TextureNet: Consistent Local Parametrizations for Learning from High-Resolution Signals on Meshes

Jingwei Huang[1] Haotian Zhang[1] Li Yi[1] Thomas Funkhouser[2,3]
Matthias Nießner[4] Leonidas Guibas[1,5]

[1]Stanford University [2]Princeton University [3]Google
[4]Technical University of Munich [5]Facebook AI Research

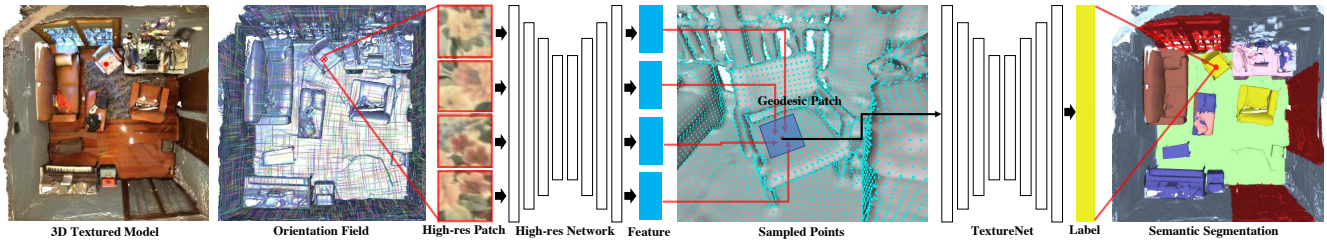


Figure 1: TextureNet takes as input a 3D textured mesh. The mesh is parameterized with a consistent 4-way rotationally symmetric (4-RoSy) field, which is used to extract oriented patches from the texture at a set of sample points. Networks of 4-RoSy convolutional operators extract features from the patches and used for 3D semantic segmentation.

Abstract

We introduce, TextureNet, a neural network architecture designed to extract features from high-resolution signals associated with 3D surface meshes (e.g., color texture maps). The key idea is to utilize a 4-rotational symmetric (4-RoSy) field to define a domain for convolution on a surface. Though 4-RoSy fields have several properties favorable for convolution on surfaces (low distortion, few singularities, consistent parameterization, etc.), orientations are ambiguous up to 4-fold rotation at any sample point. So, we introduce a new convolutional operator invariant to the 4-RoSy ambiguity and use it in a deep network to extract features from high-resolution signals on geodesic neighborhoods of a surface. In comparison to alternatives, such as PointNet-based methods which lack a notion of orientation, the coherent structure given by these neighborhoods results in significantly stronger features. As an example application, we demonstrate the benefits of our architecture for 3D semantic segmentation of textured 3D meshes. The results show that our method outperforms all existing methods on the basis of mean IoU by a significant margin in both geometry-only (6.4%) and RGB+Geometry (6.9-8.2%) settings.

1. Introduction

In recent years, there has been tremendous progress in RGB-D scanning methods that allow reliable tracking and reconstruction of 3D surfaces using hand-held, consumer-grade devices [8, 18, 27, 28, 41, 21, 11]. Though these methods are now able to reconstruct high-resolution textured 3D meshes suitable for visualization, understanding the 3D semantics of the scanned scenes is still a relatively open research problem.

There has been a lot of recent work on semantic segmentation of 3D data using convolutional neural networks (CNNs). Typically, features extracted from the scanned inputs (e.g., positions, normals, height above ground, colors, etc.) are projected onto a coarse sampling of 3D locations, and then a network of 3D convolutional filters is trained to extract features for semantic classification – e.g., using convolutions over voxels [42, 25, 30, 36, 9, 13], octrees [33], point clouds [29, 31], or mesh vertices [24]. The advantage of these approaches over 2D image-based methods is that convolutions operate directly on 3D data, and thus are relatively unaffected by view-dependent image effects, such as perspective, occlusion, lighting, and background clutter. However, the resolution of current 3D representations is generally quite low (2cm is typical), and so the ability of 3D CNNs to discriminate fine-scale semantic patterns is usually far below their color image counterparts [23, 16].

To address this issue, we propose a new convolutional neural network, TextureNet, that extracts features directly from high-resolution signals associated with 3D surface meshes. Given a map that associates high-resolution signals with a 3D mesh surface (e.g., RGB photographic tex-

ture), we define convolutional filters that operate on those signals within domains defined by geodesic surface neighborhoods. This approach combines the advantages of feature extraction from high-resolution signals (as in [10]) with the advantages of view-independent convolution on 3D surface domains (as in [39]). This combination is important for the example in labeling the chair in Figure 1, whose surface fabric is easily recognizable in a color texture map.

During our investigation of this approach, we had to address several research issues, the most significant of which is how to define on geodesic neighborhoods of a mesh. One approach could be to compute a global UV parameterization for the entire surface and then define convolutional operators directly in UV space; however, that approach may induce significant deformations due to flattening, not always follow surface features, and/or produce seams at surface cuts. Another approach could be to compute UV parameterizations for local neighborhoods independently; however, then adjacent neighborhoods might not be oriented consistently, reducing the ability of a network to properly learn orientation-dependent features. Instead, we compute a 4-RoSy (four-fold rotationally symmetric) field on the surface using QuadriFlow [17] and define a new 4-RoSy convolutional operator that explicitly accounts for the 4-fold rotational ambiguity of the cross field parameterization. A 4-RoSy (four-way rotationally symmetric) field is a configuration of 4 orthogonal tangent directions associated with each vertex in the shape of a cross that varies smoothly over the mesh surface. Since the 4-RoSy field from QuadriFlow has no seams, aligns to shape features, induces relatively little distortion, has few singularities, and consistently orients adjacent neighborhoods (up to 4-way rotations), it provides an attractive trade-off between distortion and orientation invariance.

Results on 3D semantic segmentation benchmarks show an improvement of the 4-RoSy convolution on surfaces over alternative geometry-only approaches (by 6.4%), plus significantly further improvement when applied to high-resolution color signals (by 6.9-8.2%). With ablation studies, we verify the importance of the consistent orientation of a 4-RoSy field and demonstrate that our sampling and convolution operator works better than other alternatives.

Overall, our core research contributions are:

- a novel learning-based method for extracting features from high-resolution signals living on surfaces embedded in 3D, based on consistent local parameterizations,
- a new 4-RoSy convolutional operator designed for cross fields on general surfaces in 3D,
- a new deep network architecture, TextureNet, composed of 4-RoSy convolutional operators,
- an extensive experimental investigation of alternative convolutional operators for semantic segmentation of surfaces in 3D.

2. Related Work

3D Deep Learning. With the availability of 3D shape databases [42, 7, 36] and real-world labeled 3D scanning data [35, 1, 9, 6], there is significant interest in deep learning on three-dimensional data. Early work developed CNNs operating on 3D volumetric grids [42, 25]. They have been used for 3D shape classification [30, 33], semantic segmentation [9, 13], object completion [12], and scene completion [13]. More recently, researchers have developed methods that can take a 3D point cloud as input to a neural network and predict object classes or semantic point labels [29, 31, 39, 37, 2]. AtlasNet [14] learns to generate surfaces of the 3D shape. In our work, we utilize a sparse point sampled data representation, however, we exploit high resolution signals on geometric surface structures with a new 4-RoSy surface convolution kernel.

Convolutions on Meshes. Several researchers have proposed methods for applying convolutional neural networks intrinsically on manifold meshes. FeaStNet [40] proposes a graph operator that establishes correspondences between filter weights. Jiang *et al.* [20] applies differential operators on unstructured spherical grids. GCNN [24] proposes using discrete patch operators on tangent planes parameterized by radius and angles. However, the orientation of their selected geodesic patches is arbitrary, and the parameterization is highly distorted or inconsistent at regions with high Gaussian curvature. ACNN [3] observes this limitation and introduces the anisotropic heat kernels derived from principal curvatures. MoNet [26] further generalizes the architecture with the learnable gaussian kernels for convolutions. The principal curvature based frame selection method is adopted by Xu *et al.* [43] for segmentation of nonrigid surfaces, by Tatarchenko *et al.* [39] for semantic segmentation of point clouds, and by ADD [4] for shape correspondence in the spectral domain. It naturally removes orientation ambiguity but fails to consider frame inconsistency problem, which is critical when performing feature aggregation. Its problems are particularly pronounced in indoor scenes (which often have many planar regions where principal curvatures are undetermined) and in real-world scans (which often have noisy and uneven sampling where consistent principal curvatures are difficult to predict). In contrast, we define a 4-RoSy field that provides consistent orientations for neighboring convolution domains.

Multi-view and 2D-3D Joint Learning. Other researchers have investigated how to incorporate features from RGB inputs to 3D deep networks. The typical approach is to simply assign color values to voxels, points, or mesh vertices and treat them as additional feature channels. However, given that geometry and RGB data are at vastly different resolutions, this approach leads to significant downsampling of the color signal and thus does not

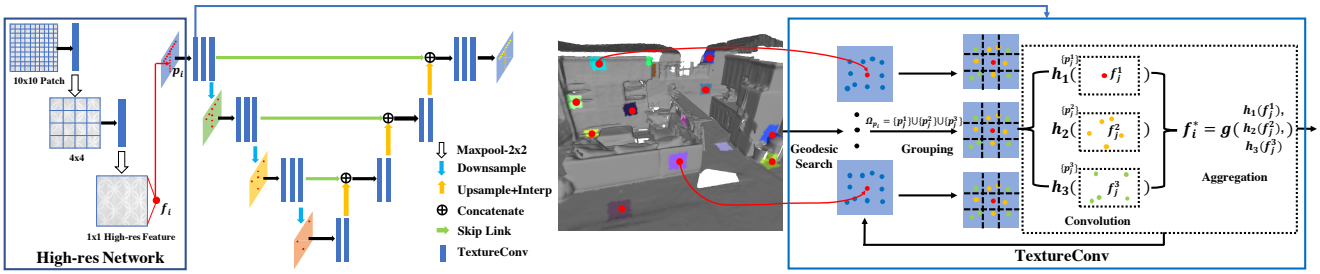


Figure 2: TextureNet architecture. We propose a UNet [34] architecture for hierarchical feature extraction. The key innovation in the architecture is the texture convolution layer. We efficiently query the local geodesic patch for each surface point, associating each neighborhood with a local, orientation-consistent texture coordinate system. This allows us to extract the local 3D surface features as well as high-resolution signals such as associated RGB input.

take full advantage of the high-frequency patterns therein. An alternative approach is to combine features extracted from RGB images in a multi-view CNN [38]. This approach has been used for 3D semantic segmentation in 3DMV [10], where features are extracted from 2D RGB images and then back-projected into a 3D voxel grid where they are merged and further processed with 3D voxel convolutions. Like our approach, 3DMV processes high-resolution RGB signals; however it convolves them in a 2D image plane, where occlusions and background clutter are confounding. In contrast, our method directly convolves high-resolution signals intrinsically on the 3D surface which is view-independent.

3. The TextureNet Approach

Our approach performs convolutions on high-resolution signals with geodesic convolutions directly on 3D surface meshes. The input is a 3D mesh associated with a high-resolution surface signal (e.g., a color texture map), and the outputs are learned features for a dense set of sample points that can be used for semantic segmentation and other tasks.

Our main contribution is defining a smooth, consistently oriented domain for surface convolutions based on four-way rotationally symmetric (4-RoSy) fields. We observe that 3D surfaces can be mapped with low-distortion to two-dimensional parameterizations anchored at dense sample points with locally consistent orientations and few singularities if we allow for a four-way ambiguity in the orientation at the sample points. We leverage that observation in TextureNet by computing a 4-RoSy field and point sampling using QuadriFlow [17] and then building a network using new 4-RoSy convolutional filters (TextureConv) that are invariant to the four-way rotational ambiguity.

We utilize this network design to learn and extract features from high-resolution signals on surfaces by extracting surface patches with high-resolution signals oriented by the 4-RoSy field at each sample point. The surface patches are convolved by a few TextureConv layers, pooled at sample points, and then convolved further with TextureConv layers in a UNet [34] architecture, as shown in figure 2. For down-sampling and up-sampling, we use the furthest point sampling and three-nearest neighbor interpolation method proposed by PointNet++ [31]. The output of the network is a set of features associated with point samples that can be used for classification and other tasks. The following sections describe the main components of the network in detail.

3.1. High-Resolution Signal Representation

Our network takes as input a high-resolution signal associated with a 3D surface mesh. In the first steps of processing, it generates a set of sample points on the mesh and defines a parameterized high-resolution patch for each sample (Section 3.2) as follows: For each sample point $\mathbf{p}_i$, we first compute its geodesic neighborhood $\Omega_\rho(\mathbf{p}_i)$ (Eq. 1) with radius ρ. Then, we sample an NxN point cloud $\{\mathbf{q}_{xy} | -N/2 \leq x, y < N/2\}$. The texture coordinates for $\mathbf{q}_{xy}$ are $((x+0.5)d, (y+0.5)d) - d$ is the distance between the adjacent pixels in the texture patch. In practice, we select $N = 10$ and $d = 4$mm. Finally, we use our newly proposed "TextureConv" and max-pooling operators (Section 3.3) to extract the high-res feature $\mathbf{f}_i$ for each point $\mathbf{p}_i$.

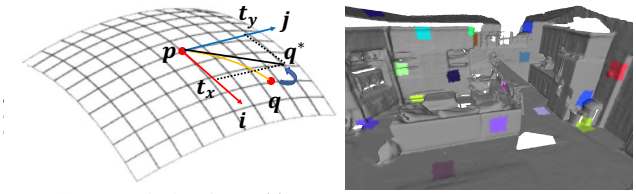
3.2. 4-RoSy Surface parameterization

A critical aspect of our network is to define a consistently-oriented geodesic surface parameterization for any position on a 3D mesh. Starting with some basic definitions, for a sampled point $\mathbf{p}$ on the surface, we can locally parameterize its tangent plane by two orthogonal tangent vectors $\mathbf{i}$ and $\mathbf{j}$. Also, for any point $\mathbf{q}$ on the surface, there exists a shortest path on the surface connecting $\mathbf{p}$ and $\mathbf{q}$, e.g., the orange path in figure 3(a). By unfolding it to the tangent plane, we can map $\mathbf{q}$ along the shortest path to $\mathbf{q}^*$. Using these constructs, we define the local texture coordinate $\mathbf{q}$ in $\mathbf{p}$'s neighborhood as

$$\mathbf{t}_\mathbf{p}(\mathbf{q}) = \begin{bmatrix}\mathbf{i}^T & \mathbf{j}^T\end{bmatrix} (\mathbf{q}^* - \mathbf{p}).$$

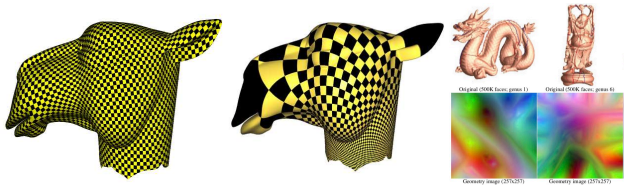
We additionally define the local geodesic neighborhood of $\mathbf{p}$ with receptive field ρ as

$$\Omega_\rho(\mathbf{p}) = \{\mathbf{q} \mid ||\mathbf{t}_\mathbf{p}(\mathbf{q})||_\infty < \rho\}. \quad (1)$$



(a) Parameterization of the $\Omega_\rho(p)$ (b) Visualization of the Geodesic Patches

Figure 3: (a) Local texture coordinates. (b) Visualization of geodesic neighborhoods Ω_ρ (ρ = 20 cm) on a set of randomly sampled vertices.

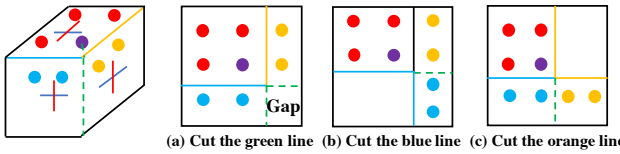


(a) QuadriFlow Parametrization (b) Harmonic Parametrization (c) Geometry Image

Figure 4: (a) With an appropriate method like Quadriflow, we can get the surface parameterization aligned with shape features with negligible distortion. (b) Harmonic parameterization leads to high distortion in the scale. (c) Geometry images [15] result in high distortion in the orientation.

The selection for the set of mesh sampled positions $\{\mathbf{p}\}$ and their tangent vectors $\mathbf{i}$ and $\mathbf{j}$ is critical for the success of learning on a surface domain. Ideally, we would select points whose spacing is uniform and whose tangent directions are consistently oriented at neighbors, such that the underlying parameterization has no distortions or seams, as shown in Figure 4(a). With those properties, we could learn convolutional operators with translation invariance exactly as we would for images. Unfortunately, these properties are only achievable if the surface is a flat plane. For a general 3D surface, we can only hope to select a set of point samples and tangent vectors that minimize deviations between spacings of points and distortions of local surface parameterizations. Figure 4(b) shows an example where harmonic surface parameterization introduces large-scale distortion – a 2D convolution would include a large receptive field at the nose but a small one at the neck. Figure 4(c) shows a geometry image [15] parameterization with high distortion in the orientation – convolutions on such a map would have randomly distorted and irregular receptive fields, making it difficult for a network to learn canonical features.

Unfortunately, a smoothly varying direction field on the surface is usually hard to obtain. According to the study of the direction field design [32, 22], the best-known approach to mitigate the distortion is to compute a four-way rotationally symmetric (*4-RoSy*) *orientation field*, which minimizes the deviation by incorporating directional ambiguity. Additionally, the orientation field needs a consistent definition among different geometries, and the most intuitive way is to make it align with the shape features like the principal curvatures. Fortunately, the extrinsic energy is used by [19, 17]



(a) Cut the green line (b) Cut the blue line (c) Cut the orange line

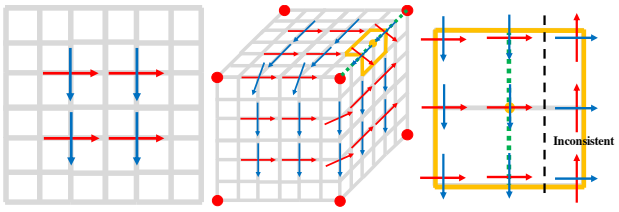
Figure 5: Singularity at a cube vertex, (a)-(c) demonstrate three different ways of unfolding the local neighborhood. Such ambiguity is removed around the singularity by our texture coordinate definition using the shortest path. For the purple point, (a) is a valid neighborhood, while the blue points in (b) and orange points in (c) are unfolded along the paths which are not the shortest. Similarly, the ambiguity in the gap location is removed.

to realize it. Therefore, we compute the extrinsic 4-Rosy orientation field at a uniform distribution of point samples using QuadriFlow [17] and use it to define the tangent vectors at any position on the surface. Because of the directional ambiguity, we randomly pick one direction from the cross as $\mathbf{i}$ and compute $\mathbf{j} = \mathbf{n} \times \mathbf{i}$ for any position.

Although there is a 4-way rotational ambiguity in this local parameterization of the surface (which will be addressed with a new convolutional operator in the next section), the resulting 4-RoSy field provides a way to extract geodesic neighborhoods consistently across the entire surface, even near singularities. Figure 5 (a,b,c) shows the ambiguity of possible unfolded neighborhoods at a singularity. Since QuadriFlow [17] treats singularities as faces rather than vertices, all sampled positions have the well-defined orientation field. More importantly, the parameterization of every geodesic neighborhood is well-defined with our shortest path patch parameterization. For example, only Figure 5(a) is a valid parameterization for the purple spot, while the location for the blue and orange spots in Figures 5(b) and (c) are unfolded along the paths that are not the shortest. Unfolding a geodesic neighborhood around the singularity also causes another potential issue that a seam cut is usually required, leading to a gap at the 3-singularity or multiple-surface coverage at the 5-singularity. For example, there is a gap at the bottom-right corner in Figure 5(a) caused by the seam cut shown as the green dot line. Fortunately, the location of the seam is also well-defined with our shortest-path definition: it must be the shortest geodesic path going through the singularity. Therefore, our definition of the local neighborhood guarantees a canonical way of surface parameterization even around corners and singularities.

3.3. 4-RoSy Surface Convolution Operator

TextureNet is a network architecture composed of convolutional operators acting on geodesic neighborhoods of sample points with 4-RoSy parameterizations. The input to each convolutional layer is three-fold: 1) a set of 3D sample points associated with features (e.g., RGB, normals, or features computed from high-resolution surface patches or pre-



(a) Image Coordinate (b) 3D parametrization (c) Inconsistent Frame

Figure 6: (a) Traditional convolution kernel on a regular grid. (b) Frames defined by the orientation field on a 3D cube. (c) For the patch highlighted in orange in (b), multi-layer feature aggregation would be problematic with traditional convolution due to the frame inconsistency caused by the directional ambiguity of the orientation field.

vious layers); 2) a coordinate system stored as two tangent vectors representing the 4-RoSy cross field for each point sample; and 3) a coarse triangle mesh, where each face is associated with the set of extracted sampled points and connectivity indices that support fast geodesic patch query and texture coordinate computation for the samples inside a geodesic neighborhood, much like the PTex [5] representation for textures.

Our key contribution in this section is the design of a convolution operator suitable for 4-RoSy fields. The problem is that we cannot use traditional 3x3 convolution kernels on domains parameterized with 4-RoSy fields without inducing inconsistent feature aggregation at higher levels. Figure 6 demonstrates the problem for a simple example. Figure 6(a) shows 3x3 convolution in a traditional flat domain. Figure 6(b) shows the frames defined by our 4-RoSy orientation field of the 3D cube where red spots represent the singularities. Although the cross-field in the orange patch is consistent under the 4-RoSy metric, the frames are not parallel when they are unfolded into a plane (figure 6(c)). Aggregation of features inside such a patch is therefore problematic.

"TextureConv" is our solution to remove the directional ambiguity. It consists of four layers (in figure 2), including geodesic patch search, texture space grouping, convolution and aggregation. To extract the geodesic patch for each input point $\Omega_\rho(\mathbf{p})$, we use breadth-first search with the priority queue to extract the face set in the order of geodesic distance from face center to $\mathbf{p}$. We estimate the texture coordinate at the face center as well as its local tangent coordinate system, recorded as $(\mathbf{t}_f, \mathbf{i}_f, \mathbf{j}_f)$. In order to expand the search tree from face u to v, we can approximate the texture coordinate at the face center as $\mathbf{t}_v = \mathbf{t}_u + (\mathbf{i}_u, \mathbf{j}_u)^T(\mathbf{c}_v - \mathbf{c}_u)$, where $\mathbf{c}_f$ represents the center position of the face f. $\mathbf{i}_v$ and $\mathbf{j}_v$ can be computed by rotating the coordinate system around the shared edge from face u to v. After having the face set inside the geodesic patch, we can find the sampled points set associated with these faces. We estimate the texture coordinate of every sampled point $\mathbf{q}$ associated with each face f as $\mathbf{t}_\mathbf{p}(\mathbf{q}) = \mathbf{t}_f + (\mathbf{i}_f, \mathbf{j}_f)^T(\mathbf{q} - \mathbf{c}_f)$. By testing $||\mathbf{t}_\mathbf{p}(\mathbf{q})||_\infty < \rho$, we can determine the sampled points inside the geodesic patch $\Omega_\rho(\mathbf{p})$.

The texture space grouping layer segments the local neighborhood into 3x3 patches in the texture space, each of which is a square with edge length as $2\rho/3$, as shown in figure 2 (after the "grouping arrow"). We could directly borrow the image convolution method linearly transform each point feature with 9 different weights according to their belonging patch. However, we propose a 4-RoSy convolution kernel to deal with the directional ambiguity. As shown in figure 2, all sampled points can be categorized as at the corners ($\{\mathbf{p}_j^1\}$), edges ($\{\mathbf{p}_j^2\}$) or the center ($\{\mathbf{p}_j^3\}$). Each sampled point feature is convolved with a 1x1 convolution as h_1, h_2 or h_3 based on its category. The extracted 4-rosy feature removes the ambiguity and allows higher-level feature aggregation. The channel-wise aggregation operator g can be max-pooling or average-pooling followed by the ReLu layer. In the task for semantic segmentation, we choose max-pooling since it is better at preserving salient signals.

4. TextureNet Evaluation

To investigate the performance of TextureNet, we ran a series of 3D semantic segmentation experiments for indoor scenes. In all experiments, we train and test on the standard splits of the ScanNet [9] and Matterport3D [9] datasets. Following previous works, we report mean class intersection-over-union (mIoU) results for ScanNet and mean class accuracy for Matterport3D.

Comparison to State-of-the-Art. Our main result is a comparison of TextureNet to state-of-the-art methods for 3D semantic segmentation. For this experiment, all methods utilize both color and geometry in their native formats. Specifically, PointNet++ [31], Tangent Convolution [39], SplatNet [37] use points with per-point normals and colors; 3DMV [10] uses 2D image features back-projected onto voxels; and Ours uses high-res 10x10 texture patches extracted from geodesic neighborhoods at sample points.

Table 1 reports the mean IoU scores for all 20 classes of the ScanNet benchmark on the ScanNet (v2) and mean class accuracy on Matterport3D datasets. They show that TextureNet (Ours) provides the best results on 18/20 classes for Scannet and 12/20 classes for Matterport3D. Overall, the mean class IoU for Ours is 8.2% higher than the previous state-of-the-art (3DMV) on ScanNet (48.4% vs. 56.6%), and our mean class accuracy is 6.9% higher on Matterport3D (56.1% vs. 63.0%).

Qualitative visual comparisons of the results shown in Figures 7-9 suggest that the differences between methods are often where high-resolution surface patterns are discriminating (e.g., the curtain and pillows in the top row of Figure 7) and where geodesic neighborhoods are more in-

Input	wall	floor	cab	bed	chair	sofa	table	door	wind	shf	pic	cntr	desk	curt	fridg	show	toil	sink	bath	other	avg
PN$^+$ [31]	66.4	91.5	27.8	56.3	64.0	52.7	37.3	28.3	36.1	59.2	6.7	28.0	26.2	45.4	25.6	22.0	63.5	38.8	54.4	20.0	42.5
SplatNet [37]	**69.9**	92.5	31.1	51.1	65.6	51.0	38.3	19.7	26.7	60.6	0.0	24.5	32.8	40.5	0.0	24.9	59.3	27.1	47.2	22.7	39.3
Tangent [39]	63.3	91.8	36.9	64.6	64.5	56.2	42.7	27.9	35.2	47.4	14.7	35.3	28.2	25.8	28.3	29.4	61.9	48.7	43.7	29.8	43.8
3DMV [10]	60.2	79.6	42.4	53.8	60.6	50.7	41.3	37.8	53.9	64.3	21.4	31.0	43.3	57.4	**53.7**	20.8	69.3	47.2	48.4	30.1	48.4
Ours	68.0	**93.5**	**49.4**	**66.4**	**71.9**	**63.6**	**46.4**	**39.6**	**56.8**	**67.1**	**22.5**	**44.5**	**41.1**	**67.8**	41.2	**53.5**	**79.4**	**56.5**	**67.2**	**35.6**	**56.6**

(a) ScanNet (v2) (mean class IoU)

Input	wall	floor	cab	bed	chair	sofa	table	door	wind	shf	pic	cntr	desk	curt	ceil	fridg	show	toil	sink	bath	other	avg
PN$^+$ [31]	80.1	81.3	34.1	71.8	59.7	63.5	**58.1**	49.6	28.7	1.1	34.3	10.1	0.0	68.8	79.3	0.0	29.0	70.4	29.4	62.1	8.5	43.8
SplatNet [37]	**90.8**	**95.7**	30.3	19.9	**77.6**	36.9	19.8	33.6	15.8	15.7	0.0	0.0	0.0	12.3	75.7	0.0	0.0	10.6	4.1	20.3	1.7	26.7
Tangent [39]	56.0	87.7	41.5	73.6	60.7	69.3	38.1	55.0	30.7	33.9	50.6	38.5	19.7	48.0	45.1	22.6	35.9	50.7	49.3	56.4	16.6	46.8
3DMV [10]	79.6	95.5	**59.7**	82.3	70.5	**73.3**	48.5	64.3	55.7	8.3	55.4	34.8	2.4	**80.1**	**94.8**	4.7	54.0	71.1	47.5	76.7	19.9	56.1
Ours	63.6	91.3	47.6	**82.4**	66.5	64.5	45.5	**69.4**	**60.9**	**30.5**	**77.0**	**42.3**	**44.3**	75.2	92.3	**49.1**	**66.0**	**80.1**	**60.6**	**86.4**	**27.5**	**63.0**

(b) Matterport3D (mean class accuracy)

Table 1: Comparison with the state-of-the-art methods for 3D semantic segmentation on the (a) ScanNet v2, and (b) Matterport3D [6] benchmarks. PN$^+$, SplatNet, and Tangent Convolution use points with per-point normal and color as input. 3DMV uses 2D images and voxels. Ours uses grid points with high-res 10x10 texture patches.

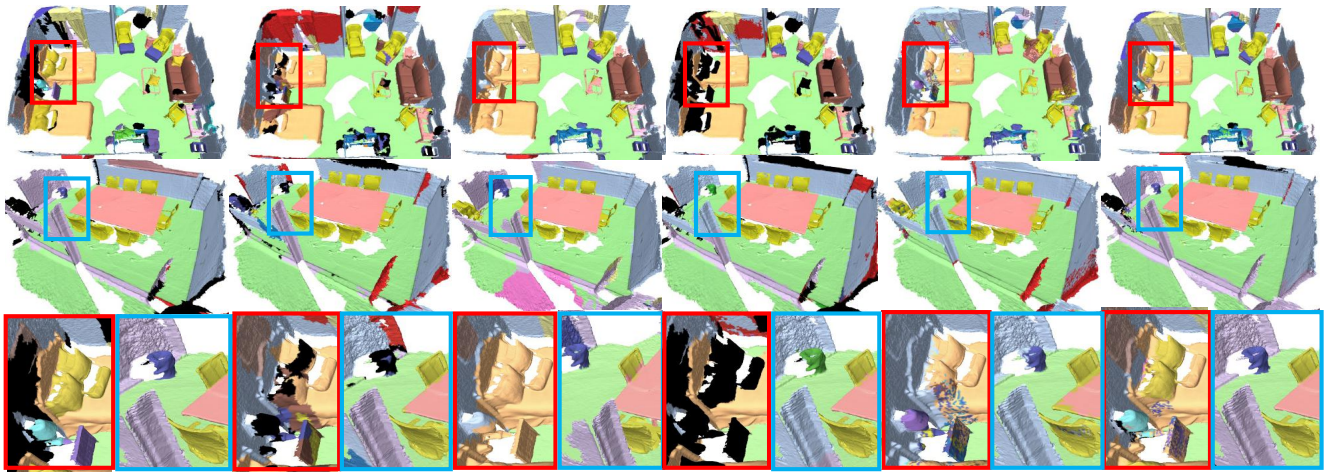
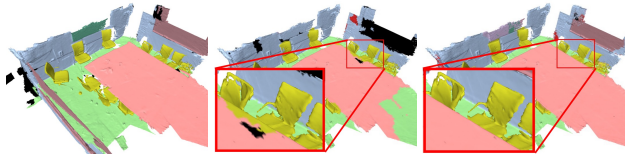


Figure 7: Visualization on ScanNet (v2) [9]. In the first row, we correctly predicts the lamp, pillow, picture, and part of the cabinet, while other methods fail. In the second row, we predict the window and the trash bin correctly, while 3DMV [10] predicts part of the window as the trash bin and other methods fail. The third row (zoom-in) highlights the differences.



(a) Ground Truth (b) Ball (c) Ours

Figure 8: Visual results using different neighborhoods. With euclidean ball as a neighborhood, part of the table is predicted as the chair, since they belong to the same euclidean ball. This issue is solved by extracting features from the geodesic patches.

formative than Euclidean ones (e.g., the lamp next to the bed). Figure 8 shows a case where convolutions with the geodesic neighborhoods clearly outperform their Euclidean counterparts. In Figure 8(b), part of the table is predicted as chair, probably because it is in a Euclidean ball covering nearby chairs. This problem is solved with our method based on geodesic patch neighborhoods. As shown in Figure 8(c), the table and the chairs are clearly segmented.

Effect of 4-RoSy Surface Parameterization. Our second experiment is designed to test how different surface parameterizations affect semantic segmentation performance – i.e., how does the choice of the orientation field affect the learning process? The simplest choice is to pick an arbitrary direction on the tangent plane as the x-axis, similar to GCNN [24], (Figure 10(a)). A second option adopted by Tangent Convolution [39] considers a set of points $\mathbf{q}$ in a Euclidean ball centered at $\mathbf{p}$ and parameterizes the tangent

Input	wall	floor	cab	bed	chair	sofa	table	door	wind	bkshf	pic	cntr	desk	curt	fridg	show	toil	sink	bath	other	ave
Random	37.6	**92.5**	37.0	63.7	28.5	56.9	27.6	15.3	31.0	47.6	16.5	36.6	**53.3**	**51.2**	15.4	24.7	59.3	47.6	53.3	27.0	41.1
Intrinsic	47.4	91.9	35.3	62.5	55.8	44.8	37.5	29.8	40.5	40.9	16.7	41.5	39.9	42.1	20.4	24.3	85.6	44.5	58.3	29.5	44.4
EigenVec	45.3	79.0	32.2	53.4	59.8	40.4	32.2	28.8	40.5	43.4	**17.8**	39.5	32.7	40.6	22.5	25.0	82.4	48.1	54.8	32.6	42.5
Extrinsic	**69.8**	92.3	**44.8**	**69.4**	**75.8**	**67.1**	**56.8**	**39.4**	**41.1**	**63.1**	15.8	**57.4**	46.5	48.3	**36.9**	**40.0**	**78.1**	**54.0**	**65.4**	**34.4**	**54.8**

Table 2: Mean IoU for different direction fields on ScanNet (v2). The input is a pointcloud with a normal and rgb color for each point. *Random* refers to randomly picking an arbitrary direction for each sampled point. *Intrinsic* refers to solving for a 4-rosy field with intrinsic energy. *EigenVec* refers to solving for a direction field with the principal curvature. *Extrinsic* is our method, which solves a 4-rosy field with extrinsic energy.

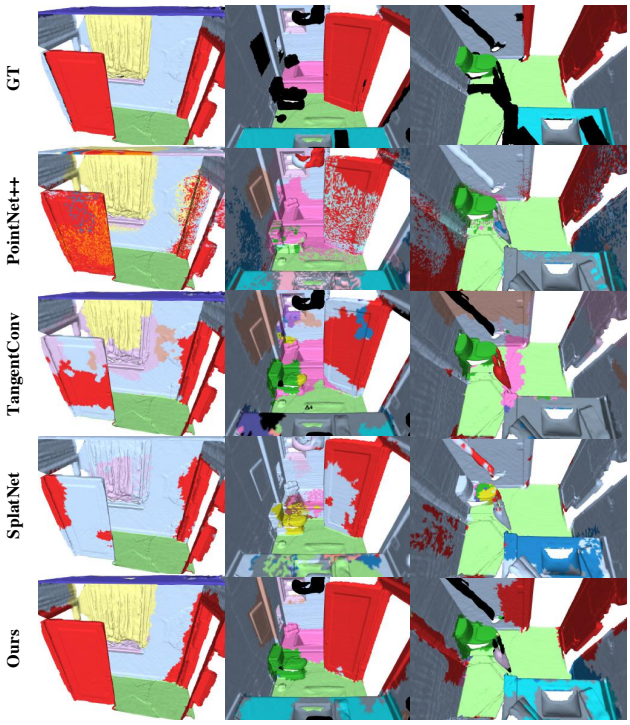
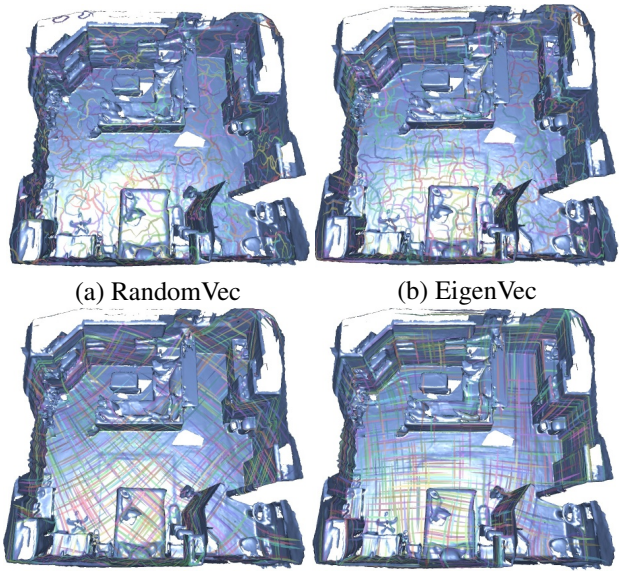


Figure 9: Visual results on Matterport3D [6]. In all examples, our method is better at predicting the door, the toilet, the sink, the bathtub, and the curtain.



(a) RandomVec (b) EigenVec

(c) Intrinsic (d) Extrinsic

Figure 10: Direction fields from different methods. (a) Random directions lead to inconsistent frames. (b) Eigenvectors suffer from the same issue at flat area. (c) Intrinsic-energy based orientation field does not align to the shape features. (d) Our extrinsic-based method generates consistent orientation fields aligned with surface features.

plane by two eigenvectors corresponding to the largest two eigenvalues of the covariance matrix $\sum_q (p-q)(p-q)^T$. A critical problem of this formulation is that the principal directions cannot be robustly analyzed at planar regions or noisy surfaces (Figure 10(b)). It also introduces inconsistency to the coordinate systems of the neighboring points, which vexes the feature aggregation at higher levels. A third alternative is to use the intrinsic energy function [19] or other widely used direction field synthesis technique [32, 22], which is not geometry-aware and therefore variant to 3D rigid transformation (Figure 10(c)). Our choice is to use the extrinsic energy to synthesize the direction field [17, 19], which is globally consistent and only variant to geometry itself (Figure 10(d)).

To test the impact of this choice, we compare all of these alternative direction fields to create the local neighborhood parameterizations for our architecture and compare the results of 3D semantic segmentation on ScanNet (v1) test set. As shown in Table 2, the choice for random direction field performs worst since it does not provide consistent parameterization. The tangent convolution suffers from the same issue, but gets a better result since it aligns with the shape features. The intrinsic parameterization aligns with the shape features, but is not a canonical parameterization – for example, different rigid transformations of the same shape lead to different parameterizations. The extrinsic energy provides a canonical and consistent surface parameterization. As a result, the extrinsic 4-rosy orientation field achieves the best results.

Effect of 4-RoSy Surface Convolution. Our third experiment is designed to test how the choice for the surface convolution operator affects learning. In Table 4, PN$^+$(A)

Input	wall	floor	cab	bed	chair	sofa	table	door	wind	bkshf	pic	cntr	desk	curt	fridg	show	toil	sink	bath	other	ave
XYZ	64.8	90.0	39.3	65.8	74.8	66.6	50.5	33.9	35.6	58.0	14.0	54.3	42.1	45.4	30.9	43.0	67.7	47.9	55.8	32.2	50.6
NRGB	69.8	92.3	44.8	**69.4**	75.8	**67.1**	56.8	39.4	41.1	63.1	15.8	**57.4**	46.5	48.3	**36.9**	40.0	78.1	**54.0**	65.4	34.4	54.8
Highres	**75.0**	**94.4**	**46.8**	67.3	**78.1**	64.0	**63.5**	**44.8**	**46.0**	**71.3**	**21.1**	44.4	**47.5**	**52.5**	35.2	**51.3**	**80.3**	51.7	**67.6**	**40.2**	**58.1**

Table 3: Mean IoU for different color inputs on ScanNet (v2). *XYZ* represents our network using raw point input; i.e., geometry only. *NRGB* represents our network taking input as the sampled points with per-point normal and color. *Highres* represents our network taking per-point normal and the 10x10 surface texture patch for each sampled point.

Input	$PN^+(A)$	PN^+	$GCNN^1$	GCNN	ACNN
Geometry	32.6	43.5	48.7	24.6	29.7
NRGB	38.1	48.2	49.6	27.0	32.4

Input	$RoSy^1$	$RoSy^4$	$RoSy^1$(m)	Ours(A)	Ours
Geometry	37.8	30.8	40.3	38.0	**50.6**
NRGB	47.8	34.5	42.6	39.1	**54.8**

Table 4: Mean Class IoU with different texture convolution operators on ScanNet (v2). The input is the pointcloud for the first row (Geometry) and the pointcloud associated with the normal and rgb signal for the second row (NRGB).

and PN^+ represent PointNet++ with average and max pooling, respectively. $GCNN^1$ and GCNN are geodesic convolutional neural networks [24] with $N_\rho = 3, N_\theta = 1$ and $N_\rho = N_\theta = 3$ respectively. ACNN represents anisotropic convolutional neural networks [3] with $N_\rho = 3, N_\theta = 1$. $RoSy^1$ means a 3x3 convolution along the direction of the 1-rosy orientation field. $RoSy^4$ picks an arbitrary direction from the cross in the 4-rosy field. $RoSy^4$(m) applies 3x3 convolution for each direction of the cross in the 4-rosy field, aggregated by max pooling. Ours(A) and Ours represent our method with average and max pooling aggregation.

We find that GCNN, ACNN and $RoSy^4$ produce the lowest IoUs, because they suffer from inconsistency of frames when features are aggregated. $GCNN^1$ does not suffer from this issue since there is only a single bin in the angle dimension. $RoSy^4$(m) uses the max-pooling to canonicalize the feature extraction, which is independent of the orientation selection, and produces better results than $RoSy^4$. $RoSy^1$ achieves a higher score by generating a more globally consistent orientation field with higher distortion. From this study, the combination of 4-rosy orientation field and our TextureNet is the best option for the segmentation task among these methods. Since we precompute the local parametrization, our training efficiency is similar to GCNN.

Effect of High-Resolution Color. Our fourth experiment tests how much convolving with high-resolution surface colors affects semantic segmentation. Table 3 compares the performance of our network with uncolored sampled points (XYZ), sampled points with the per-point surface normal and color (NRGB), and with the per-point normal and the 10x10 color texture patch (Highres) as input. According to Table 4, our network is already superior with only XYZ or additional NRGB because of the convolution operator. We find that providing TextureNet with Highres colors improves the mean class IoU by 3.3%. As expected, the impact is stronger from some semantic classes than others – e.g., the IoUs for the bookshelf and picture classes increase 63.1→71.3% and 15.8→21.1%, respectively.

Comparisons Using Only Surface Geometry. As a final experiment, we evaluate the value of the proposed 3D network for semantic segmentation of inputs with only surface geometry (without color). During experiments on ScanNet, TextureNet achieves 50.6% mIoU, which is 6.4% better than the previous state-of-the-art. In comparison, ScanNet [9] = 30.6%, Tangent Convolution [39] = 40.9%, PointNet++ [31] = 43.5%, and SplatNet [37] = 44.2%.

5. Conclusion

TextureNet bridges the gap between 2D image convolutions and 3D deep learning using 4-RoSy surface parameterizations. We have demonstarted a new method for learning from high-resolution signals on 3D meshes by computing local geodesic neighborhoods with consistent 4-RoSy coordinate systems. We have designed a network of 4-RoSy texture convolution operators that are able to learn surface features that significantly improve over the state-of-the-art performance for 3D semantic segmentation of 3D surfaces with color (by 6.9-8.2%). Code and data will be publicly available. Topics for further work include investigating the utility of TextureNet for extracting features from other high-resolution signals on meshes (e.g., displacement maps, bump maps, curvature maps, etc.) and applications of TextureNet to other vision tasks (e.g., instance detection, pose estimation, part decomposition, texture synthesis, etc.).

Acknowledgements

This work is supported in part by Google, Intel, Amazon, a Vannevar Bush faculty fellowship, a TUM Foundation Fellowship, a TUM-IAS Rudolf Mößbauer Fellowship, a ERC Starting Grant *Scan2CAD*, a Hans Fischer IAS-TUM Senior Fellowship, and NSF grants VEC-1539014/1539099, CHS-1528025 and IIS-1763268. It makes use of data from Matterport. .

References

[1] I. Armeni, S. Sax, A. R. Zamir, and S. Savarese. Joint 2d-3d-semantic data for indoor scene understanding. *arXiv preprint arXiv:1702.01105*, 2017.

[2] M. Atzmon, H. Maron, and Y. Lipman. Point convolutional neural networks by extension operators. *arXiv preprint arXiv:1803.10091*, 2018.

[3] D. Boscaini, J. Masci, E. Rodolà, and M. Bronstein. Learning shape correspondence with anisotropic convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 3189–3197, 2016.

[4] D. Boscaini, J. Masci, E. Rodolà, M. M. Bronstein, and D. Cremers. Anisotropic diffusion descriptors. In *Computer Graphics Forum*, volume 35, pages 431–441. Wiley Online Library, 2016.

[5] B. Burley and D. Lacewell. Ptex: Per-face texture mapping for production rendering. In *Computer Graphics Forum*, volume 27, pages 1155–1164. Wiley Online Library, 2008.

[6] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Nießner, M. Savva, S. Song, A. Zeng, and Y. Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017.

[7] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[8] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312. ACM, 1996.

[9] A. Dai, A. X. Chang, M. Savva, M. Halber, T. A. Funkhouser, and M. Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, volume 2, page 10, 2017.

[10] A. Dai and M. Nießner. 3dmv: Joint 3d-multi-view prediction for 3d semantic scene segmentation. *arXiv preprint arXiv:1803.10409*, 2018.

[11] A. Dai, M. Nießner, M. Zollhöfer, S. Izadi, and C. Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics (TOG)*, 36(4):76a, 2017.

[12] A. Dai, C. R. Qi, and M. Nießner. Shape completion using 3d-encoder-predictor cnns and shape synthesis. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 3, 2017.

[13] A. Dai, D. Ritchie, M. Bokeloh, S. Reed, J. Sturm, and M. Nießner. Scancomplete: Large-scale scene completion and semantic segmentation for 3d scans. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2018.

[14] T. Groueix, M. Fisher, V. G. Kim, B. C. Russell, and M. Aubry. A papier-mâché approach to learning 3d surface generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 216–224, 2018.

[15] X. Gu, S. J. Gortler, and H. Hoppe. Geometry images. *ACM Transactions on Graphics (TOG)*, 21(3):355–361, 2002.

[16] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE, 2017.

[17] J. Huang, Y. Zhou, M. Nießner, J. R. Shewchuk, and L. J. Guibas. Quadriflow: A scalable and robust method for quadrangulation. In *Computer Graphics Forum*, volume 37, pages 147–160. Wiley Online Library, 2018.

[18] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568. ACM, 2011.

[19] W. Jakob, M. Tarini, D. Panozzo, and O. Sorkine-Hornung. Instant field-aligned meshes. *ACM Transactions on Graphics*, 34(6):189:1–189:15, Oct. 2015.

[20] C. Jiang, J. Huang, K. Kashinath, P. Marcus, M. Niessner, et al. Spherical cnns on unstructured grids. *arXiv preprint arXiv:1901.02039*, 2019.

[21] O. Kähler, V. A. Prisacariu, C. Y. Ren, X. Sun, P. Torr, and D. Murray. Very high frame rate volumetric integration of depth images on mobile devices. *IEEE transactions on visualization and computer graphics*, 21(11):1241–1250, 2015.

[22] Y.-K. Lai, M. Jin, X. Xie, Y. He, J. Palacios, E. Zhang, S.-M. Hu, and X. Gu. Metric-driven RoSy field design and remeshing. *IEEE Transactions on Visualization and Computer Graphics*, 16(1):95–108, 2010.

[23] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[24] J. Masci, D. Boscaini, M. Bronstein, and P. Vandergheynst. Geodesic convolutional neural networks on riemannian manifolds. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 37–45, 2015.

[25] D. Maturana and S. Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 922–928. IEEE, 2015.

[26] F. Monti, D. Boscaini, J. Masci, E. Rodola, J. Svoboda, and M. M. Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proc. CVPR*, volume 1, page 3, 2017.

[27] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, pages 127–136. IEEE, 2011.

[28] M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger. Real-time 3d reconstruction at scale using voxel hashing. *ACM Transactions on Graphics (ToG)*, 32(6):169, 2013.

[29] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 1(2):4, 2017.

[30] C. R. Qi, H. Su, M. Nießner, A. Dai, M. Yan, and L. Guibas. Volumetric and multi-view cnns for object classification on

3d data. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2016.

[31] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, pages 5099–5108, 2017.

[32] N. Ray, B. Vallet, W. C. Li, and B. Lévy. n-symmetry direction field design. *ACM Transactions on Graphics (TOG)*, 27(2):10, 2008.

[33] G. Riegler, A. O. Ulusoy, and A. Geiger. Octnet: Learning deep 3d representations at high resolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 3, 2017.

[34] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[35] S. Song, S. P. Lichtenberg, and J. Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 567–576, 2015.

[36] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser. Semantic scene completion from a single depth image. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 190–198. IEEE, 2017.

[37] H. Su, V. Jampani, D. Sun, S. Maji, E. Kalogerakis, M.-H. Yang, and J. Kautz. Splatnet: Sparse lattice networks for point cloud processing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2530–2539, 2018.

[38] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 945–953, 2015.

[39] M. Tatarchenko, J. Park, V. Koltun, and Q.-Y. Zhou. Tangent convolutions for dense prediction in 3d. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3887–3896, 2018.

[40] N. Verma, E. Boyer, and J. Verbeek. Feastnet: Feature-steered graph convolutions for 3d shape analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2598–2606, 2018.

[41] T. Whelan, R. F. Salas-Moreno, B. Glocker, A. J. Davison, and S. Leutenegger. Elasticfusion: Real-time dense slam and light source estimation. *The International Journal of Robotics Research*, 35(14):1697–1716, 2016.

[42] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015.

[43] H. Xu, M. Dong, and Z. Zhong. Directionally convolutional networks for 3d shape segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2698–2707, 2017.