GAMesh: Guided and Augmented Meshing for Deep Point Networks

Nitin Agarwal M Gopi
University of California, Irvine

Supplementary Document

A. Effects of Mesh Priors on GAMesh

As GAMesh uses a mesh prior to generate a surface for the output points from a point network, a natural question to ask is what are the prerequisite of a mesh prior. Here we study the effects of the mesh prior on the final reconstructed surface.

We voxelize the ground truth shape at three different resolutions and use the same set of points to reconstruct surfaces using GAMesh. Specifically, we create three mesh priors whose number of octree leaf nodes are 100, 500 and 4000 respectively and use 2500 points which are sampled on the ground truth mesh. As evident from the Fig. 2, using the same points, GAMesh reconstructs geometrically similar shapes whose topology is dictated by the topology of the mesh prior. This confirms that GAMesh does not need accurate mesh priors for surface generation. As long as the mesh priors have correct topology and coarsely resemble the original mesh, accurate surfaces will be generated. In our experience, such mesh priors, if not already present, can be easily obtained using existing reconstruction methods (volumetric/implicit networks). We, however, observe that surfaces generated using coarser meshes have different triangulation. This is due to the edge collapse operation during simplification where the cost to collapse an edge depends on the length of the edge. If we allow edge-flip as an operation during simplification, it is possible that we can generate consistent triangulation's independent of the template mesh resolution.

Non-Manifold Mesh Priors. We also show few surfaces generated from GAMesh using non-manifold mesh priors which contain multiple connected components (Fig. 3). As GAMesh borrows the topology of the mesh prior, we observe that the output mesh is disconnected as the mesh prior itself is not a single watertight mesh.

Bounds on the Mesh Priors. Having seen a few qualitative examples, we now provide theoretical bounds on the mesh prior for GAMesh to generate accurate surfaces with correct topology. We first define what is a feature size and then provide a formal proof on the bounds of the mesh prior.

For reconstructing a surface, a feature size f for a point on the surface is defined as the Euclidean distance from that point to the nearest point on the medial axis [1], which can be represented by the set of Voronoi vertices as shown in Fig. 1.

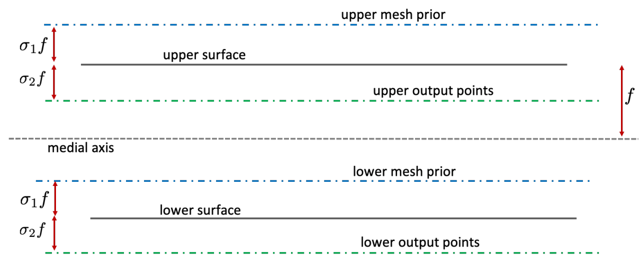

Figure 1: **Bounds on Mesh Prior.** For GAMesh to reconstruct a surface with correct topology, the sum of the error in both the mesh prior (blue) and the output points from the point network (green) need to be less than a threshold.

Lemma 1. *In order for GAMesh to reconstruct a surface with correct topology, the sum of the error in the mesh prior and the error in the output points needs to be less than 1.*

Proof. Let the error in the mesh prior be $\sigma_1 f$ and the error in the output points be $\sigma_2 f$. In order for GAMesh to reconstruct a surface with correct topology, the upper output points need to be projected onto the upper mesh prior. Thus from Fig. 1 we get,

$$\begin{aligned} \sigma_1 f + \sigma_2 f &< (f - \sigma_2 f + f - \sigma_1 f) \\ (\sigma_1 + \sigma_2) f &< (2f - (\sigma_1 + \sigma_2) f) \\ 2(\sigma_1 + \sigma_2) f &< 2f \\ \sigma_1 + \sigma_2 &< 1 \end{aligned} \tag{1}$$

Hence, the upper bound on the mesh prior is when $\sigma_1 = 0$ i.e. the mesh prior is the same as the ground truth surface and the lower bound is $\sigma_1 < 1 - \sigma_2$. However, for all practical purposes, $\sigma_1 + \sigma_2 < 0.5$ for reconstructing a surface with correct topology using GAMesh. From Fig. 2 we can see that the mesh prior with resolution 100 (second column) does not satisfy Eq. 1 resulting in an output surface with incorrect topology.

□

Figure 2: **Effects of Mesh Prior on GAMesh.** Surfaces generated using the same points (orange) but with three mesh priors of different resolution (shown in inset). GAMesh creates geometrically similar but topologically different meshes. (Please zoom in for details)

Figure 3: **Non-Manifold Mesh Priors.** GAMesh reconstruct accurate surfaces even with non-manifold mesh priors which have multiple connected components. The topology of mesh prior is carried over to the output mesh (see pink arrow).

B. Effects of Outputs Points on GAMesh

Next, we demonstrate that GAMesh always reconstructs meshes with correct topology while preserving as much details as maintained by the output points of the network. We show this by injecting noise in the vertices of a mesh and evaluating the surface reconstructed from GAMesh, which uses the original mesh as prior. Specifically, keeping the mesh prior fixed, we reconstruct surfaces with GAMesh by adding Gaussian noise in the normal direction with standard deviation 5%, 20% and 30% of the length of the bounding box diagonal to the vertices of the mesh.

From Fig. 4 we observe that adding noise in the normal direction does not affect the output (i.e. triangulation) of GAMesh, as the output points are projected to the same point on the mesh prior. This suggests that the error in the final reconstruction (low F1 score) is due to the point location

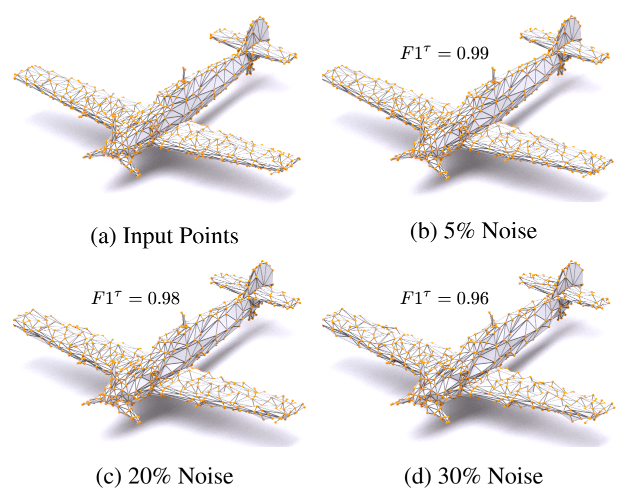

Figure 4: **Effects of Points on GAMesh.** Surfaces generated from GAMesh when Gaussian noise is added to the mesh vertices. Using the same mesh prior (ground truth mesh), we preserve as much detail as maintained by the points (Please zoom in for details).

and *not* from GAMesh. We also observe that GAMesh reconstructs a smooth surface provided the points lie close to the surface of the mesh prior. However, as GAMesh guarantees to connect all output points, rough surfaces can be generated even if there is noise in only few points. This helps GAMesh differentiate subtle differences between two point clouds making it an ideal meshing algorithm for evaluating point reconstruction networks (as shown in Section 5.1 of the main paper).

C. GAMesh preserves Topology and Geometry

GAMesh preserves the topology of the mesh prior as all operations such as local triangulation and edge collapse are topology preserving operations. Under extreme simplification, certain edges, if collapsed, may geometrically (visually) close genus of the model, although in implementation with multi-edge data-structure between vertices, the genus and rest of the topology can be maintained. With sufficient number of output points from the network such extreme simplifications and hence visual change of topology can be prevented. GAMesh also preserves the geometry as much as possible based on the output quality of points networks. Since these reconstructed points are projected to the closest point in the mesh, GAMesh does its best to preserve the geometry by imposing the connectivity of the input mesh to the closest reconstructed points.

D. Single View Reconstruction

In Figure 7 and 8 we provide more qualitative comparisons of surfaces reconstructed from GAMesh when used in post-processing to combine the output points of our point network (PSG$^+$) and meshes from IM-NET. Unlike GEOMetric which deforms a fixed template, we borrow the topology from IM-NET to reconstruct meshes with correct topology. Furthermore, using the output of point networks which are specifically trained for geometry, we reconstruct meshes with higher fidelity than MeshRCNN using similar number of points (see the wings/engine of the planes and the details around the wheels of the cars in Fig. 7 & 8). Lastly, using BPA to generate a surface for the output points of the point network gives inconsistent results as it fails to connect all the output points.

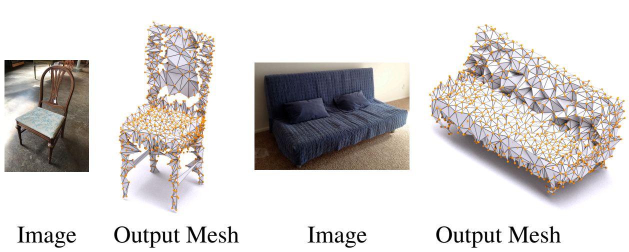

Figure 5: **SVR on Natural Images.** Single-view reconstruction results on two images from Pix3D [4].

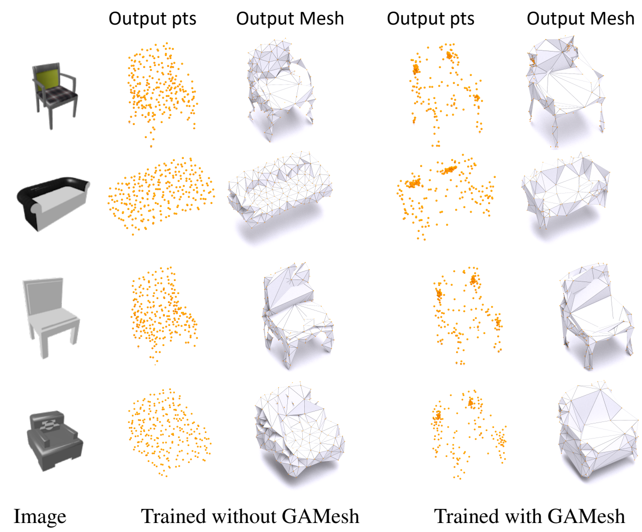

Figure 6: **Training with GAMesh.** More qualitative comparison of surfaces reconstructed using point networks trained with GAMesh.

We also test our method on natural images such as those from Pix3D [4] in Fig. 5.

E. Training Point Networks with GAMesh

Data & Implementation Details To demonstrate the advantages of training point networks with GAMesh, we train two networks one with Chamfer loss and the other with mesh loss ($\mathcal{L}_{mesh}$). Both networks have the same architecture where for the image encoder we use ResNet-18 [3] and for the point decoder 4 fully-connected layers of size 1024, 512, 256, (3x250). We use ReLU non-linearity and batch normalization on the first three and tanh on the final layer. We use data from two categories (chair and couch) of ShapeNet, specifically one image per shape making a total of 7956 train and 1991 test images. We train both networks with a batch size of 4 images and Adam optimizer with learning rate 10^{-4} for 100 epochs.

Qualitative Results In Fig. 6 we provide more qualitative results of surfaces reconstructed when GAMesh is used to train point networks. Training point networks with a mesh loss on surfaces generated by GAMesh allows us to focus on high curvature regions like edges of chair and couch. Although here we only used the reconstruction loss ($\mathcal{L}_{mesh}$), similar to MeshRCNN [2] and P2M [5], additional shape regularizers can also be introduced to impose further smoothness.

Figure 7: **Single View Reconstruction.** Qualitative comparison of ShapeNet testset meshes from various SVR approaches. Using meshes from IM-NET as priors, GAMesh reconstructs accurate surfaces for the output points (orange) of PSG^{+}.

Figure 8: **Single View Reconstruction.** More Qualitative results from ShapeNet testset for Single-View Reconstruction.

References

[1] N. Amenta and M. Bern. Surface reconstruction by voronoi filtering. *Discrete & Computational Geometry*, 1999. 1

[2] G. Gkioxari, J. Malik, and J. Johnson. Mesh r-cnn. In *ICCV*, 2019. 3

[3] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3

[4] X. Sun, J. Wu, X. Zhang, Z. Zhang, C. Zhang, T. Xue, J. B. Tenenbaum, and W. T. Freeman. Pix3d: Dataset and methods for single-image 3d shape modeling. In *CVPR*, 2018. 3

[5] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, and Y.-G. Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *ECCV*, 2018. 3