



HAL
open science

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. CVPR 2024 - The IEEE/CVF Conference on Computer Vision and Pattern Recognition, Jun 2024, Seattle, United States. 2024. hal-04622974

HAL Id: hal-04622974

<https://inria.hal.science/hal-04622974>

Submitted on 11 Jul 2024

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.



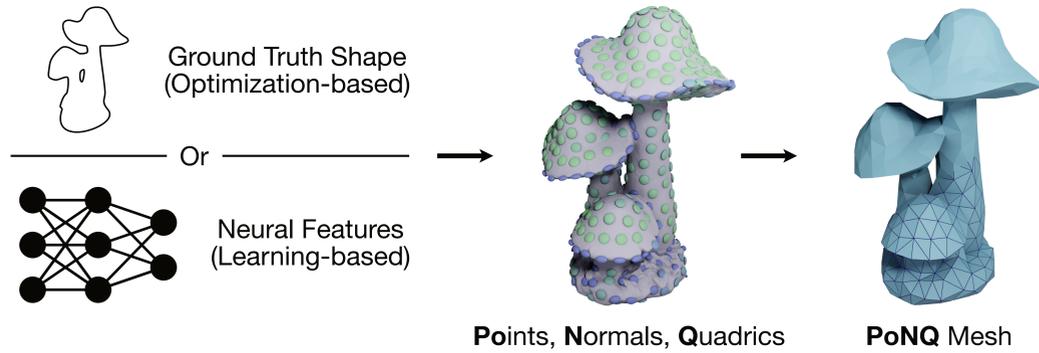
Distributed under a Creative Commons Attribution 4.0 International License

PoNQ: a Neural QEM-based Mesh Representation

Nissim Maruani¹, Maks Ovsjanikov², Pierre Alliez¹, Mathieu Desbrun^{1,2}

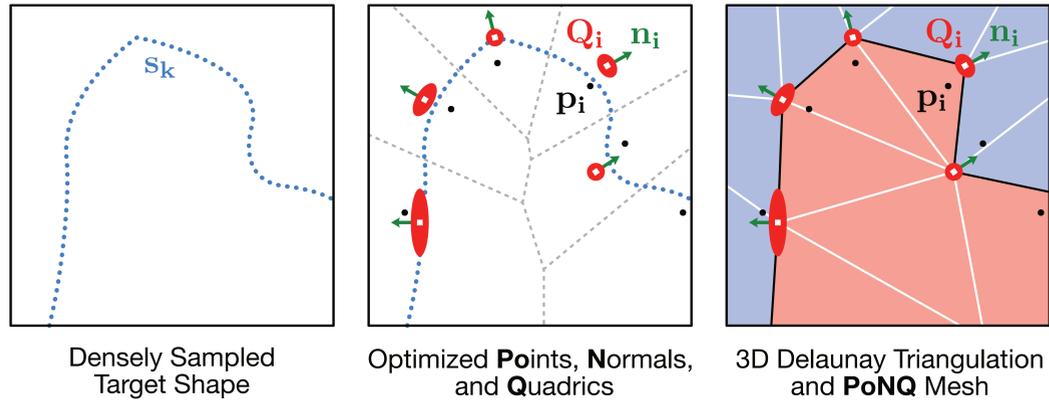
¹Inria, France ²École polytechnique, France

Overview and Contributions



- 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.

PoNQ Representation

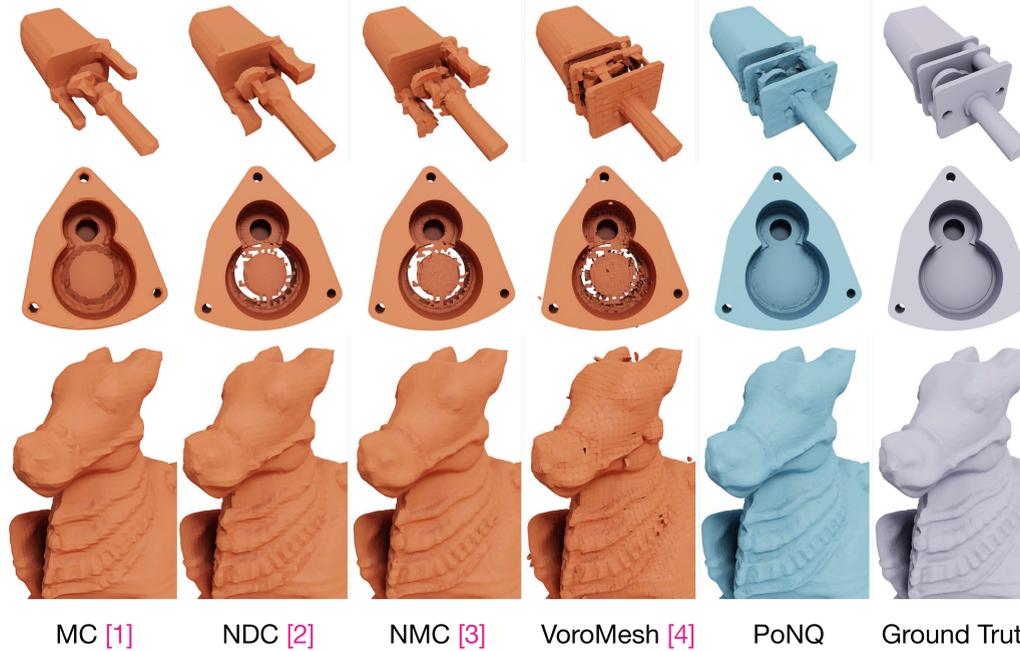
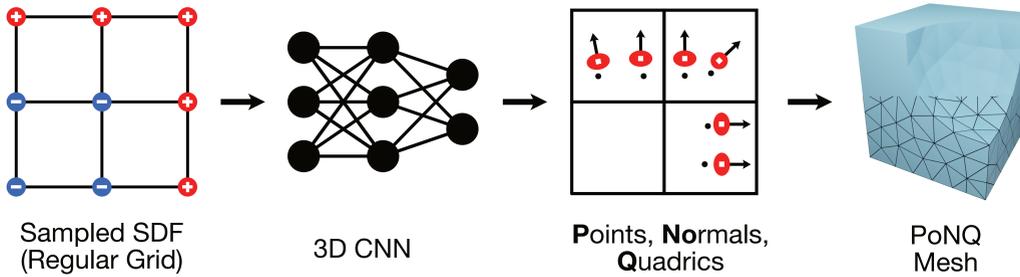


- Points p_i define spatial regions through their Voronoi diagram.
- Normals n_i (resp. Quadrics Q_i) represent the average normal $n(s_k)$ (resp. QEM) within each region R_i . The QEM metric is defined as:

$$QEM(x, R_i) = \sum_{k \in R_i} d_{s_k, n(s_k)}(x)^2 = [x, 1]^t Q_i [x, 1], \text{ with } Q_i \in \mathbb{R}^{4 \times 4}$$

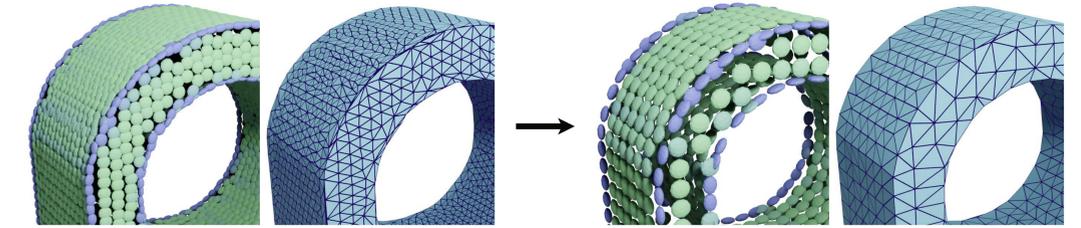
$$v_i^* = \operatorname{argmin}_x [x, 1]^t Q_i [x, 1] = A_i^{-1} b_i, \text{ with } Q_i = \begin{pmatrix} A_i & -b_i \\ -b_i^t & c_i \end{pmatrix}$$

Application: Reconstruction from SDF grids



Method (on ABC)	Grid	CD↓	F1↑	NC↑	ECD↓	EF1↑	Watertight
NDC [2]	32 ³	66.004	0.787	0.941	0.445	0.658	44 %
NMC [3]	32 ³	60.755	0.833	0.954	0.350	0.693	26 %
VoronMesh [4]	32 ³	2.228	0.835	0.941	0.802	0.232	100 %
PoNQ-lite	32 ³	3.539	0.810	0.953	0.296	0.658	100 %
PoNQ	32 ³	1.514	0.852	0.964	0.184	0.713	100 %
NDC [2]	64 ³	2.211	0.882	0.975	0.223	0.855	23 %
NMC [3]	64 ³	2.138	0.891	0.980	0.254	0.854	18 %
VoronMesh [4]	64 ³	1.219	0.886	0.966	0.796	0.207	100 %
PoNQ-lite	64 ³	1.074	0.888	0.978	0.128	0.858	100 %
PoNQ	64 ³	0.886	0.892	0.980	0.109	0.866	100 %

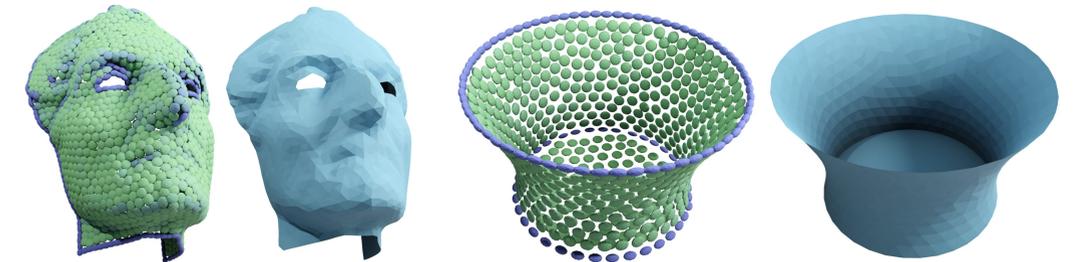
Extension 1: Output Mesh Simplification



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

$$P_i \leftarrow \frac{1}{P} \sum P_i, \quad n_i \leftarrow \frac{1}{P} \sum n_i, \quad Q_i \leftarrow \sum Q_i$$

Extension 2: Open Surfaces (Optimization-based)



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

References

- [1] W. Lorensen and H. Cline. Marching cubes: A high resolution 3D surface construction algorithm. In *ACM SIGGRAPH Computer Graphics*, 1987.
- [2] Z. Chen, A. Tagliasacchi, T. Funkhouser, and H. Zhang. Neural dual contouring. *ACM Transactions on Graphics*, 2022.
- [3] Z. Chen and H. Zhang. Neural marching cubes. *ACM Transactions on Graphics*, 2021.
- [4] N. Maruani, R. Klokov, M. Ovsjanikov, P. Alliez, and M. Desbrun. VoronMesh: 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.



Project Page