Geodesic-HOF: 3D Reconstruction Without Cutting Corners

Ziyun Wang, Eric A. Mitchell, Volkan Isler, Daniel D. Lee
Samsung AI Center, New York, NY 10011
saicny@samsung.com

Abstract

Single-view 3D object reconstruction is a challenging fundamental problem in machine perception, largely due to the morphological diversity of objects in the natural world. In particular, high curvature regions are not always represented accurately by methods trained with common set-based loss functions such as Chamfer Distance, resulting in reconstructions short-circuiting the surface or "cutting corners." To address this issue, we propose an approach to 3D reconstruction that embeds points on the surface of an object into a higher-dimensional space that captures both the original 3D surface as well as geodesic distances between points on the surface of the object. The precise specification of these additional "lifted" coordinates ultimately yields useful surface information without requiring excessive additional computation during either training or testing, in comparison with existing approaches. Our experiments show that taking advantage of these learned lifted coordinates yields better performance for estimating surface normals and generating surfaces than using point cloud reconstructions alone. Further, we find that this learned geodesic embedding space provides useful information for applications such as unsupervised object decomposition.

Introduction

Reconstructing the 3D model of an object from a single image is a central problem in computer vision, with many applications. For example, in computer graphics, surface models such as triangle meshes are used for computing object appearance. In robotics, surface normal vectors are used for grasp planning, and in computer aided design and manufacturing, complete object models are needed for production. Motivated by such use cases, recent deep learning-based approaches to 3D reconstruction have shown exciting progress by using powerful function approximators to represent a mapping from the space of images to the space of 3D geometries. It has been shown that deep learning architectures, once trained on large datasets such as ShapeNet (Chang et al. 2015), are capable of outputting accurate object models in various representations including discrete voxelizations, unordered point sets, or implicit functions.

These representations can generally produce aesthetically pleasing reconstructions, but extracting topological information from them, such as computing neighborhoods of a point on the surface, can be difficult. In particular, naively computing a neighborhood of a point using a Euclidean ball or Euclidean nearest neighbors can give incorrect results if the object has regions of high curvature. Such mistakes can in turn degrade performance on downstream tasks such as unsupervised part detection or surface normal estimation. We thus propose a new method for single-view 3D reconstruction based on the idea of explicitly learning surface geodesics.

The key insight of our approach is to embed points sampled from the surface of a 3-dimensional object into a higher-dimensional space such that geodesic distances on the surface of the object can be computed as Euclidean distance in this 'lifted' space. This embedding space is constrained such that the 3D surface of the object lies in its first 3 dimensions (Figure 1-left shows a 2D example of this constraint). The remaining embedding dimensions can thus be interpreted as a quantification of curvature. To test the efficacy of this approach, we present a neural network architecture and training regime that is able to effectively learn this representation, and our experiments demonstrate that using the learned surface geodesics yields meaningful improvements in surface normal estimation. Further, we find that this embedding enables unsupervised decomposition of an object's surface into non-overlapping 2D sub-manifolds (i.e., charts), which can be useful for texture mapping and surface triangulation (see Figure 1-right).

Our contributions: We present Geodesic Higher Order Function Networks (Geodesic-HOF) for surface generation and provide several experimental evaluations to assess its performance. In the Single View Reconstruction section, we show that Geodesic-HOF exhibits reconstruction quality on par with state-of-the-art surface reconstruction methods. Further, we find that using the learned surface geodesics meaningfully improves surface normal estimation. Finally, we show how the learned representation can be used for decomposing the object surface into non-overlapping charts which can be used for generating triangulation or explicit function representations of surfaces. We hope that our results will provide an effective method for reconstructing surfaces and unfold new research directions for incorporating higher-order surface properties for shape reconstruction.

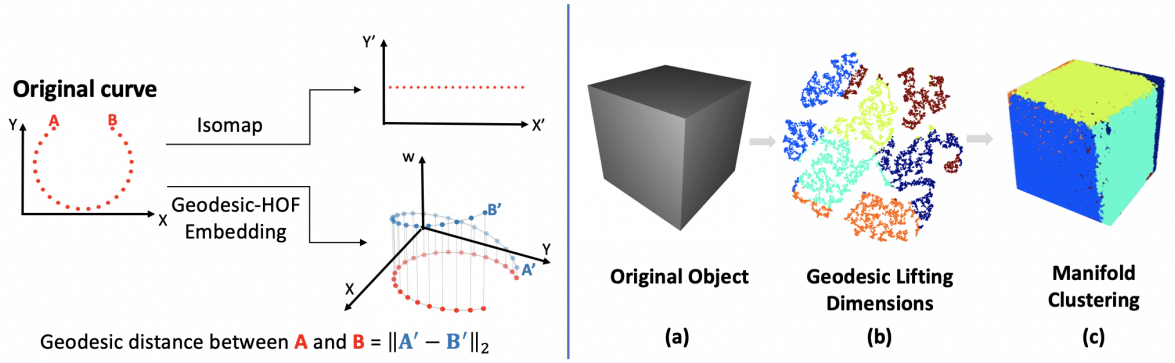

Figure 1: Left: Embedding a cut circle with Isomap (Tenenbaum, De Silva, and Langford 2000) vs. Geodesic-HOF. In contrast to embedding methods such as Isomap, Geodesic-HOF preserves the original Euclidean coordinates and lifts the points to the higher-dimensional space by adding extra "geodesic lifting coordinates". Right: Unsupervised Face Decomposition using the (9-dimensional) geodesic lifting coordinates. Figure (a) shows the object of interest. Figure (b) shows a low-dimensional projection of the geodesic lifting dimensions using t-SNE (Maaten and Hinton 2008). Figure (c) shows the same clustering results by coloring the output 3D points. For clarity of correspondence, we retrospectively color the points in (b) to match the clustering labels in (c). Best viewed in color.

Related Work

Several recently-developed object representations have shown promise in the 3D reconstruction literature. With the success of Convolutional Neural Network (CNN) in processing image data, it is natural to extend it to 3D. Therefore, many methods have been developed to directly output a regular voxel grid in 3D (Choy et al. 2016a; Tatarchenko, Dosovitskiy, and Brox 2017; Riegler, Osman Ulusoy, and Geiger 2017; Wu et al. 2016). However, the naive voxel representation requires the output to be the same resolution during training and during inference and the computational and memory usage of such methods grows cubically with the resolution. More complex octree-style approaches have been proposed to address these issues, but scaling to high resolutions remains a challenge (Riegler, Osman Ulusoy, and Geiger 2017).

In light of these resource demands, unordered point sets have become a popular alternative to voxelization for representing 3D shapes (Fan et al. 2017; Yang et al. 2018; Lin, Kong, and Lucey 2018; Mitchell et al. 2019), as the inherent sparsity of point sets scale more gracefully to high-resolution reconstructions. However, unordered point sets still lack intrinsic information about the topology of the underlying surface; thus, some work has investigated the use of implicit functional surface representations. Implicit functions such as occupancy (Mescheder et al. 2019) or signed distance (Park et al. 2019), which are continuous by definition, have shown promise as object representations, and can effectively store detailed information about an object's geometry. One of the main drawbacks of such methods is the need for a post-processing step such as running marching cubes to generate the object, making extracting the underlying object extremely time-consuming (Park et al. 2019). Furthermore, generating training data for these methods is non-trivial since they require dense samples near the surface of the object. Finally, as noted in (Mescheder et al. 2019), in terms of Chamfer score, implicit function methods are not as accurate as *direct methods* such as AtlasNet (Groueix et al. 2018), Pixel2Mesh (Wang et al. 2018), Mesh R-CNN (Gkioxari, Malik, and Johnson 2019) and GEOMetrics (Smith et al. 2019) which are trained directly using Chamfer-based loss functions.

A substantial body of work exists at the intersection of differential geometry, computer vision, and deep learning that studies the usefulness of geodesics in many 3D settings, such as surface matching (Wang, Peterson, and Staib 2000; Wang, Peterson, and Staib 2003), shape classification (Luciano and Hamza 2018), and manifold learning (Pai et al. 2019; Groueix et al. 2018; Wang et al. 2018). In particular, the well-known Isomap (Tenenbaum, De Silva, and Langford 2000) algorithm uses shortest paths on k-nearest neighbors graphs and applies Multidimensional Scaling (Cox and Cox 2008) to find the dimensionality and embedding of the data. As illustrated in Figure 1, direct embedding of geodesic distances does not necessarily yield the object surface. In Geodesic-HOF, the network is designed to explicitly output a sampling of the surface manifold and learn geodesic distances.

Learning Geodesics for 3D Reconstruction

From a finite set of points, connecting points based on Euclidean distance proximity alone is insufficient to produce an accurate depiction of the surface topology. If distant points on the manifold are erroneously considered to be close because they are close in the Euclidean space used for computing neighborhoods, the so-called "short-circuiting" problem arises (Balasubramanian and Schwartz 2002). Short-circuiting can be observed in Figure 2 where the points on opposite sides of a single wing are erroneously connected because they are nearby in terms of their Euclidean distance, although they are quite far on the surface. We propose using surface geodesics as a natural tool to solve this problem.

A *geodesic* between two points on a surface is a shortest

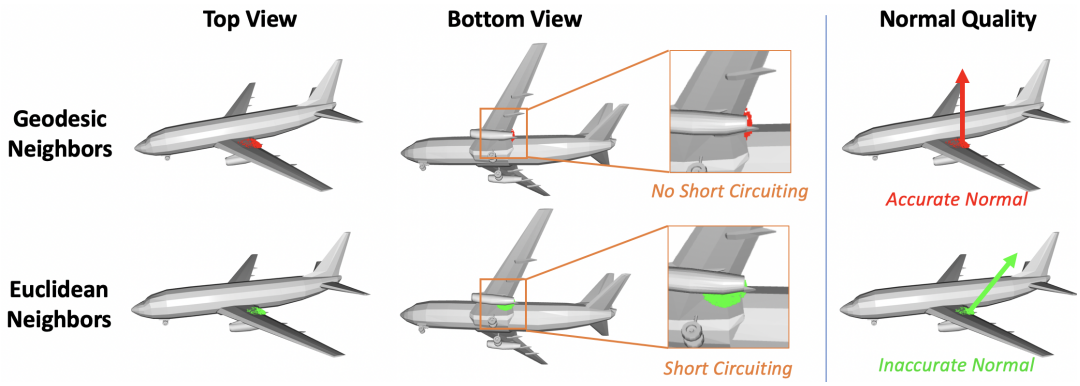

Figure 2: Top: Estimating the normal of a point on the wing by using geodesic neighborhoods. Bottom: Normal estimation of the same point using the Euclidean neighborhoods. We observe spurious neighbor points from the bottom of the wing are included in the Euclidean case, which causes the normal estimation to be inaccurate. Best viewed in color.

path between these points which lies entirely on the surface[1]. Geodesics carry a significant amount of information about surface properties. In particular, the difference between the geodesic and Euclidean distances between two points is directly related to curvature. Intuitively, connecting points based on geodesic distance, rather than Euclidean distance yields a more faithful reconstruction of the true surface. Given this setup, we can formalize the surface reconstruction problem that is the focus of this work.

Problem statement: Given a single image I of an object O, our goal is to be able to (i) generate an arbitrary number of samples from the surface of O, and (ii) compute the geodesic distance between any two generated points on the surface. We use Chamfer distance (Eqn. 1) to quantify the similarity between a set of samples from the model and the ground truth O.

In the next section, we present our approach to solving this problem through a deep neural network.

Geodesic-HOF: Method Overview

Geodesic-HOF is a neural network architecture and set of training objectives that aim to address the problem presented above. In order to describe the technical details of Geodesic-HOF, we first establish our notation and terminology. During training, we are given an image I of an object O as well as $X^* = \{x_1^*, \ldots, x_n^*\}$, a set of points sampled from O. We denote the ground truth surface geodesic function $g(\cdot, \cdot)$. In practice, the function g can be computed using either a very dense sampling of the ground truth object O or a surface triangulation. Using a convolutional neural network, Geodesic-HOF maps an input image I to a continuous mapping function $f_I : M \subset \mathbb{R}^D \rightarrow \mathbb{R}^{3+K}$. f_I maps samples from a canonical sampling domain M (in this work, the unit solid sphere) to an embedding space $Z = \{z_i = f_I(m_i) \in \mathbb{R}^{3+K}\}$. Applying f_I to M gives a set of high-dimensional points from which we extract an object reconstruction, including surface curvature information.

For every $z \in Z$, we denote the vector obtained by taking the first three components of z as x and refer to it as *point coordinates* of z. We call the remaining K dimensions as the *geodesic lifting coordinates* of z. We define the set of predicted point coordinates from a set of samples $\{m_i\}$ as $\hat{O} = \{\hat{x}_i = f_I(m_i)[:3]\}$ (the first three components of each output of f_I). Finally, for any two $m_i, m_j \in M$, with $\hat{z}_i = f_I(m_i)$ and $\hat{z}_j = f_I(m_j)$, the predicted geodesic distance is given by $\hat{g}(\hat{z}_i, \hat{z}_j) = ||\hat{z}_i - \hat{z}_j||_2$.

The mapping f_I is trained such that $\hat{O}$ accurately approximates the ground truth object O in terms of Chamfer distance and $\hat{g}(z_i, z_j)$ accurately approximates the ground truth geodesic distance $g(x_i^*, x_j^*)$, where $x_i^*, x_j^* \in X^*$ are the two ground truth samples closest to $\hat{x}_i$ and $\hat{x}_j$ (the point coordinates of z_i and z_j). The first condition requires that the point coordinates represent an accurate sampling of the object surface; the second condition requires the embedding space to accurately capture geodesic distances between two point coordinates. We can show that the lifting coordinates encode curvature information: the quantity $\hat{g}(\hat{z}_i, \hat{z}_j)^2 - ||\hat{x}_i - \hat{x}_j||^2$ is equal to the squared norm of the geodesic lifting coordinates. This quantity approaches the geodesic curvature when the samples are close to each other on the manifold.

Network Architecture and Training

The network design of Geodesic-HOF follows the **Higher Order Function (HOF)** method, which has three main components: an image encoder, a Higher-Order Function network and a point mapping network. An image encoder extracts the semantic and geometric information from an image I of an object. From this representation, the Higher-Order Function network predicts a set of parameters of a mapping function $f_I : \mathbb{R}^D \rightarrow \mathbb{R}^{3+K}$. To use the mapping function f_I, we start by sampling the unit sphere to generate a set of points $M = \{m_i\}$. Then we use the learned network to map these sampled points to a set of embeddings

[1] In some contexts, geodesics are defined as locally shortest paths. For example, two points on the sphere making an angle less than π with the center has two geodesics even though one is shorter than the other. In this paper, we use the term to refer to a globally shortest path. Note that, there might still be multiple geodesic as in the case of two diametrically opposite points on the sphere.

Category	3D-R2N2 (Choy et al.)	PSGN (Fan et al.)	Pix2Mesh (Wang et al.)	AtlasNet (Groueix et al.)	OccNet (Mescheder et al.)	**Ours**
Airplane	0.227	0.137	0.187	0.104	0.147	**0.099**
Bench	0.194	0.181	0.201	0.138	0.155	**0.122**
Cabinet	0.217	0.215	0.196	0.175	0.167	**0.134**
Car	0.213	0.169	0.180	0.141	0.105	**0.100**
Chair	0.270	0.247	0.265	0.209	0.180	**0.173**
Display	0.314	0.284	0.239	0.198	0.278	**0.193**
Lamp	0.778	0.314	0.308	0.305	0.479	**0.229**
Speaker	0.318	0.316	0.285	0.245	0.300	**0.206**
Rifle	0.183	0.134	0.164	0.115	0.141	**0.096**
Sofa	0.229	0.224	0.212	0.177	0.194	**0.162**
Table	0.239	0.222	0.218	0.190	0.189	**0.145**
Telephone	0.195	0.161	0.149	0.128	0.140	**0.109**
Vessel	0.238	0.188	0.212	0.151	0.218	**0.137**
mean	0.278	0.215	0.216	0.175	0.215	**0.141**

Table 1: Chamfer Comparison: Geodesic-HOF achieves state of the art performance in Chamfer distance. We sample 100,000 points on the object of interest and the output of each method to compute the Chamfer distance.

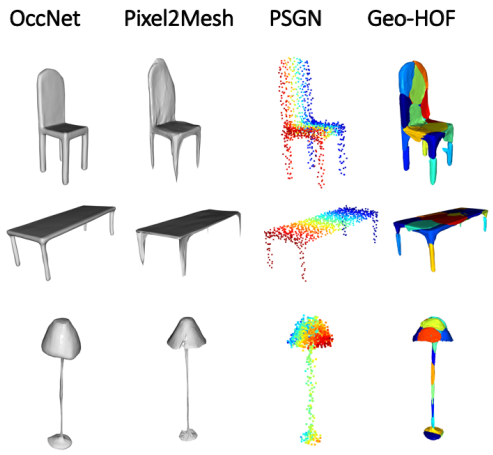

Figure 3: Qualitative Comparison of reconstruction results with existing methods. Best viewed in color.

$Z = \{z_i = f_I(m_i) \in \mathbb{R}^{3+K}\}$. The advantages of Higher-Order Function Networks over the Latent Vector Concatenation (LVC) paradigm are discussed in detail in (Mitchell et al. 2019). Please refer to the Supplementary Material Section for the architecture details. In Geodesic-HOF, we optimize the loss function L, the weighted sum of the standard Chamfer loss L_C and the geodesic loss L_G, with weight λ_C and λ_G, respectively: $L := \lambda_C L_C + \lambda_G L_G$. These losses are defined in the next section. Since the precision of the point coordinates is important for finding the correct geodesic distance, we weigh the Chamfer loss more than the Geodesic loss. Practically, we choose λ_G and λ_C to be 0.1 and 1.0 respectively. We use the Adam Optimizer (Kingma and Ba 2015) with learning rate 1e-5.

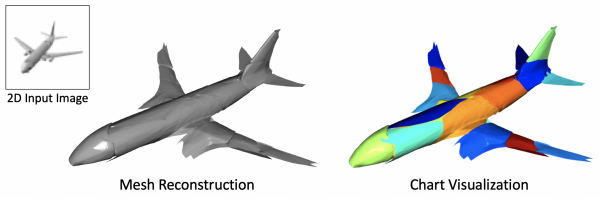

Figure 4: Top Left: 2D Input image. Left: Mesh Reconstruction from Geodesic-HOF by fitting a chart onto each non-overlapping manifold generated by clustering the geodesic lifting coordinates. Right: Visualization of the individual charts from our decomposition. Each color indicates a different chart (20 in total). Best viewed in color.

Loss Functions

The weights of the mapping function f_I for a given image I are learned so as to minimize the weighted sum of two losses: *Chamfer loss* and *Geodesic loss*. Recall M is a set of points randomly sampled from the pre-mapping space. The predicted set of embeddings from M is $Z = \{z_i = f_I(m_i) \mid m_i \in M\}$.

Chamfer loss is defined as the set distance between the point coordinates $X = \{x_i\}$ and the ground truth point set Y.

$$L_C(X,Y) = \frac{1}{|X|}\sum_{x\in X}\min_{y\in Y}||x-y||_2^2 + \frac{1}{|Y|}\sum_{y\in Y}\min_{x\in X}||y-x||_2^2 \quad (1)$$

We optimize this loss so that our predicted point set accurately represents the surface of the ground truth object.

Geodesic loss ensures the geodesic distance is learned accurately between every pair of points. We denote the ground truth geodesic distance on the object surface as $g(x_i, x_j)$, where $(x_i, x_j) \in \hat{O}$ and $\hat{O}$ is our prediction of O, which is the union of several 2-manifolds representing the object surface in $\mathbb{R}^3$. Since the geodesic distance is only defined on

Category	3D-R2N2 (Choy et al.)	Pix2Mesh (Wang et al.)	AtlasNet (Groueix et al.)	OccNet (Mescheder et al.)	Ours (Euc)	Ours (Geo)
Airplane	0.629	0.759	0.836	0.840	0.846	**0.863**
Bench	0.678	0.732	0.779	**0.813**	0.778	0.795
Cabinet	0.782	0.834	0.850	**0.879**	0.871	0.870
Car	0.714	0.756	0.836	**0.852**	0.819	0.827
Chair	0.663	0.746	0.791	**0.823**	0.802	0.810
Display	0.720	0.830	0.858	0.854	0.834	**0.862**
Lamp	0.560	0.666	0.694	0.731	0.712	**0.732**
Speaker	0.711	0.782	0.825	0.832	0.828	**0.838**
Rifle	0.670	0.718	0.725	0.766	**0.797**	**0.797**
Sofa	0.731	0.820	0.840	**0.863**	0.853	0.855
Table	0.732	0.784	0.832	**0.858**	0.827	0.841
Telephone	0.817	0.907	0.923	**0.935**	0.922	0.933
Vessel	0.629	0.699	0.756	**0.794**	0.774	0.790
mean	0.695	0.772	0.811	**0.834**	0.818	0.828

Table 2: Normal Comparison: Geodesic-HOF achieves competitive performance with state of the art methods in normal consistency. We sample 100,000 points on the object of interest and the output of each method to compute the normal consistency. Note that OccNet (Mescheder et al. 2019) is an implicit method, which gives it advantages in normal estimation due to marching cubes post-processing.

the object surface, we project the point coordinates $\{x_i\}$ onto the object and compute the pair-wise geodesic distance. The details of this formulation are described in Equation 6. We want to reconstruct the object while learning the embedding of each point so that the geodesic distance in the 3D object space $\mathbb{R}^3$ is the same as the Euclidean distance in the embedding space $\mathbb{R}^k$. For a set of embeddings $Z = \{z_i := [x_i, w_i]\}$, the geodesic loss is defined as:

$$L_{\mathrm{G}}(Z) = \frac{1}{|Z|^2} \sum_{i=1}^{|Z|} \sum_{j=1}^{|Z|} (||z_i - z_j||_2 - g(x_i, x_j))^2 \quad (2)$$

For computing the geodesic loss, we need the ground truth geodesic distance matrix on the object O. We build a Nearest Neighborhood graph $G = (V, A)$ on X^*, a set of samples from O. We define $D(v_i, v_j)$, the geodesic distance between v_i and v_j, as the length of the shortest path between $v_i \in V$ and $v_j \in V$ computed by Dijkstra's algorithm on G. For each point x_i from our prediction, we find its k-nearest neighbors, denoted as $\Lambda(x_i) = \{v_i^p\}$ where p is the index of each neighbor. For a pair of point coordinates (x_i, x_j), assume the set of nearest neighbors in V of x_i and x_j are $\{v_i^p\}$ and $\{v_j^q\}$, respectively. Here, we use $\gamma_{pq}(x_i, x_j)$ to denote the unnormalized confidence score that path between x_i and x_j goes through v_i^p and v_j^q. σ here is a generic Gaussian radial basis function. We define α_{ij}, the confidence of an undirected path between $v_i \in V$ and $v_j \in V$, as:

$$\alpha_{pq}(x_i, x_j) = \frac{\gamma_{pq}(x_i, x_j)}{\sum_{p \in |\Lambda(x_i)|, q \in |\Lambda(x_j)|} \gamma_{pq}(x_i, x_j)} \quad (3)$$

$$\gamma_{pq}(x_i, x_j) = \sigma(x_i, v_i^p)\sigma(x_j, v_j^q) \quad (4)$$

$$= \exp(-(||x_i - v_i^p||_2^2 + ||x_j - v_j^q||_2^2)) \quad (5)$$

The confidence of a path between x_i and x_j going through the two vertices v_i^p and v_j^q is the normalized similarity between (x_i, x_j) from our prediction and their possible corresponding vertices (v_i^p, v_j^q) in the graph measured by a radial basis kernel. Because of the normalization step, the confidence over all possible paths can be seen a probability distribution over which vertices (v_i^p, v_j^q) to choose on the ground truth object. Thus, we can define the **soft** geodesic loss function as:

$$g(x_i, x_j) = \sum_{\substack{v_i^p \in \Lambda(x_i), \\ v_j^q \in \Lambda(x_j)}} \alpha_{pq}(x_i, x_j) D(v_i^p, v_j^q) \quad (6)$$

Experiments

In this section, we demonstrate the utility of Geo-HOF in several 3D reconstruction settings. First, we show that Geodesic-HOF is able to reconstruct 3D objects accurately while learning the geodesic distance. On the ShapeNet (Chang et al. 2015) dataset, Geodesic-HOF performs competitively in terms of Chamfer distance (Table 1) and in normal consistency (Table 2) compared against the current state of the art 3D reconstruction methods. Then we show a set of interesting applications of our learned embeddings in tasks such as mesh reconstruction and manifold decomposition (Figures 4 and 5). Our experiments show that we can learn important properties such as curvature from the learned embeddings to render a better representation of the object shape.

Single View Object Reconstruction

In this section, we evaluate the quality of our learned representation by performing a single-view reconstruction task on the ShapeNet (Chang et al. 2015) dataset. For fair comparison, we use the data split provided in (Choy et al. 2016b). We use the pre-processed ShapeNet renderings and ground truth objects from (Mescheder et al. 2019). For evaluation, we use two main metrics: Chamfer distance and Normal consistency. The detail of the dataset can be found in Supplementary Material.

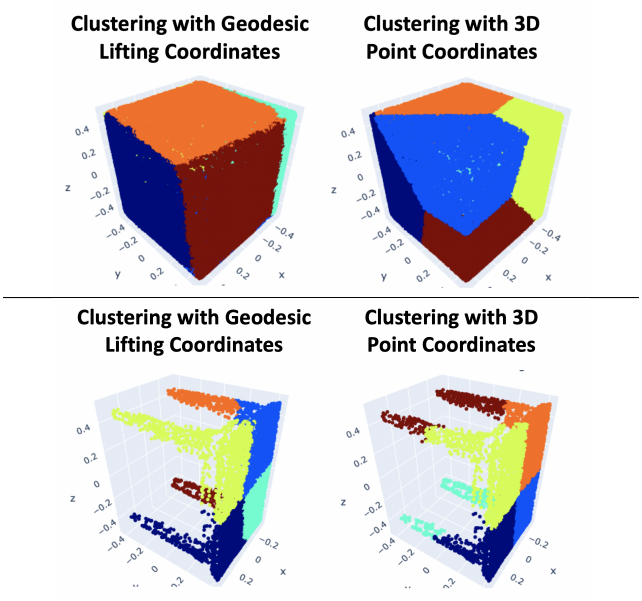

Figure 5: Manifold Decomposition of a canonical cube shape (Top) and a table from ShapeNet (Bottom). Comparing surface manifold decompositions by clustering raw 3D point coordinates x_i and geodesic lifting coordinates w_i of the predicted embedding $z_i = [x_i, w_i]$. K-Means is used for clustering with K=6. Geodesic lifting coordinates give more meaningful features that partition the surface into intuitive components. Best viewed in color.

Evaluation Metrics Chamfer distance is defined identically as Equation 1. We sample 100,000 points from both our output representation and the ground truth point cloud. Note that the original objects are normalized so that the bounding box of each object is a unit cube and we follow the evaluation procedure of (Mescheder et al. 2019) to use $1/10$ of the bounding box size as the unit. Normal consistency between two normalized unit vectors n_i and n_j is defined as the dot product between the two vectors. For evaluating the surface normals, we first sample oriented points from the object surface, denoted as $X_{pred} = \{(\vec{x_i}, \vec{n_i})\}$ and the set of ground truth points and the corresponding normals $X_{gt} = \{(\vec{y_j}, \vec{m_j})\}$. The surface normal consistency between two set of oriented points sample from the object, denoted as Γ, is defined as:

$$\Gamma(X_{gt}, X_{pred}) = \frac{1}{|X_{gt}|} \sum_{i \in |X_{gt}|} |\vec{n_i} \cdot \vec{m}_{\theta(x, X_{pred})}| \quad (7)$$

$$\theta(x, X_{gt} := \{(\vec{y_j}, \vec{m_j})\})) = \underset{j \in |X_{gt}|}{\arg\min} ||x - y_j||_2^2 \quad (8)$$

In Table 1 we show our Chamfer distance comparison with 3D-R2N2 (Choy et al. 2016a), PSGN (Point Set Generating Networks) (Fan et al. 2017), Atlasnet (Groueix et al. 2018) and Pixel2Mesh (Wang et al. 2018). In Table 2, we show the normal comparison results with the same set of methods. In Table 1, it can be seen that Geodesic-HOF can accurately represent the object as indicated by the best overall chamfer performance. By learning a continuous mapping function between the mapping domain and the embedding domain, we allow high-resolution sampling of the embedding space to represent the object. In addition, by using a Higher-Order Function Network as the decoder, we avoid the dimensionality constraints of the latent vector code to provide a more direct way for image information to flow into the mapping function. In comparison, the methods that have pre-defined connectivity, such as Pix2Mesh (Wang et al. 2018) and AtlasNet (Groueix et al. 2018), suffer from the topological constraints of the sampling domain. Furthermore, we present the qualitative results of our reconstruction results in Figure 3. While achieving visually more accurate mesh reconstruction, Geodesic-HOF provides additional information such as chart decomposition, which we explain in detail next.

Table 2 shows that our normal estimation results compared with other methods. The last two columns show the normal consistency obtained with two methods. Both methods use a nearest neighbor search to find the local planar neighborhood and estimate the normals based on a principal axes analysis of the neighborhood. Ours (Euc) uses Euclidean distance to find the neighborhood whereas Ours (Geo) uses the learned geodesic embedding to find the neighborhood. The comparison between the last two columns shows that the geodesic embeddings provide a better guide for finding the low curvature neighborhood. For example, on the edge of the table, if we estimate the normal of the points on the tabletop near the edge, we should avoid the inclusion of the points on the side of the table. From our experiments, the geodesic neighborhoods yield overall better normal estimation results. Note that implicit methods such as Occupancy Network have a natural advantage in normal estimation due to the filtering effect of the marching cubes post-processing, which is advantageous for datasets with few high-frequency surface features. Despite this difference, our normal consistency performs competitively with state of the art methods. This result highlights the effectiveness of the learned geodesics in understanding the local structure, such as curvature, of the object.

Applications

In this section, we show two example applications of our learned geodesic embeddings. We use the geodesic lifting dimensions of the embeddings to decompose shapes into simpler, non-overlapping charts. We also show how each chart from decomposition can be represented as an explicit function $y = f(u, v)$ which can then be used for mesh reconstruction.

Chart Decomposition The object surface can be represented as a differentiable 2-manifold, also known as an atlas, using a collection of charts. Clustering Geodesic-HOF embeddings according to the lifting coordinates provides a natural way to split the object into a small number of low curvature charts as illustrated in Figure 5, where we contrast this technique with a standard clustering method based on Euclidean distance in $\mathbb{R}^3$. Our method correctly separates the faces of the cube (left) and the legs and the table-top (right). The charts can be useful in many applications such as establishing a uv-map for texture mapping and for triangulation

	Pix2Mesh	AtlasNet	OccNet	Ours (Mesh)
Chamfer	0.216	0.175	0.215	**0.169**
Normal	0.772	0.811	**0.834**	0.780

Table 3: Mesh Reconstruction comparison on entire ShapeNet test set. Evaluation is done by sampling 100,000 points uniformly on the predicted mesh. The surface normal of each point comes from the surface normal of the triangles the points belong to. Pix2Mesh, AtlasNet and Geodesic-HOF are explicit methods whereas OccNet is based on learning occupancy as an implicit function. Mean values are reported.

	Chamfer		Normal	
Category	HOF	Geo-HOF	HOF	Geo-HOF
Airplane	**0.096**	0.099	0.845	**0.863**
Bench	0.123	**0.122**	0.779	**0.795**
Cabinet	**0.134**	**0.134**	0.868	**0.870**
Car	**0.996**	0.100	0.818	**0.827**
Chair	0.174	**0.173**	0.795	**0.810**
Display	0.194	**0.193**	0.836	**0.862**
Lamp	0.235	**0.229**	0.707	**0.732**
Speaker	0.209	**0.206**	0.829	**0.838**
Rifle	0.097	**0.096**	0.793	**0.797**
Sofa	**0.156**	0.162	0.853	**0.855**
Table	**0.138**	0.145	0.826	**0.841**
Telephone	0.110	**0.109**	0.923	**0.933**
Vessel	**0.129**	0.137	0.776	**0.790**
mean	**0.137**	0.141	0.816	**0.828**

Table 4: Chamfer and normal evaluation of regular HOF vs. Geodesic-HOF.

based mesh generation.

Mesh Reconstruction Once we decompose the object into charts, we can fit a surface to each chart and establish a 2D coordinate frame. The 2D coordinates can then be used for triangulation. Here, we present a general approach using a multi-layer Perceptron f_θ to represent the manifold. This is related to the approach of AtlasNet; however, with Geo-HOF, we have low-curvature regions already partitioned into charts and have point coordinates associated with each chart, allowing learning of the parameters θ in an unsupervised manner. We present additional details in the Supplementary Material and an illustration of this method in Figure 4.

In Figure 4, we decompose an airplane to 20 non-overlapping charts by clustering the lifting coordinates, triangulating each chart using the method described above. We observe that the charts are split nicely at high-curvature areas. Finally, we compare the resulting meshes with a state of the art implicit function method: Occupancy Networks learns an occupancy function to represent the object and uses marching cubes to generate a mesh. In Table 3, we present mean values over all ShapeNet classes, and present full comparison across classes in the Supplementary Material section. Note that, unlike the other methods in this table, we fit the manifolds as an optimization step without accessing the ground truth. Nevertheless, we show competitive Chamfer distance and normal consistency.

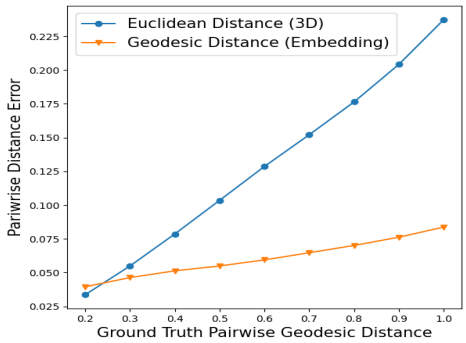

Figure 6: Pairwise distance error from network output vs ground truth pairwise geodesic distance.

Ablation Study

To further understand the benefits of geodesic supervision, we compared regular HOF (without geodesic loss) with Geodesic-HOF in Table 4. While the geodesic loss poses additional constraints on the network, we show that Geodesic-HOF is able to learn the geodesic information without sacrificing the chamfer distance performance. Further, we plot the distance error between every pair of points from the output of the two networks in Figure 6. As the ground truth distance increases, the Euclidean approximation of geodesic distance becomes worse. With the geodesic supervision, Geodesic-HOF can predict the pairwise distance much more robustly. The additional geodesic information allows the network to query neighbors more accurately. This explains why Geodesic-HOF significantly outperforms baselines in normal estimation.

Conclusion

We presented Geodesic-HOF for generating continuous surface models. Geodesic-HOF can generate surface samples at arbitrary resolution and estimating the geodesic distance between any two samples. Comparisons showed that our method can generate more accurate point samples and surface normals than state of the art methods which can directly generate surfaces. It also performs comparably to implicit function methods which rely on post-processing the output to generate the object surface. We also presented two applications of Geodesic-HOF: partitioning the surface into non-overlapping, low-curvature charts and learning a functional mapping of each chart. Combining these two applications allows us to generate triangulations of the object directly from the network output. This approach alleviates some of the limitations of the previous mesh generation methods such as overlapping charts or genus constraints. We then performed ablation study to understand the effect of geodesic supervision on reconstruction and neighbor estimation.

We hope that this work will motivate future research in 3D reconstruction that learns higher-order surface properties like geodesics in order to produce more useful, accurate representations of 3D objects.

References

Balasubramanian, M.; and Schwartz, E. L. 2002. The isomap algorithm and topological stability. *Science* 295(5552): 7–7.

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

Choy, C. B.; Xu, D.; Gwak, J.; Chen, K.; and Savarese, S. 2016a. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European Conference on Computer Vision*, 628–644. Springer.

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

Cox, M. A.; and Cox, T. F. 2008. Multidimensional scaling. In *Handbook of data visualization*, 315–347. Springer.

Fan et al. 2017. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 605–613.

Gkioxari, G.; Malik, J.; and Johnson, J. 2019. Mesh R-CNN. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

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

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In Bengio, Y.; and LeCun, Y., eds., *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. URL http://arxiv.org/abs/1412.6980.

Lin, C.-H.; Kong, C.; and Lucey, S. 2018. Learning Efficient Point Cloud Generation for Dense 3D Object Reconstruction. In *AAAI Conference on Artificial Intelligence (AAAI)*.

Luciano, L.; and Hamza, A. B. 2018. Deep learning with geodesic moments for 3D shape classification. *Pattern Recognition Letters* 105: 182–190. doi:10.1016/j.patrec.2017.05.011. URL https://doi.org/10.1016/j.patrec.2017.05.011.

Maaten, L. v. d.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9(Nov): 2579–2605.

Mescheder, L.; Oechsle, M.; Niemeyer, M.; Nowozin, S.; and Geiger, A. 2019. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4460–4470.

Mitchell, E.; Engin, S.; Isler, V.; and Lee, D. D. 2019. Higher-Order Function Networks for Learning Composable 3D Object Representations. *arXiv preprint arXiv:1907.10388* .

Pai, G.; Talmon, R.; Bronstein, A.; and Kimmel, R. 2019. DIMAL: Deep Isometric Manifold Learning Using Sparse Geodesic Sampling. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 819–828.

Park, J. J.; Florence, P.; Straub, J.; Newcombe, R.; and Lovegrove, S. 2019. Deepsdf: Learning continuous signed distance functions for shape representation. *arXiv preprint arXiv:1901.05103* .

Riegler, G.; Osman Ulusoy, A.; and Geiger, A. 2017. OctNet: Learning Deep 3D Representations at High Resolutions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Smith, E. J.; Fujimoto, S.; Romero, A.; and Meger, D. 2019. GEOMetrics: Exploiting Geometric Structure for Graph-Encoded Objects. *arXiv preprint arXiv:1901.11461* .

Tatarchenko, M.; Dosovitskiy, A.; and Brox, T. 2017. Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In *Proceedings of the IEEE International Conference on Computer Vision*, 2088–2096.

Tenenbaum, J. B.; De Silva, V.; and Langford, J. C. 2000. A global geometric framework for nonlinear dimensionality reduction. *science* 290(5500): 2319–2323.

Wang, N.; Zhang, Y.; Li, Z.; Fu, Y.; Liu, W.; and Jiang, Y.-G. 2018. Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 52–67.

Wang, Y.; Peterson, B. S.; and Staib, L. H. 2000. Shape-based 3D surface correspondence using geodesics and local geometry. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No.PR00662)*, volume 2, 644–651 vol.2.

Wang, Y.; Peterson, B. S.; and Staib, L. H. 2003. 3D Brain surface matching based on geodesics and local geometry. *Computer Vision and Image Understanding* 89(2-3): 252–271. doi:10.1016/s1077-3142(03)00015-8. URL https://doi.org/10.1016/s1077-3142(03)00015-8.

Wu, J.; Zhang, C.; Xue, T.; Freeman, B.; and Tenenbaum, J. 2016. Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling. In Lee, D. D.; Sugiyama, M.; Luxburg, U. V.; Guyon, I.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 29*, 82–90. Curran Associates, Inc. URL http://papers.nips.cc/paper/6096-learning-a-probabilistic-latent-space-of-object-shapes-via-3d-generative-adversarial-modeling.pdf.

Yang, Y.; Feng, C.; Shen, Y.; and Tian, D. 2018. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 206–215.