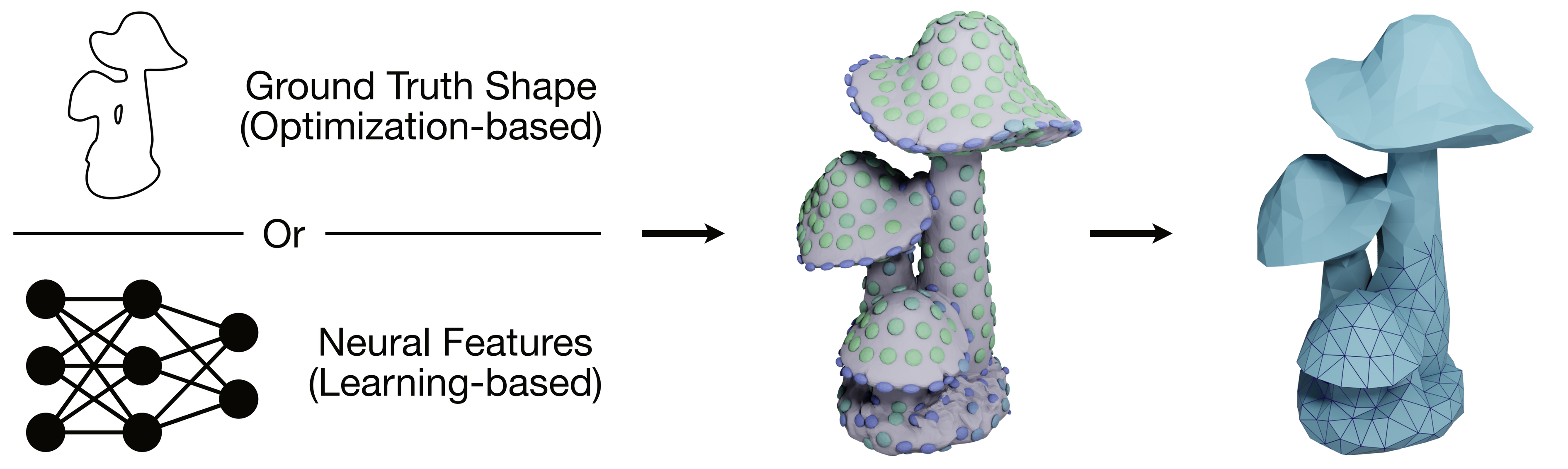
PoNQ: a Neural QEM-based Mesh Representation

Nissim Maruani[1], Maks Ovsjanikov[2], Pierre Alliez[1], Mathieu Desbrun[1,2]

[1]Inria, France [2]École polytechnique, France

CVPR JUNE 17-21, 2024 SEATTLE, WA

Overview and Contributions

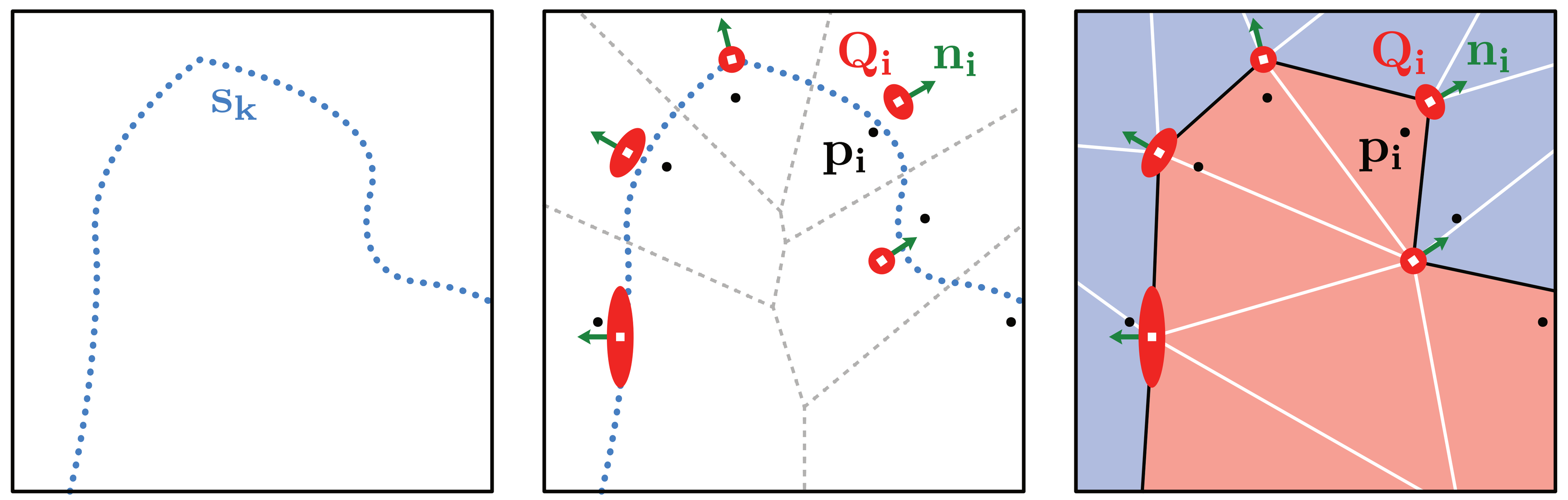


Points, **N**ormals, **Q**uadrics

PoNQ Mesh

- 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



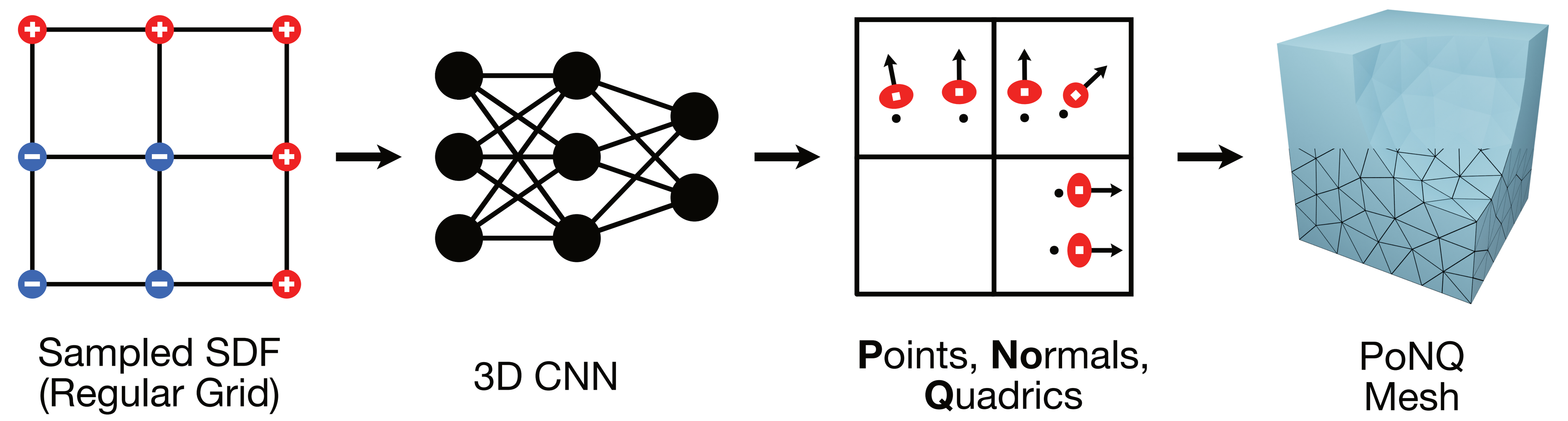
Densely Sampled Target Shape | Optimized **P**oints, **N**ormals, and **Q**uadrics | 3D Delaunay Triangulation and **PoNQ** Mesh

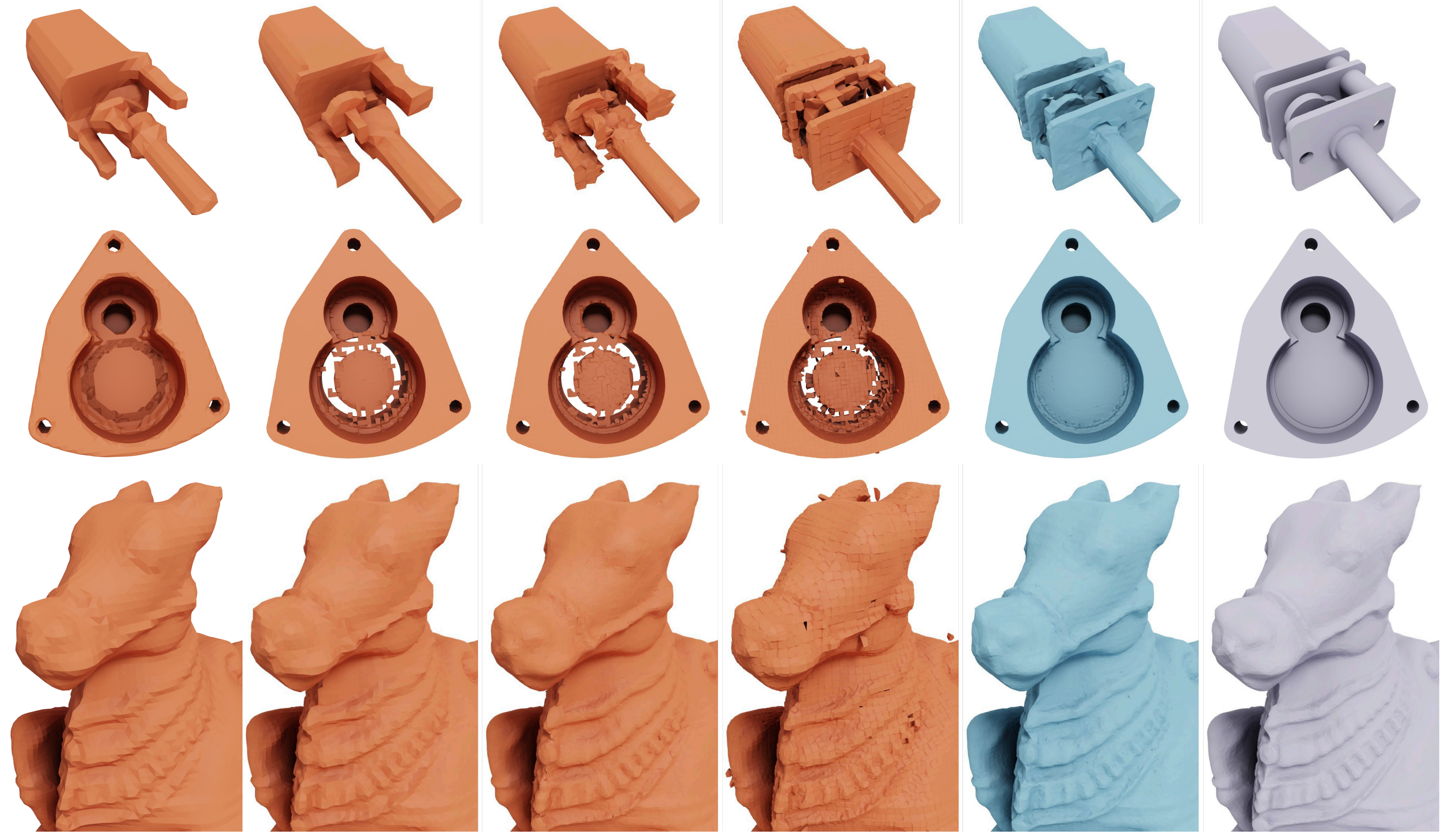
- **P**oints $\mathbf{p_i}$ define spatial regions through their Voronoi diagram.
- **N**ormals $\mathbf{n_i}$ (resp. **Q**uadrics $\mathbf{Q_i}$) represent the average normal $\mathbf{n(s_k)}$ (resp. QEM) within each region $\mathbf{R_i}$. The QEM metric is defined as:

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

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

Application: Reconstruction from SDF grids

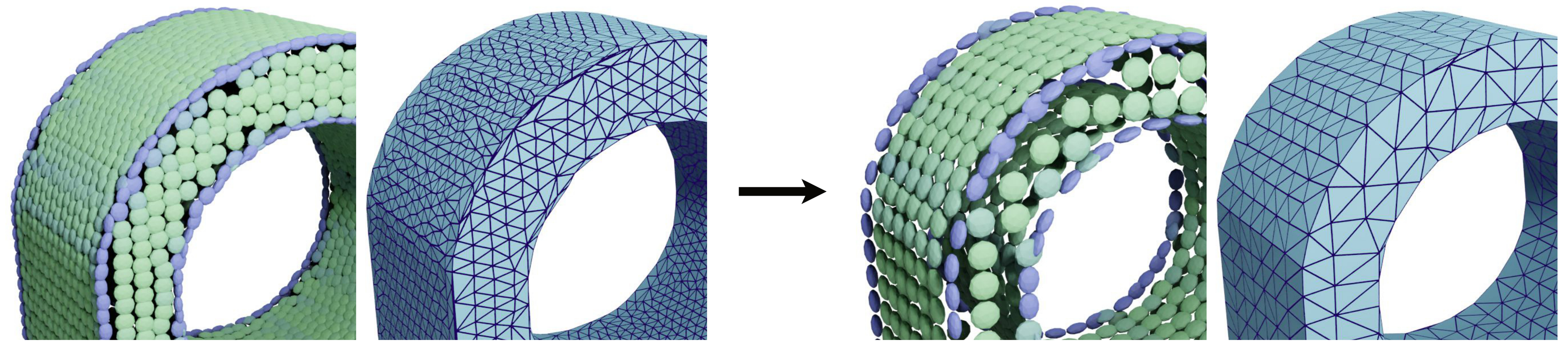




MC [1] | NDC [2] | NMC [3] | VoroMesh [4] | PoNQ | Ground Truth

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 %
VoroMesh [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 %
VoroMesh [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 %**

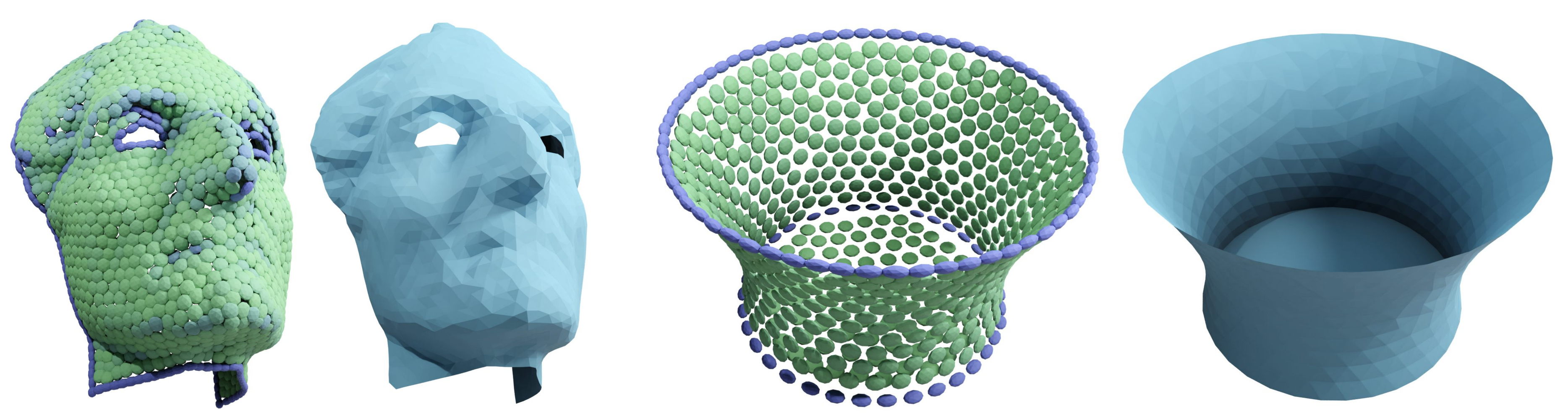
Extension 1: Output Mesh Simplification



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

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

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. 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.

Project Page