Schachter. Image representation using voronoi tessellation. *Computer Vision, Graphics, and Image Processing*, 29(3):286 – 295, 1985. 1

[2] Pierre Alliez, Eric Colin De Verdire, Olivier Devillers, and Martin Isenburg. Isotropic surface remeshing. In *Shape Modeling International.*, 2003. 3

[3] Matan Atzmon and Yaron Lipman. Sal: Sign agnostic learning of shapes from raw data. *arXiv:1911.10414*, 2019. 1

[4] Matan Atzmon, Haggai Maron, and Yaron Lipman. Point convolutional neural networks by extension operators. *CoRR*, abs/1803.10091, 2018. 1

[5] C Bradford Barber, David P Dobkin, David P Dobkin, and Hannu Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software (TOMS)*, 1996. 2

[6] Mario Botsch and Leif Kobbelt. A remeshing approach to multiresolution modeling. In *Proceedings of the symposium on Geometry processing*, 2004. 3

[7] Zhiqin Chen, Andrea Tagliasacchi, and Hao Zhang. Bsp-net: Generating compact meshes via binary space partitioning. *arXiv:1911.06971*, 2019. 1, 2, 3

[8] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. *Proc. of Comp. Vision and Pattern Recognition (CVPR)*, 2019. 1, 2, 4

[9] Angela Dai and Matthias Niezner. Scan2mesh: From unstructured range scans to 3d meshes. In *Proc. of Comp. Vision and Pattern Recognition (CVPR)*, 2019. 1

[10] Boyang Deng, Kyle Genova, Soroosh Yazdani, Sofien Bouaziz, Geoffrey Hinton, and Andrea Tagliasacchi. Cvxnet: Learnable convex decomposition. *arXiv:1909.05736*, 2019. 1, 4

[11] Kyle Genova, Forrester Cole, Daniel Vlasic, Aaron Sarna, William T Freeman, and Thomas Funkhouser. Learning shape templates with structured implicit functions. *arXiv:1904.06447*, 2019. 1

[12] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. Atlasnet: A papier-mache approach to learning 3d surface generation. *arXiv:1802.05384*, 2018. 1

[13] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. *Surface reconstruction from unorganized points*. 1992. 1

[14] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of Symposium on Geometry processing*, 2006. 1

[15] Ilya Kostrikov, Joan Bruna, Daniele Panozzo, and Denis Zorin. Surface networks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2540–2548, 2018. 1

[16] Yang Liu, Wenping Wang, Bruno Lévy, Feng Sun, Dong-Ming Yan, Lin Lu, and Chenglei Yang. On centroidal voronoi tessellation—energy smoothness and fast computation. *ACM Trans. on Graphics (Proc. of SIGGRAPH)*, 2009. 3

[17] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *ACM Trans. on Graphics (Proc. of SIGGRAPH)*, 1987. 1, 4

[18] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. *arXiv:1812.03828*, 2018. 1, 2, 3, 4

[19] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proc. of Comp. Vision and Pattern Recognition (CVPR)*, 2017. 1

[20] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. *Proc. of Comp. Vision and Pattern Recognition (CVPR)*, 2019. 1, 2, 3, 4

[21] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proc. of Comp. Vision and Pattern Recognition (CVPR)*, 2017. 1

[22] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *CoRR*, abs/1706.02413, 2017. 1

[23] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. *arXiv:1905.05172*, 2019. 1

[24] Francis Williams, Teseo Schneider, Cláudio T. Silva, Denis Zorin, Joan Bruna, and Daniele Panozzo. Deep geometric prior for surface reconstruction. *CoRR*, abs/1811.10943, 2018. 1