Differentiable Micro-Mesh Construction Supplemental Material

Yishun Dou[2] Zhong Zheng[2] Qiaoqiao Jin[1] Rui Shi[1] Yuhan Li[1] Bingbing Ni[1,2]
[1]Shanghai Jiao Tong University, Shanghai 200240, China [2] Huawei
yishun.dou@gmail.com nibingbing@sjtu.edu.cn

1. Additional Methodology Details

Update Rule for Displacements. The *scalar-valued* displacement follows the similar update rule as the vector-valued offset. We first rewrite the Eq.(8) (main text) by applying chain rule:

$$\overline{\mathcal{D}}^{\ell} \leftarrow \overline{\mathcal{D}}^{\ell} - \eta_2 \frac{\partial_{\mathcal{D}^{\ell}}}{\partial_{\overline{\mathcal{D}}^{\ell}}} \frac{\partial_{\mathcal{V}^{\ell}}}{\partial_{\mathcal{D}^{\ell}}} \frac{\partial_{\Phi}}{\partial_{\mathcal{V}^{\ell}}}, \tag{1}$$

where $\partial_{\mathcal{D}^{\ell}}/\partial_{\overline{\mathcal{D}}^{\ell}} = (\mathbf{I} + \lambda_2 \mathbf{L}^{\ell})^{-1}$ is the Jacobian of displacements $\mathcal{D}^{\ell}$ as a function of the reparameterization $\overline{\mathcal{D}}^{\ell}$. We can then derive the following update rule for original parameters $\mathcal{D}^{\ell}$:

$$\begin{aligned} \mathcal{D}^{\ell} &\leftarrow (\mathbf{I} + \lambda_2 \mathbf{L}^{\ell})^{-1} (\overline{\mathcal{D}}^{\ell} - \eta_2 \frac{\partial_{\mathcal{D}^{\ell}}}{\partial_{\overline{\mathcal{D}}^{\ell}}} \frac{\partial_{\mathcal{V}^{\ell}}}{\partial_{\mathcal{D}^{\ell}}} \frac{\partial_{\Phi}}{\partial_{\mathcal{V}^{\ell}}}) \\ &= \mathcal{D}^{\ell} - \eta_2 (\mathbf{I} + \lambda_2 \mathbf{L}^{\ell})^{-2} \frac{\partial_{\mathcal{V}^{\ell}}}{\partial_{\mathcal{D}^{\ell}}} \frac{\partial_{\Phi}}{\partial_{\mathcal{V}^{\ell}}}. \end{aligned} \tag{2}$$

Quasi-Newton Optimization. The update rules in Eq. (2) and Eq.10 (main text) are analogous to the Newton's optimization, which is consistent with the common practice of applying second-order optimization on Laplacian representation in geometry processing [1, 2, 6, 10–12]. Our μ-mesh construction is developed along the research line of conventional geometry optimization.

Adaptive Level of Detail. With the visual-guided subdivision stopping criterion $\mathsf{SubdivisionCriterion}(\cdot)$ that measures the visual improvement bring from a finer level, adaptive level of detail (LOD) can be achieved with further one-ring neighborhood lookup. Given a μ-mesh $\mathcal{M}$, we introduce $\mathsf{L} \in \mathbb{N}^{N_{\mathcal{F}}}$ to maintain the levels of base triangles in $\mathcal{M}$, which is determined by the adaptive LOD algorithm. We provide two choices to achieve adaptive LOD.

In-Optimization adaptive LOD, as shown in Algorithm 1. There is no limit to the subdivision before level ℓ_{min}. Starting from ℓ_{min}, a base triangle is continue subdivided from level ℓ to $\ell + 1$ when (*i*) no adjacent triangle terminates subdivision at previous level $\ell - 1$, and (*ii*) the gains from current level is greater than a small acceptance number. This algorithm is called before each subdivision.

Adap. LOD	μ-mesh PSNR↑	μ-mesh Volume↓	Base mesh Geom. Err.↓	Base mesh Iso.↑	Time↓	Storage↓
In-Optim. CD	44.78	0.085	1.81	0.747	6.3m	3.43MB
In-Optim.	44.89	**0.083**	1.80	0.747	**4.8**m	**3.12**MB
Post	**44.92**	0.084	1.80	0.748	6.8m	3.17MB

(a) Ablation studies on adaptive level of detail. In-Optim.(CD) in table denotes replacing the visual-guided subdivision stopping criterion with Chamfer Distance.

Smoothing Strategy	μ-mesh PSNR↑	μ-mesh Volume↓	Base mesh Geom. Err.↓	Base mesh Iso.↑
None	35.41	0.142	2.42	0.619
Laplacian Smoothing (0.1)	38.61	0.101	1.95	0.729
Laplacian Smoothing (1.0)	42.25	0.087	1.85	0.740
Laplacian Smoothing (10)	41.24	**0.079**	1.86	**0.752**
Reparameterization	**44.89**	0.083	**1.80**	0.747

(b) Ablation studies on smoothing strategies. An oversmoothed μ-mesh, produced by Laplacian smoothing with large loss weight, can have small shell volume.

λ_1	λ_2	μ-mesh PSNR↑	μ-mesh Volume↓	Base mesh Geom. Err.↓	Base mesh Iso.↑
9	9	44.85	0.084	1.84	0.744
19	9	**44.89**	**0.083**	**1.80**	0.747
19	19	44.81	0.085	**1.80**	0.747
49	19	44.79	0.089	1.86	**0.749**

(c) Ablation studies on λ_1 and λ_2 in reparameterization.

Training View Number	μ-mesh PSNR↑	μ-mesh Volume↓	Base mesh Geom. Err.↓	Base mesh Iso.↑	Time↓
16	44.67	0.082	1.83	0.747	4.2m
24	44.89	0.083	1.80	0.747	4.8m
32	45.01	0.083	1.79	0.748	5.6m
40	44.99	0.082	1.80	0.748	6.8m

(d) Ablation studies on training view number. We compute rendering losses for all views at each training iteration.

Table 1. **More studies on adaptive level of detail, smoothing strategies, hyperparameter in reparameterization, and training view numbers.** These ablation studies are experimented on the extreme subset. The underlined values denote the default configurations used in our experiments. (Geom. Err. $\times 10^{-5}$)

ALGORITHM 1: In-Optimization Adaptive LOD

Input: μ-mesh $\mathcal{M}$ (to be subdivided), base triangle levels $\mathsf{L} \in \mathbb{N}^{N_\mathcal{F}}$, current level ℓ, minimum level ℓ_{min}.
Output: Base triangle levels L.

if $\ell < \ell_{min}$ **then**
  for *each base triangle* f ***in*** *μ-mesh* $\mathcal{M}$ **do**
    $\mathsf{L}[\mathsf{f}] \leftarrow \ell + 1$
  end
end
return L

$\tau \leftarrow 0.1$ (a small acceptance number)
for *each base triangle* f ***in*** *μ-mesh* $\mathcal{M}$ **do**
  if $\mathsf{L}[\mathsf{f}] == \ell$ ***and***
    $\mathsf{L}[\texttt{OneRingNeighborhood}(\mathsf{f})] \geq \ell$ ***and***
    $\texttt{SubdivisionCriterion}\,(\mathcal{M}, \ell, \mathsf{f}) > \tau$
  then
    $\mathsf{L}[\mathsf{f}] \leftarrow \ell + 1$
  end
end
return L

ALGORITHM 2: Post-Process Adaptive LOD

Input: μ-mesh $\mathcal{M}$ with uniform subdivision, base triangle levels $\mathsf{L} \in \mathbb{N}^{N_\mathcal{F}}$, maximum level ℓ_{max}, minimum level ℓ_{min}.
Output: Base triangle levels L.

$\ell \leftarrow \ell_{\text{max}}$
$\tau \leftarrow 0.03$ (a small tolerance number)
for *each base triangle* f ***in*** *μ-mesh* $\mathcal{M}$ **do**
  if $\texttt{SubdivisionCriterion}\,(\mathcal{M}, \ell, \mathsf{f}) < \tau$ **then**
    $\mathsf{L}[\mathsf{f}] \leftarrow \ell - 1$
  end
end
$\ell \leftarrow \ell - 1$
while $\ell > \ell_{min}$ **do**
  for *each base face* f ***in*** *μ-mesh* $\mathcal{M}$ **do**
    if $\mathsf{L}[\mathsf{f}] == \ell$ ***and***
      $\mathsf{L}[\texttt{OneRingNeighborhood}(\mathsf{f})] <= \ell$ ***and***
      $\texttt{SubdivisionCriterion}\,(\mathcal{M}, \ell, \mathsf{f}) < \tau$
    then
      $\mathsf{L}[\mathsf{f}] \leftarrow \ell - 1$
    end
  end
  $\ell \leftarrow \ell - 1$
end
return L

Post-Process adaptive LOD, as shown in Algorithm 2. This algorithm is called once within a post-processing after optimization. Similarly, a base triangle at level ℓ rolls back to $\ell - 1$ when (*i*) no adjacent triangle terminates rollback at level $\ell+1$, and (*ii*) the gains from current level is negligible, *i.e.* less than a small tolerance number. Note that the logical differences between these two algorithms are inevitable. The first algorithm, for example, determines a subdivision to level $\ell + 1$ depending on the previous two levels ℓ and $\ell - 1$, because of the μ-mesh at level $\ell + 1$ is not available for now. Instead, all levels of rendering errors are available for the second one. We demonstrate the comparison among these two choices in Tab. 1a.

2. Additional Implementation Details

Camera Settings. We first scale normalize and center the meshes to fit in sphere of radius 1 centered at $(0, 0, 0)$. In order to arrange the cameras such that they evenly surround the mesh, a Fibonacci sphere [3] is chosen to determine the camera origin. Figure 1 demonstrates a Fibonacci sphere with hundreds points number (for the sense of a sphere). As an immediate benefit, the sensor number, the sphere radius, the fov and so on can be easily configured. Specifically, we experiment with the following configurations: *(i)* radius is 3.5, (ii) cameras look at $(0, 0, 0)$, and (iii) the look up direction is determined by first finding the sphere tangent plane and then finding the direction on this plane with the largest z-axis component. The influence of the training view number is shown in Tab. 1d. In practice, we find 24 views are sufficient for the models in our database.

Textures. For the μ-mesh with aggressive decimation rate, the base mesh has a large geometric difference from the high-polygon mesh. The UV coordinates can be finetuned after our μ-mesh construction, which can also be formulated within a differentiable rendering framework based on existing works on UV optimization [4, 5]. Instead of jointly optimizing the geometry and UV coordinates (or texture map), we choose to tackle these two problems separately, based on the consideration of the intractably ill-posed nature of inverse rendering.

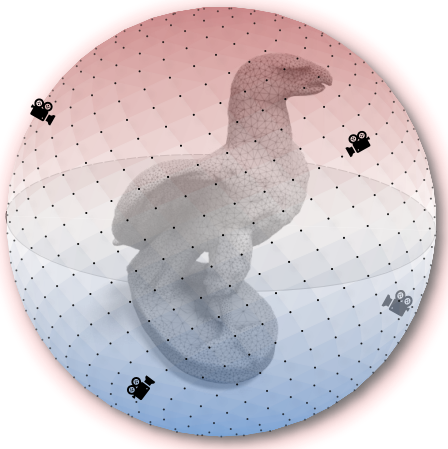

Figure 1. **Cameras are arranged on a Fibonacci sphere surface.** A Fibonacci sphere can evenly arrange any number of points on a sphere surface.

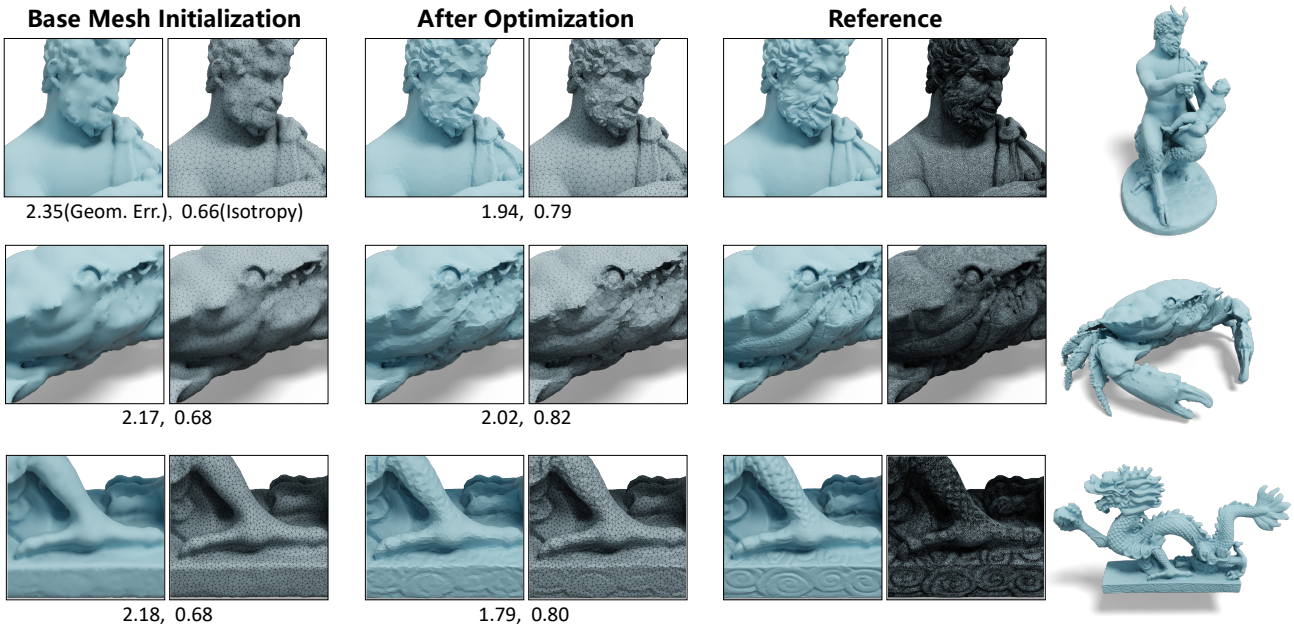

Figure 2. **Comparison of the base mesh at initialization and after optimization.** The base mesh of a μ-mesh is optimized together with displacements baking. (Geom. Err. $\times 10^{-5}$)

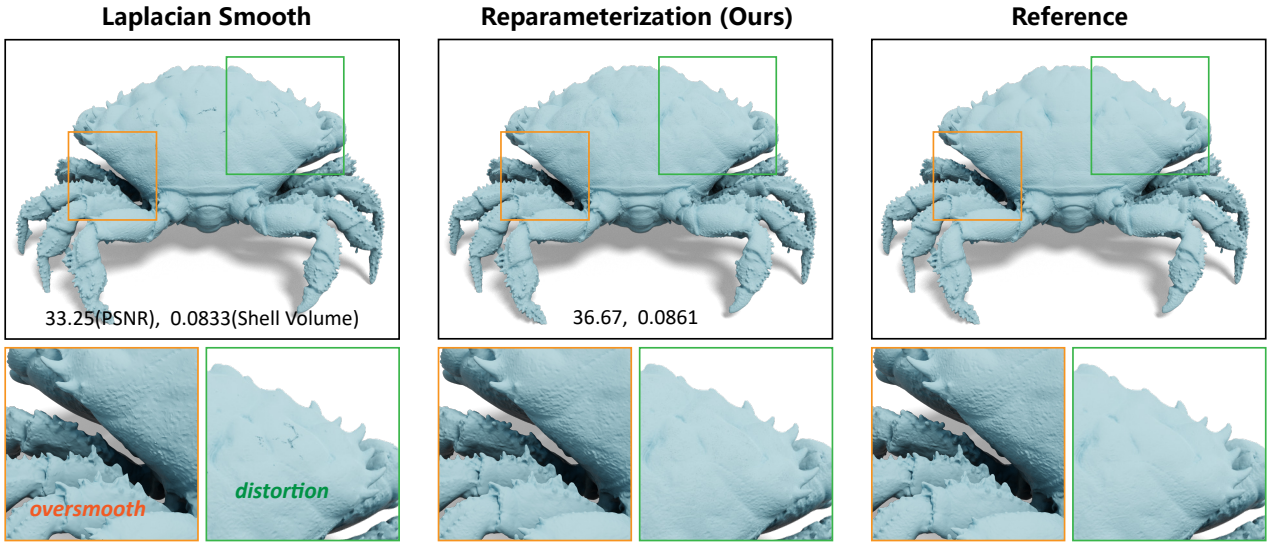

Figure 3. **Comparison of Laplacian smoothing and Reparameterization.**

Smaller Displacements. In μ-mesh compression, the displacement at level ℓ acts as a signed correction for the interpolated displacement at level $\ell - 1$, namely, hierarchical delta compression [9]. This design enables lower bitrate compression for displacements at higher levels. During optimization, we still use the *actual* displacements instead of the hierarchical delta compression format, for the compatibility with our variance-based shell volume minimization objective $\mathcal{L}_{sv}$. The compression format can be easily obtained when saving μ-mesh to disk. Minimizing the $\mathcal{L}_{sv}$ is also promising in producing smaller displacement corrections (in compression format), which in turn reduces the quantization losses.

3. Additional Experiments

Ablation Setups in Main Text. For each line in Tab.2 (main text): *(i)* We directly subdivide the base mesh up to the maximum level ℓ_{max} before optimization. The training

iterations and learning rate scheduler remain unchanged. *(ii)* The reparameterization is replaced with an additional Laplacian smooth term with a weight of 1.0. Table 1b shows more comparisons on smooth term. *(iii, iv)* We simply remove the loss term $\mathcal{L}_{sv}$ or $\mathcal{L}_{iso}$. *(v)* The visual-guided subdivision stopping criterion is replaced with Chamfer Distance.

Base Mesh after Optimization. Our method only relies on a rough base mesh initialization. To meet the requirements of high quality μ-mesh, such as small shell volume, the base mesh is deformed along with displacements baking. We demonstrate comparison between the rough initialization and the optimized result in Fig. 2. The isotropy gets better after optimization, suggesting the effectiveness of the as-equilateral-as-possible constraint $\mathcal{L}_{iso}$. Indeed, our approach can benefit from better initialization when better mesh decimation algorithms emerge in the future.

Laplacian Smoothing *v.s.* Reparameterization. We demonstrate the quantitative comparison between Laplacian smoothing and reparameterization in Tab. 1b. As indicated, Laplacian smoothing lags behind reparameterization for all experimented loss weights, among which the weight 1 shows the relatively best results. Figure 3 illustrates a qualitative comparison between Laplacian smoothing with weight 1 and reparameterization. The constructed μ-mesh with Laplacian smoothing exhibits oversmooth and distortion artifacts.

Influence of the λ in Reparameterization. The tunable parameters λ_1 and λ_2 control the implicit time of the diffusion process. As indicated by Nicolet et al. [8], large values of λ dampen high-frequencies, which can impede the effectiveness to reconstruct local details. Different from them that all mesh vertices can freely move, in our case, the μ-vertices are restricted to move along normal direction with scalar-valued displacements. Due to the small degree of freedom of μ-vertices, together with the variance-based shell volume constraint, μ-vertices may have a smaller distortion impact than vector-valued base vertices. Based on this insight, the values of λ_1 and λ_2 are searched independently. We investigate the influence of λ_1 and λ_2, as shown in Tab. 1c. We experimentally find a configuration of $\lambda_1 = 19$ and $\lambda_2 = 9$ shows the best results.

4. Discussion and Future Work

Self-intersection Free and **Poor-isotropy-face Free.** We devise several strategies to meet these properties during μ-mesh construction: ***(i) Reparameterization.*** The diffusion reparameterization can effectively bias gradient steps towards smooth solutions (without requiring the final solution to be oversmoothed) [8], as shown in Tab. 1b and Fig. 3. ***(ii)*** *Larger λ for base mesh.* Larger λ impose smoothness assumption. We only assign smaller λ for tessellated μ-mesh to reconstruct local details (shown in Tab. 1c), due to the two properties are mainly related to the base mesh. ***(iii)*** *Gradient clip.* We employ gradient clip in our implementation to prevent the exploding gradients. ***(iv)*** *Isotropy regularizer,* introduced in Sec.3.4.2 and validated in Sec.4.3. Nevertheless, the gradient-based optimization cannot guarantee the fulfillment of these restrictions. Further studies on imposing strict restrictions at some training milestone is useful in handling potential corner cases.

Topology. One of limitation of our method is that it doesn't support topology modification during optimization, and thus it relies on a topology-preserving or genus-preserving mesh decimation. However, aggressive decimation with inadequate triangles inevitably breaks the original topology, so an interaction process with the artist is required to intervene the simplification process. In the future, we will add the support of handling topology changes [7], which can further increase the degree of automation of μ-mesh construction.

Concave Structures. Another limitation is that the cameras are located on a bounding sphere surface, which may cause the geometry optimization neglects deep concave structures. A possible solution is to modify the camera trajectory, determined by some measurements of concaveness or the visibility of base triangles via multi-view rasterization. We consider this is a common issue for most differentiable rendering applications, and we will further investigate this along the above possible avenues.

References

[1] Mario Botsch, Leif Kobbelt, Mark Pauly, Pierre Alliez, and Bruno Lévy. *Polygon mesh processing*. CRC press, 2010. 1

[2] Sebastian Claici, Mikhail Bessmeltsev, Scott Schaefer, and Justin Solomon. Isometry-aware preconditioning for mesh parameterization. In *Computer Graphics Forum*, pages 37–47. Wiley Online Library, 2017. 1

[3] Álvaro González. Measurement of areas on a sphere using fibonacci and latitude–longitude lattices. *Mathematical Geosciences*, 42(1):49–64, 2010. 2

[4] Jon Hasselgren, Jacob Munkberg, Jaakko Lehtinen, Miika Aittala, and Samuli Laine. Appearance-driven automatic 3d model simplification. In *EGSR (DL)*, pages 85–97, 2021. 2

[5] Julian Knodt, Zherong Pan, Kui Wu, and Xifeng Gao. Joint uv optimization and texture baking. *ACM Transactions on Graphics*, 43(1):1–20, 2023. 2

[6] Shahar Z Kovalsky, Meirav Galun, and Yaron Lipman. Accelerated quadratic proxy for geometric optimization. *ACM Transactions on Graphics (TOG)*, 35(4):1–11, 2016. 1

[7] Ishit Mehta, Manmohan Chandraker, and Ravi Ramamoorthi. A level set theory for neural implicit evolution under explicit flows. In *European Conference on Computer Vision*, pages 711–729. Springer, 2022. 4

[8] Baptiste Nicolet, Alec Jacobson, and Wenzel Jakob. Large

steps in inverse rendering of geometry. *ACM Transactions on Graphics (TOG)*, 40(6):1–13, 2021. 4

[9] NVIDIA. Nvidia ada gpu architecture, 2022. https://www.nvidia.com/it-it/geforce/ada-lovelace-architecture, https://images.nvidia.com/aem-dam/Solutions/geforce/ada/ada-lovelace-architecture/nvidia-ada-gpu-architecture-whitepaper-1.03.pdf. 3

[10] Michael Rabinovich, Roi Poranne, Daniele Panozzo, and Olga Sorkine-Hornung. Scalable locally injective mappings. *ACM Transactions on Graphics (TOG)*, 36(4):1, 2017. 1

[11] Chris Yu, Henrik Schumacher, and Keenan Crane. Repulsive curves. *ACM Transactions on Graphics (TOG)*, 40(2):1–21, 2021.

[12] Yufeng Zhu, Robert Bridson, and Danny M Kaufman. Blended cured quasi-newton for distortion optimization. *ACM Transactions on Graphics (TOG)*, 37(4):1–14, 2018. 1