

### **PoNQ:** a Neural QEM-based Mesh Representation

Nissim Maruani, Maks Ovsjanikov, Pierre Alliez, Mathieu Desbrun

#### ▶ To cite this version:

Nissim Maruani, Maks Ovsjanikov, Pierre Alliez, Mathieu Desbrun. PoNQ: a Neural QEM-based Mesh Representation. The IEEE/CVF Conference on Computer Vision and Pattern Recognition 2024, Jun 2024, Seattle, United States. 2024. hal-04622974

#### HAL Id: hal-04622974 https://inria.hal.science/hal-04622974

Submitted on 11 Jul2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers. L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License



# **Overview and Contributions**



Points, Normals, Quadrics

- We introduce a novel learnable mesh representation that leverages the Quadric Error Metrics (QEM), previously used for mesh decimation [5].
- Output meshes are sharp and guaranteed to be watertight.



**Densely Sampled** Target Shape

**PoNQ Representation** 





Optimized **Points**, **Normals**, and **Q**uadrics

- **Points**  $\mathbf{p}_{i}$  define spatial regions through their Voronoi diagram.
- Normals  $n_i$  (resp. Quadrics  $Q_i$ ) represent the average normal  $n(s_k)$  (resp. QEM) within each region  $\mathbf{R}_{\mathbf{R}}$ . The QEM metric is defined as:

$$QEM(x, R_i) = \sum_{k \in R_i} d_{\mathbf{s}_k, \mathbf{n}(\mathbf{s}_k)}(x)^2 = [x, 1]^t \mathbf{Q}_i[x, 1], \text{ with } \mathbf{Q}_i \in \mathbb{R}^{4 \times 4}$$
$$\mathbf{v}_i^* = \operatorname{argmin}_x [x, 1]^t \mathbf{Q}_i[x, 1] = A_i^{-1} b_i, \text{ with } \mathbf{Q}_i = \begin{pmatrix} A_i & -b_i \\ -b_i^t & c_i \end{pmatrix}$$

# **PoNQ: a Neural QEM-based Mesh Representation**

# Nissim Maruani<sup>1</sup>, Maks Ovsjanikov<sup>2</sup>, Pierre Alliez<sup>1</sup>, Mathieu Desbrun<sup>12</sup>

<sup>1</sup>Inria, France <sup>2</sup>École polytechnique, France

## **Application: Reconstruction from SDF grids**



PoNQ Mesh

**3D Delaunay Triangulation** and **PoNQ** Mesh



Sampled SDF (Regular Grid)



3D CNN



MC [1

NDC [2]

NMC [3]

Method (on ABC)	Grid	CD↓	<b>F1↑</b>
NDC [2]	<b>32</b> <sup>3</sup>	66.004	0.787
NMC [3]	<b>32</b> <sup>3</sup>	60.755	0.833
VoroMesh [4]	<b>32</b> <sup>3</sup>	2.228	0.835
PoNQ-lite	<b>32</b> <sup>3</sup>	3.539	0.810
PoNQ	<b>32</b> <sup>3</sup>	1.514	0.852
NDC [2]	64 <sup>3</sup>	2.211	0.882
NMC [3]	64 <sup>3</sup>	2.138	0.891
VoroMesh [4]	<b>6</b> 4 <sup>3</sup>	1.219	0.886
PoNQ-lite	<b>6</b> 4 <sup>3</sup>	1.074	0.888
PoNQ	64 <sup>3</sup>	0.886	0.892







PoNQ Mesh

VoroMesh [4]

PoNQ

Ground Truth

NC↑	ECD↓	EF1↑	Watertight
0.941	0.445	0.658	44 %
0.954	0.350	0.693	26 %
0.941	0.802	0.232	100 %
0.953	0.296	0.658	100 %
0.964	0.184	0.713	100 %
0.975	0.223	0.855	23 %
0.980	0.254	0.854	18 %
0.966	0.796	0.207	100 %
0.978	0.128	0.858	100 %
0.980	0.109	0.866	100 %

# **Extension 1: Output Mesh Simplification**





### Extension 2: Open Surfaces (Optimization-based)



1] W. Lorensen and H. Cline. Marching cubes: A high resolution 3D surface construction algorithm. In ACM SIGGRAPH Computer Graphics, 1987.

- on Graphics, 2022.

[4] N. Maruani, R. Klokov, M. Ovsjanikov, P. Alliez, and M. Desbrun. VoroMesh: Learning Watertight Surface Meshes with Voronoi Diagrams. In 2023 IEEE/CVF International Conference on Computer *Vision (ICCV)*, 2023.

[5] M. Garland and P. Heckbert. Surface simplification using quadric error metrics. In Proceedings of the 24th annual conference on Computer graphics and interactive techniques - SIGGRAPH, 1997.



• A PoNQ with half the original resolution can be computed on the GPU before meshing via average pooling:

• QEMs can be utilized in downstream geometry processing tasks, such as the detection of open boundaries in this optimization-based experiment.

### References

[2] Z. Chen, A. Tagliasacchi, T. Funkhouser, and H. Zhang. Neural dual contouring. ACM Transactions

[3] Z. Chen and H. Zhang. Neural marching cubes. ACM Transactions on Graphics, 2021.



Project Page