GAMesh: Guided and Augmented Meshing for Deep Point Networks

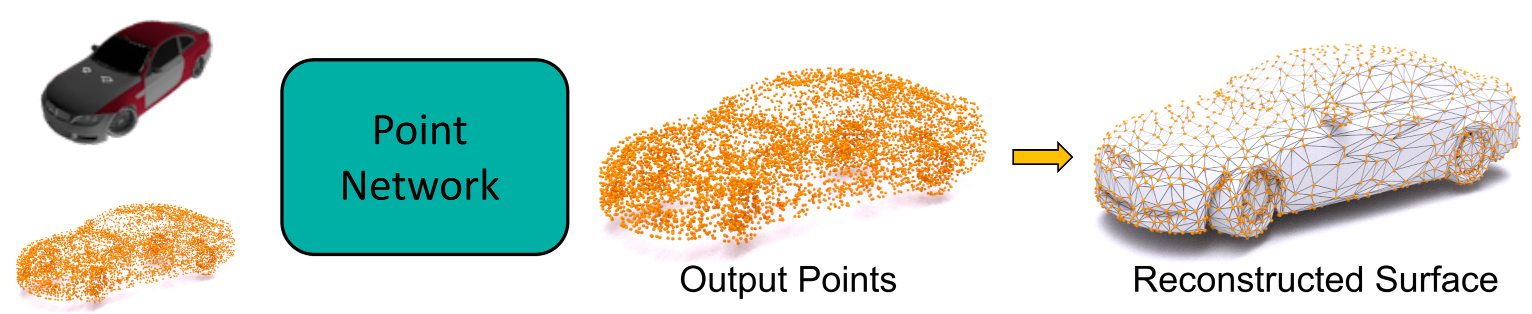
Nitin Agarwal and M. Gopi

Department of Computer Science, University of California, Irvine

Project Webpage - https://www.ics.uci.edu/~agarwal/GAMesh

3DV 2020

Motivation



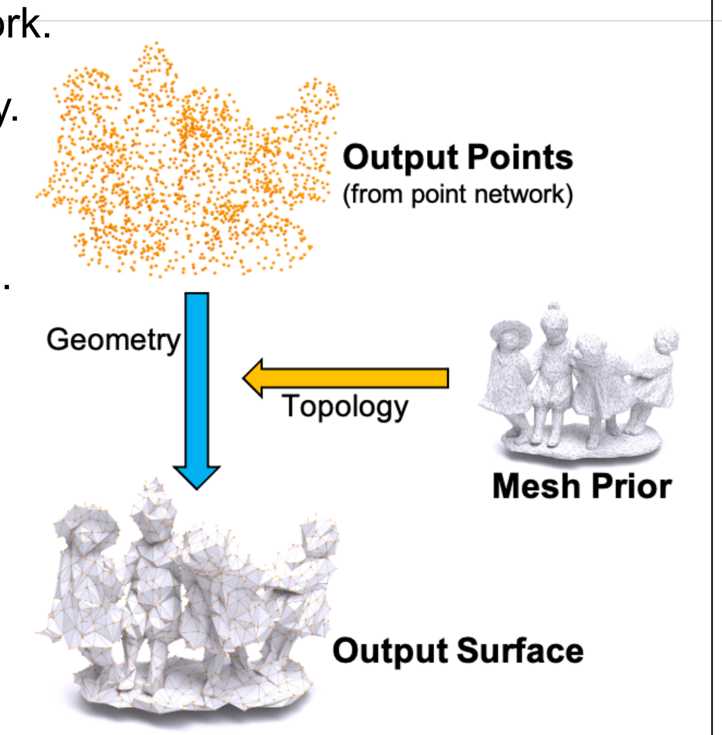
- Evaluating Point Networks.
- Visualization of Results.
- Training Point Network.

Generate a surface with correct topology by connecting all the output points from a point reconstruction network.

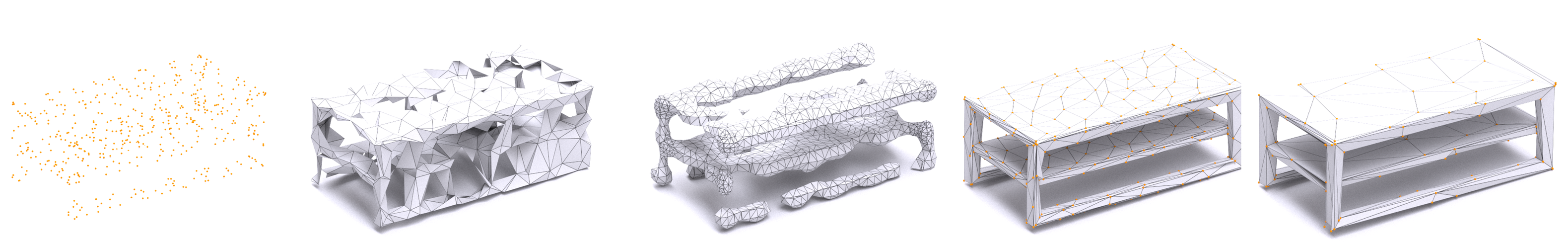
Contributions

We propose a new meshing algorithm, **GAMesh** which:

- Meshes the output pts from a point network.
- Requires a mesh prior w/ correct topology.
- **Decouples** geometry from topology.
- **Invariant** to point density and distribution.
- Requires no parameter tuning.
- **Differentiable**.



The code and data are available on project page

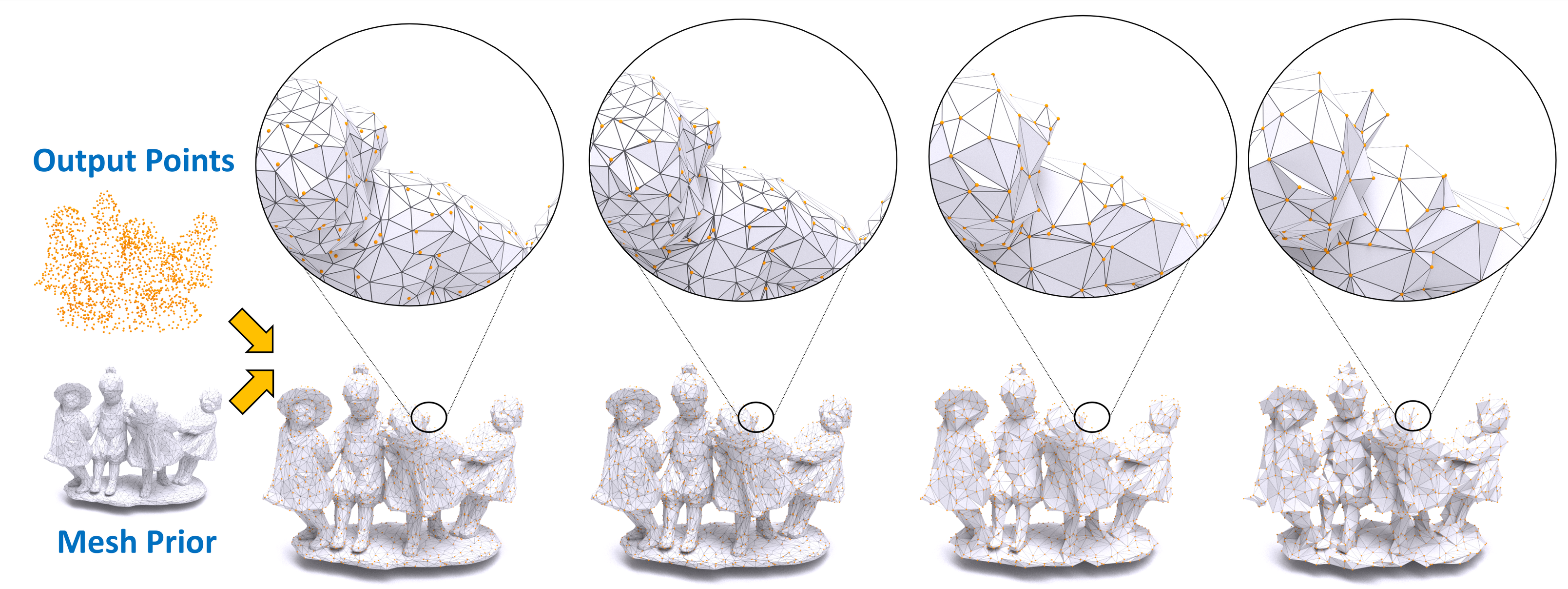


Point Cloud | Ball Pivot Algorithm | Screened Poisson Reconstruction | Ours | Ours (30% pts)

References

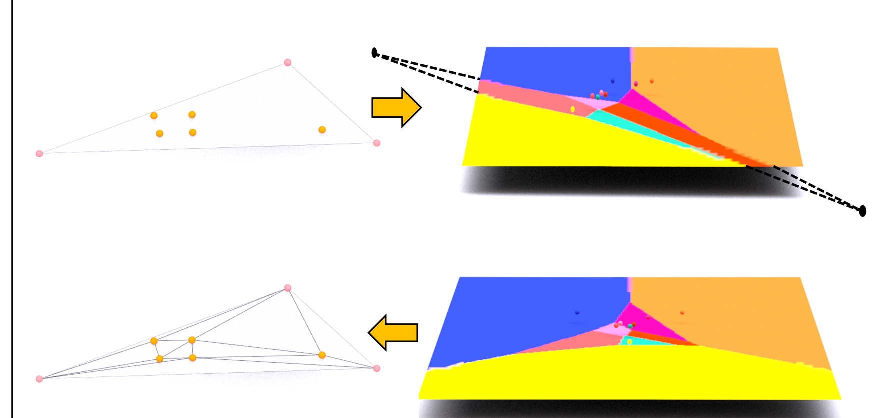
Bernardini et al., The ball-pivoting algorithm for surface reconstruction. TVCG 1999.
Kazhdan et al., Screened Poisson surface reconstruction. TOG 2013.
Choy et al., 3d- r2n2: A unified approach for single and multi-view 3d object reconstruction. ECCV 2016.
Fan et al., A point set generation network for 3d object reconstruction from a single image. CVPR 2017.
Kato et al., Neural 3D mesh renderer. CVPR 2018.
Wang et al., Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images, ECCV 2018.
Smith et al., Exploiting geometric structure for graph-encoded objects. ICML 2019.
Chen et al., Learning implicit fields for generative shape modeling. CVPR, 2019.
Mescheder et al., Occupancy networks: Learning 3d reconstruction in function space. CVPR 2019.
Agarwal et al., Learning embedding of 3d models with quadric loss. BMVC 2019.
Gkioxari et al., Mesh RCNN. ICCV 2019.

Guided and Augmented Meshing



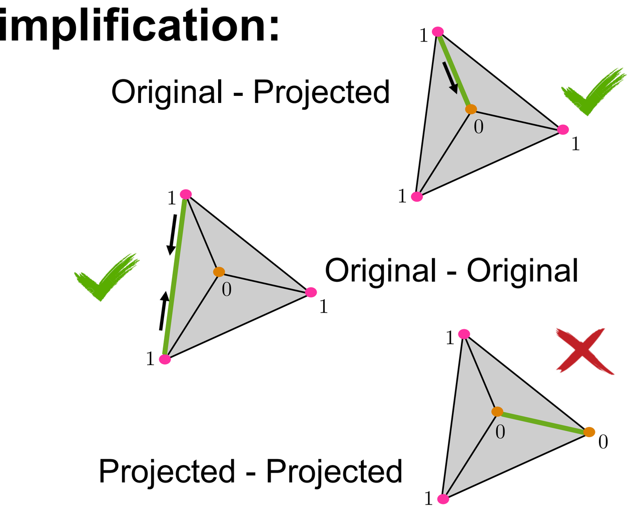
Projection | Triangulation | Simplification | Output Surface

Delaunay Triangulation:



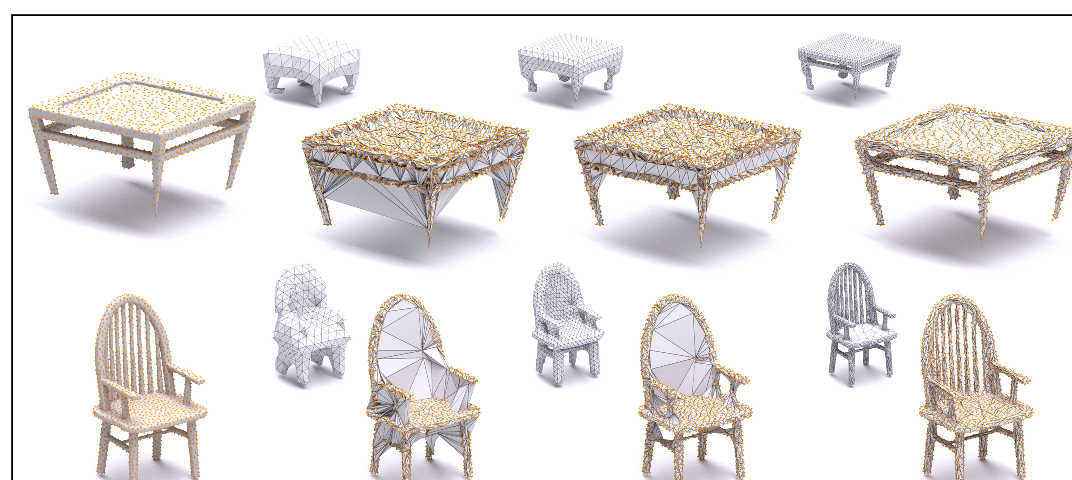
Order-independent + Parallel processing

Simplification:



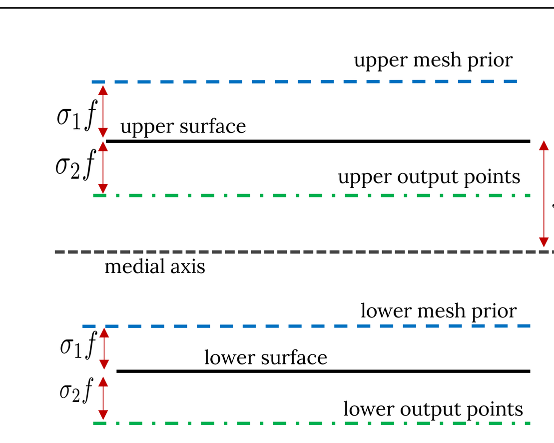
Valid edge collapse operations

Effects of Mesh Prior on GAMesh



2500 Input Points | Voxel Res 100 | Voxel Res 500 | Voxel Res 4000

Mesh prior voxelized at three different resolutions



$\sigma_1 + \sigma_2 < 1.0$

Bounds on Mesh Prior

Fair Evaluation of Point Networks

Trained four point networks with same backbone but different # of output points.

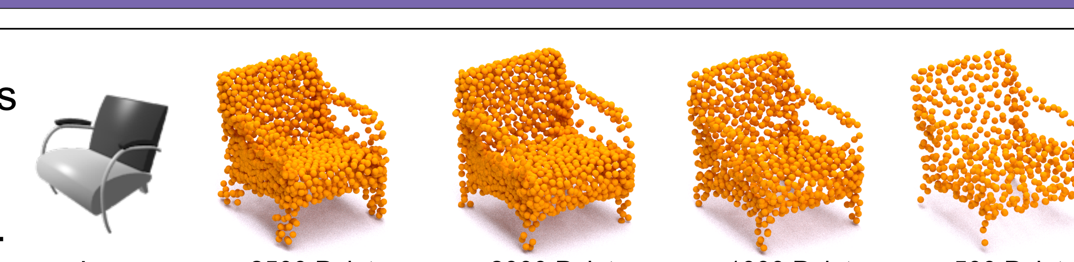


Image | 2500 Points | 2000 Points | 1000 Points | 500 Points

		$\lvert V\rvert$				$\Delta(\downarrow)$
		2500	2000	1000	500	
Chair	CD (↓)	4.60	5.01	6.76	9.13	98.47%
	$F1^\tau$ (BPA) (↑)	$68.77_{5.9\%}$	$69.08_{5.8\%}$	$65.63_{5.4\%}$	$57.39_{5.4\%}$	19.82%
	$F1^\tau$ (SPR) (↑)	$56.64_{99.9\%}$	$55.30_{99.9\%}$	$50.39_{100\%}$	$44.64_{99.9\%}$	26.88%
	$F1^\tau$ (GAMesh) (↑)	$\mathbf{72.74}_{0\%}$	$\mathbf{73.32}_{0\%}$	$\mathbf{70.95}_{0\%}$	$\mathbf{69.25}_{0\%}$	**4.79%**

Evaluate both Output Points and Output Surface using GAMesh.

Single-View Reconstruction

Post-Processing with GAMesh:

We combine the outputs of point generation networks and implicit networks to generate meshes with high fidelity and correct topology.

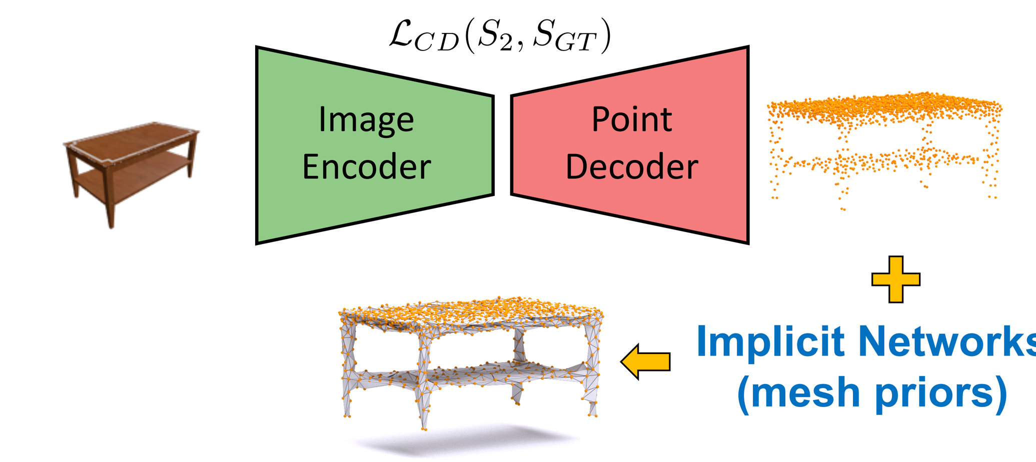


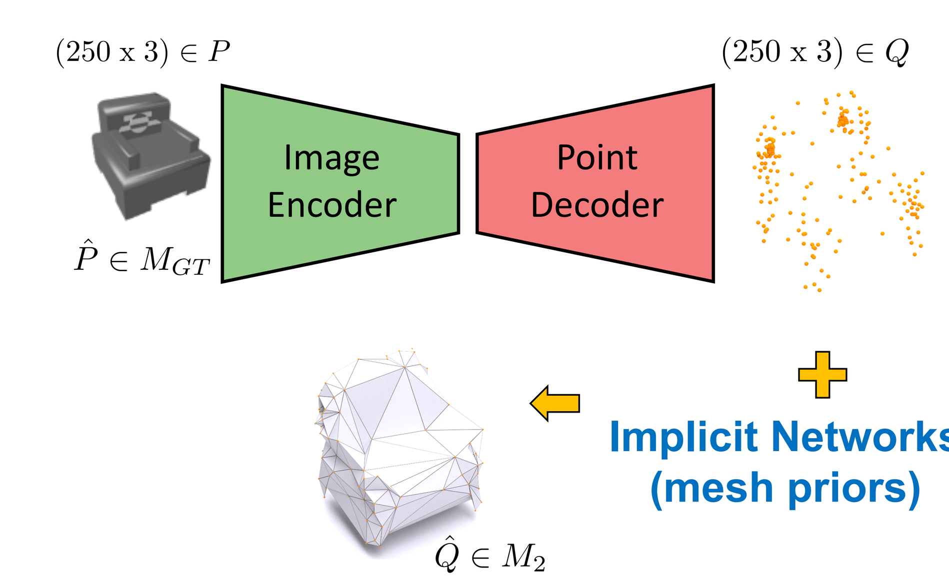
Image | IM-NET | IM-NET + GT Mesh | GAMesh w/ IM-NET

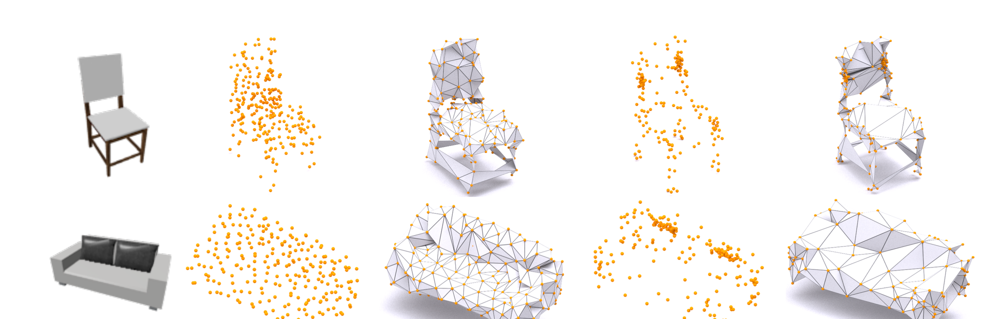
Image | GEOMetric | MeshRCNN | BPA | GAMesh w/ IM-NET

	$\lvert V\rvert$	CD (↓)	$F1^\tau$(↑)	Topology
N3MR	$642_{\pm0}$	-	33.80	fixed
3DR2N2	-	-	39.01	arbitrary
PSG	$1024_{\pm0}$	-	48.58	-
OccNET	$1998_{\pm8}$	0.825	62.91	arbitrary
MVD	-	-	66.39	arbitrary
GEOMetrics	$574_{\pm99}$	-	67.37	fixed
IM-NET	$1999_{\pm21}$	0.502	67.68	arbitrary
P2M	$2466_{\pm0}$	0.444	68.94	fixed
MeshRCNN	$1896_{\pm928}$	0.397	69.30	arbitrary
PSG^+	$2000_{\pm0}$	0.424	56.41	-
SPR	$9069_{\pm929}$	1.206	59.53	arbitrary
BPA	$1871_{\pm27}$	0.399	70.81	arbitrary
Ours w/ OccNET	$2000_{\pm0}$	0.415	70.39	arbitrary
Ours w/ IM-NET	$2000_{\pm0}$	0.387	**70.82**	arbitrary

Training with GAMesh:

Most SVR methods generate meshes with uniform distribution of points. By training point networks with GAMesh, we can directly optimize the vertex positions to generate adaptive meshes.





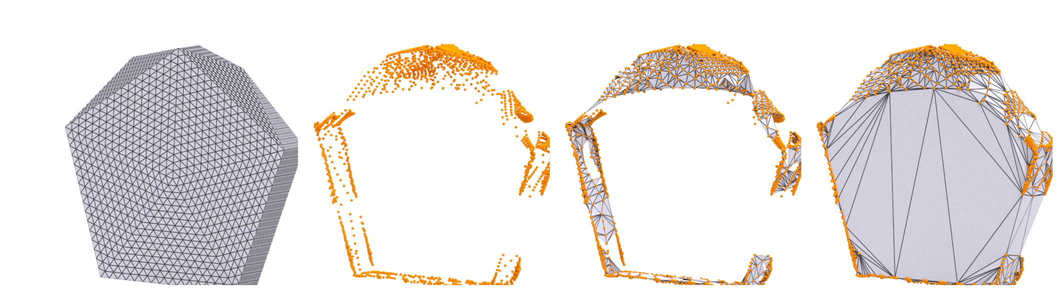
Training w/ Chamfer Loss | Training w/ Mesh Loss

$$\mathcal{L}_{Mesh}(M_2, M_{GT}) = \sum_{p\in\mathcal{P}}\Phi(p,\hat{\mathcal{Q}}) + \sum_{q\in\mathcal{Q}}\Phi(q,\hat{\mathcal{P}})$$

Other Applications

Meshing Sparse Point Clouds:

As GAMesh is indifferent to both point density & distribution, it can be used with various point networks which output sparse point clouds.



Original Mesh | Output w/ Quadric Loss | BPA | GAMesh