Feature-Preserving Mesh Decimation for Normal Integration

Moritz Heep[1] Sven Behnke[1] Eduard Zell[2]

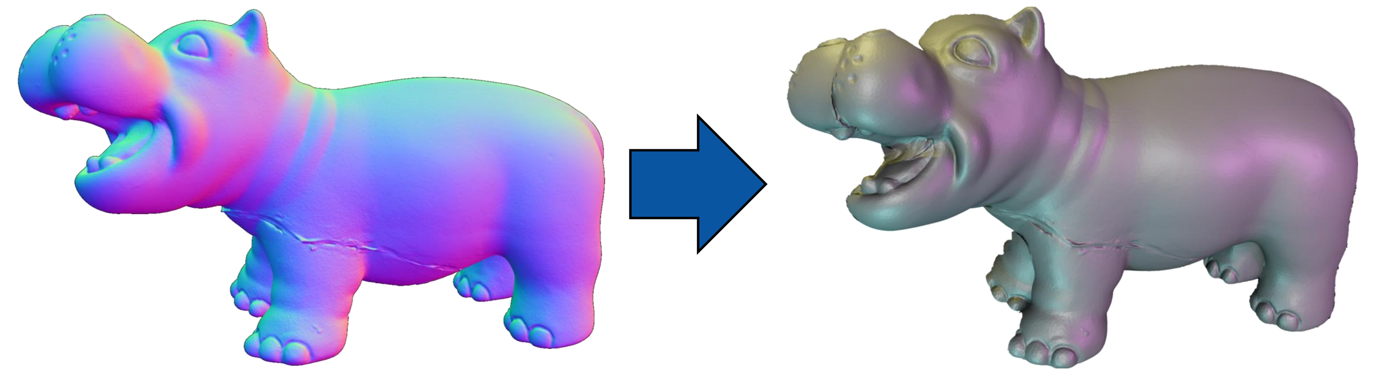
[1]University of Bonn [2]Independent Researcher

CVPR Nashville JUNE 11-15, 2025

Motivation

Normal Integration

- Normal integration calculates **depth maps from normal maps** (e.g. measured through photometric stereo)



Normal Map 2.5D Surface

- Normal integration in photometric stereo becomes **computationally expensive** at high image resolutions.

Mesh-Based Normal Integration

- Replacing the regular pixel grid by a triangle mesh before integration decouples image and geometric resolution (⇒ runtime)

Pixel-Based [1] (21k Pixels @ 256^2) Isotropic Mesh-Based [2] (22k Vertices @ 2048^2) Ours (20k Vertices @ 2048^2)

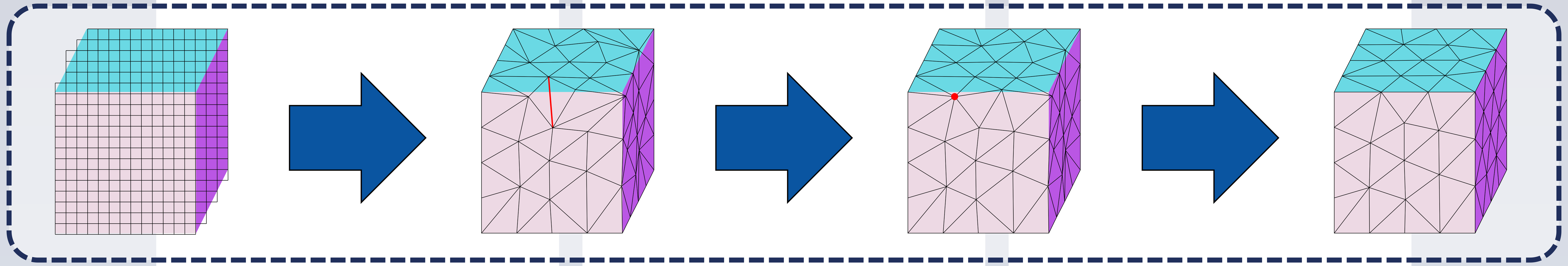
- Careful adaptation of the mesh allows **high quality surfaces at low runtimes** [2].
- By aligning vertices and edges with ridges and furrows of the surface, we maintain better quality at the same level of compression.

Methodology

Decimation

Edge Alignment

Vertex Alignment



Decimation

- **Estimate** influence of vertex on overall shape through quadratic error metric.
- **Remove** the least influential vertices through edge collapses.

Edge Alignment

- **Identify** misaligned edges through generalized Delaunay criterion.
- **Flip** edges to align them with ridges and furrows of the underlying surface.

Vertex Alignment

- **Move** vertices to the position that minimizes the quadratic error metric to align them with ridges and furrows of the surface.

Screen-Space Quadrics

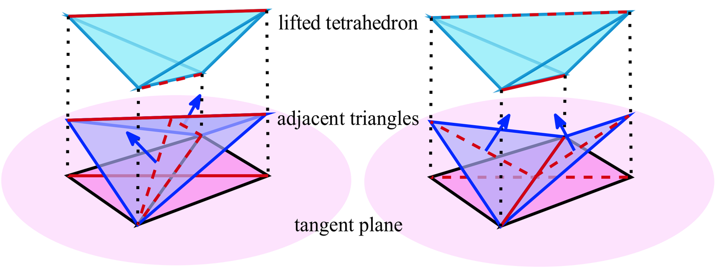
$$E = \sum_{v\in\mathcal{V}} \int_{\mathcal{S}_v} \underbrace{\langle \vec{n}, \vec{x} - \vec{x}_v \rangle^2}_{\text{QEM}} + \lambda \cdot \underbrace{\|\vec{x} - \vec{x}_v\|^2}_{\text{ODT}} \,\mathrm{d}^2 x = \sum_{v\in\mathcal{V}} \int_{\mathcal{S}_v} \underbrace{\|\vec{x} - \vec{x}_v\|_M^2}_{\|\vec{x}\|_M^2 := \sqrt{\langle \vec{x}, M\vec{x} \rangle}} \,\mathrm{d}^2 x$$

$\mathcal{S}_v$: „Star of v“

$$M := \vec{n} \cdot \vec{n}^t + \lambda \cdot I$$

- **Quadratic Error Metrics (QEM [4]):** Estimates deviation between the mesh and the underlying surface.
- **Optimal Delaunay Triangulations (ODT [3]):** Measures regularity of the mesh.
- We derive mathematical tools to calculate the quadrics in screen-space, i.e. from normal maps alone.

- QEM and ODT can be unified through the generalized norm $\|\cdot\|_M$.
- This also generalizes the Delaunay flip algorithm for flat surfaces [3].
- We develop a local version for curved surfaces.



Mesh-Based Integration

$$E_{\text{Int}} = \int_{\Omega} (\vec{n} \cdot \vec{r}\, \partial_u z + f^{-1} \cdot n_x)^2 + \left(\vec{n} \cdot \vec{r}\, \partial_v z + f^{-1} \cdot n_y\right)^2 \mathrm{d}u\, \mathrm{d}v$$

$\vec{r}$: camera ray, f: focal length, z: (log) depth

- We improve the discretization of this energy for triangle meshes, cf. [2].
- As in the pixel case, integration is performed by solving a **linear system but with much fewer variables**.

References

[1] M. Heep and E. Zell, ‘ShadowPatch: Shadow Based Segmentation for Reliable Depth Discontinuities in Photometric Stereo’, *Pacific Graphics,* 2022.

[2] M. Heep and E. Zell, ‘An Adaptive Screen-Space Meshing Approach for Normal Integration’, *ECCV,* 2024.

[3] L. Chen and J. Xu, ‘Optimal delaunay triangulations’, *Journal of Computational Mathematics*.

[4] M. Garland and P. S. Heckbert, ‘Surface simplification using quadric error metrics’, *SIGGRAPH,* 1997.

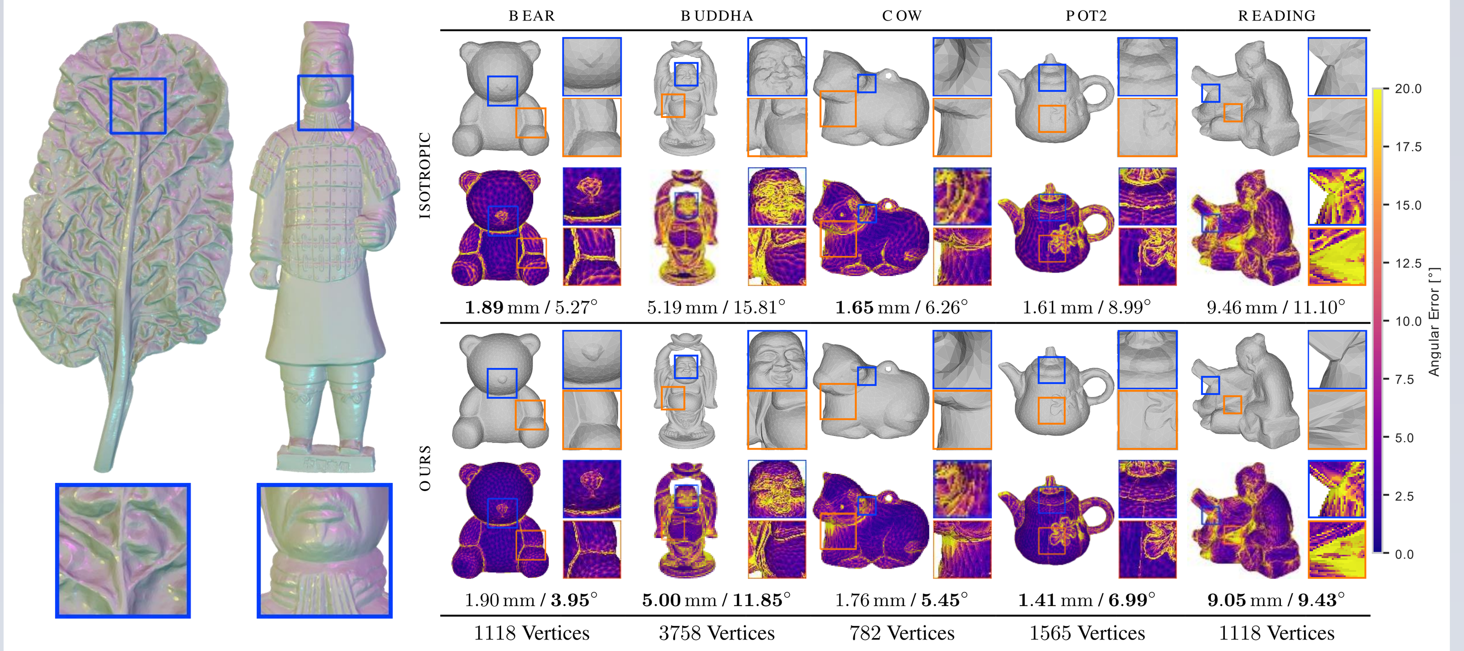
[5] R. Mecca, F. Logothetis, I. Budvytis, and R. Cipolla, ‘LUCES: A Dataset for Near-Field Point Light Source Photometric Stereo’, *BMVC*, 2021.

[6] M. Li, Z. Zhou, Z. Wu, B. Shi, C. Diao, and P. Tan, ‘Multi-view photometric stereo: A robust solution and benchmark dataset for spatially varying isotropic materials’, *Transactions on Image Processing,* 2020.

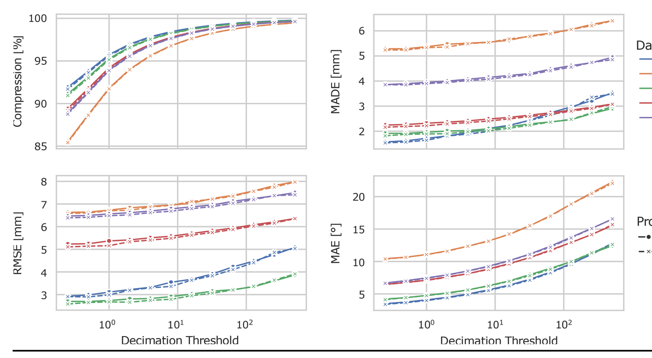
[7] C. Toler-Franklin, A. Finkelstein, and S. Rusinkiewicz, ‘Illustration of complex real-world objects using images with normals’, *International symposium on Non-photorealistic animation and rendering*, 2007.

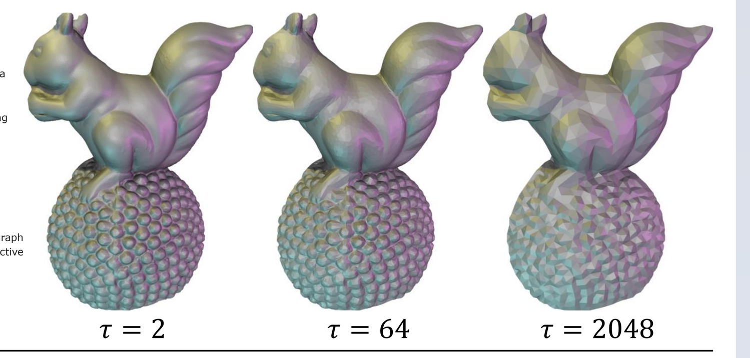
Results

Improved Quality



Adjustable Compression



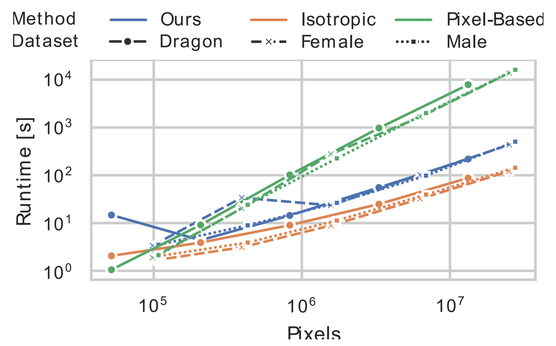


The mesh density is controlled either by:

- **directly** setting the desired number of vertices or
- **indirectly** setting a threshold τ on the error introduced by decimation.

Lower thresholds reduce angular and absolute errors.

Faster Runtimes



- A reduced number of vertices compared to pixels leads to a smaller linear system and hence **faster normal integration**
- Break-even point of the additional step is reached below 1MP.

Moritz Heep

Niebuhrstraße 1a
53113 Bonn
+49 228/73-60834
moritzheep@gmx.de
moritzheep.github.io

Project Page

Seeking PostDoc opportunities

This work has been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy, EXC-2070 - 390732324 (PhenoRob).

PHENOROB