

Improving Multi-View Reconstruction via Texture-Guided Gaussian-Mesh Joint Optimization

Supplementary Material

6. Remesh Algorithm

In this section, we provide a detailed description of the edge operations used in our mesh processing framework. Specifically, we discuss three fundamental operations: Edge Split, Edge Collapse, and Edge Flip, which allow for dynamic modification of the mesh topology while preserving geometric and color continuity. We demonstrated the detailed algorithm in Algorithm 1.

Algorithm 1 Iterative Remeshing

Require: Mesh vertices and gradient features $\mathcal{V}_{etc} \in \mathbb{R}^{V \times D}$, vertex colors $\mathcal{C} \in \mathbb{R}^{V \times 3}$, vertex texture density $F_l \in \mathbb{R}^V$, faces $\mathcal{F} \in \mathbb{Z}^{F \times 3}$, edge length tolerance ϵ , flip flag β_{flip} , color gradients $\nabla \mathcal{C}$, and max vertices V_{max} .

Ensure: Remeshed vertices \mathcal{V}_{etc} , colors \mathcal{C} , vertex texture density F_l , faces \mathcal{F} , and color gradients $\nabla \mathcal{C}$.

```

1:  $L_{ref} \leftarrow \mathcal{V}_{etc}[:, -1]$ 
2:  $L_{min} \leftarrow L_{ref} \cdot (1 - F_l) \cdot (1 - \epsilon)$ 
3:  $L_{max} \leftarrow L_{ref} \cdot (1 - F_l) \cdot (1 + \epsilon)$ 
4: — Edge Collapse —
5:  $\mathcal{V} \leftarrow \mathcal{V}_{etc}[:, :3]$ 
6:  $\mathcal{E}, \mathcal{F}_E \leftarrow \text{CalculateEdges}(\mathcal{F})$ 
7:  $L_E \leftarrow \text{CalculateEdgeLengths}(\mathcal{V}, \mathcal{E})$ 
8:  $\mathcal{P}_{collapse} \leftarrow \text{CalculateFaceCollapses}(\mathcal{V}, \mathcal{F}, \mathcal{E}, \mathcal{F}_E, L_E, L_{min})$ 
9:  $S \leftarrow \max(0, 1 - L_E / \text{mean}(L_{min}[\mathcal{E}])) \triangleright \text{Shortness term}$ 
10:  $\mathcal{P}_{priority} \leftarrow \mathcal{P}_{collapse} + S$ 
11:  $\text{CollapseEdges}(\mathcal{V}_{etc}, \mathcal{C}, F_l, \mathcal{F}, \mathcal{E}, \mathcal{P}_{priority}, \nabla \mathcal{C})$ 
12: — Edge Split —
13: if  $|\mathcal{V}_{etc}| < V_{max}$  then
14:    $\mathcal{E}, \mathcal{F}_E \leftarrow \text{CalculateEdges}(\mathcal{F})$ 
15:    $\mathcal{V} \leftarrow \mathcal{V}_{etc}[:, :3]$ 
16:    $L_E \leftarrow \text{CalculateEdgeLengths}(\mathcal{V}, \mathcal{E})$ 
17:    $S_{split} \leftarrow L_E > \text{mean}(L_{max}[\mathcal{E}])$ 
18:    $\text{SplitEdges}(\mathcal{V}_{etc}, \mathcal{C}, F_l, \mathcal{F}, \mathcal{E}, \mathcal{F}_E, S_{split}, \nabla \mathcal{C})$ 
19: end if
20: — Edge Flip —
21:  $\mathcal{V} \leftarrow \mathcal{V}_{etc}[:, :3]$ 
22: if  $\beta_{flip}$  then
23:    $\mathcal{E}, \mathcal{F}_E \leftarrow \text{CalculateEdges}(\mathcal{F})$ 
24:    $\text{FlipEdges}(\mathcal{V}, \mathcal{F}, \mathcal{E}, \mathcal{F}_E)$ 
25: end if

```

7. Loss Function

In this section, we provide a detailed explanation of the loss functions used in our framework. These losses enforce photometric consistency, geometric accuracy, and regularization for stable optimization.

The RGB loss L_{rgb} is defined as:

$$\mathcal{L}_{rgb} = \frac{1}{s} \sum_{i=1}^s \left(\alpha \|I_i - \hat{I}_i\| + (1 - \alpha) \text{SSIM}(I_i, \hat{I}_i) \right) \quad (11)$$

where I_i and \hat{I}_i are the predicted and ground-truth RGB images. α is set to 0.8 following [17].

The normal and depth map loss \mathcal{L}_{geo} can be defined as:

$$\mathcal{L}_{geo} = \frac{1}{s} \sum_{i=1}^s \left(\|n_i - \hat{n}_i\| + \|d_i - \hat{d}_i\| \right) \quad (12)$$

where n_i and \hat{n}_i are the predicted and pseudo-ground-truth normal maps, and d_i and \hat{d}_i are the predicted and pseudo-ground-truth normal maps, respectively.

The regularization loss \mathcal{L}_{reg} includes both Laplacian smoothing and mesh normal consistency:

$$\mathcal{L}_{reg} = \frac{1}{n} \sum_{i=1}^n \|Lv_i\|^2 + \frac{1}{m} \sum_{i=1}^m \|N_i - \bar{N}_i\|^2 \quad (13)$$

where L is the Laplacian matrix, n is the number of vertices, m is the number of faces, v_i are the vertex positions, N_i are the face normals, and \bar{N}_i are the averaged normals of adjacent faces for mesh \mathcal{M} .

8. Vertex-Gaussian Binding

1. **Position:** The position of each Gaussian is directly mapped from the vertex position in the mesh. Let $\mu_i \in \mathbb{R}^3$ represent the position of a gaussian, then:

$$\mu_i = v_i$$

2. **Scale:** The scale of each Gaussian is represented as a vector $\mathbf{s}_i = (s_1, s_2, s_3)$, where each component corresponds to different geometric properties of the mesh around the vertex v_i . Specifically: s_2 is the length of the projection of the longest edge e_{\max} onto the tangent plane at vertex v_i . s_3 is the average projection length of all edges incident to vertex v_i onto the tangent plane.

Finally, s_1 is defined as the average of s_2 and s_3 . Thus, the scale vector \mathbf{s}_i for each Gaussian is composed of these three components s_1, s_2, s_3 , reflecting both the local geometric properties of the vertex and the surrounding mesh structure.

3. **Rotation:** The rotation matrix R_i for each Gaussian is determined by three orthogonal direction vectors: \mathbf{v}_1 is the normal vector at vertex v_i , which is typically computed from the surrounding vertex neighbors and represents the direction perpendicular to the tangent plane at the vertex.

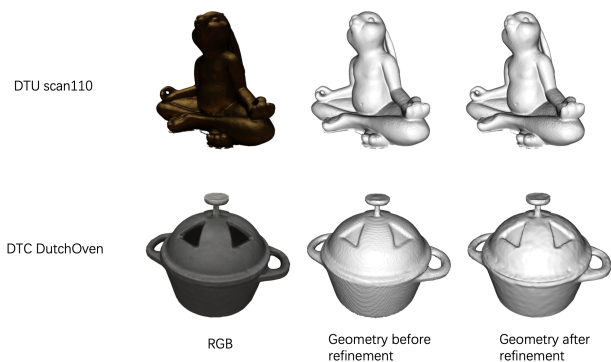


Figure 7. Visualization of failure cases under poor lighting conditions. Left: Case 110 from DTU dataset showing reconstruction artifacts in shadowed regions. Right: DutchOven from DTC dataset demonstrating degraded geometry in low-light condition.

\mathbf{v}_2 is the projection of the longest edge e_{\max} onto the tangent plane at vertex v_i .

\mathbf{v}_3 is the vector that is orthogonal to both \mathbf{v}_1 and \mathbf{v}_2 , ensuring the three vectors form an orthonormal basis. It can be computed as: $\mathbf{v}_3 = \mathbf{v}_1 \times \mathbf{v}_2$.

4. **Opacity:** In our method, we assign a constant opacity value of 0.9 to each Gaussian, assuming that every point on the mesh is visible.

5. **Spherical Harmonics (SH) coefficients:** In our method, we assign the low-order SH coefficients directly from the vertex color c_i , and set the higher-order coefficients to zero.

9. Failure Cases Analysis

While our proposed refinement process consistently demonstrates enhancements in geometric accuracy and detail, its efficacy, like many state-of-the-art methods, is correlated with the quality of the photometric information in the input images. As illustrated in Fig. 7, certain challenging lighting conditions, such as the presence of strong cast shadows or globally low-light environments, can present limitations. In these scenarios, the refinement may yield more subtle improvements or highlight areas for future research in robust reconstruction.

9.1. Strong Shadows

The "scan110" from the DTU dataset provides a valuable case study on the influence of high-contrast lighting. The input RGB image features a dark, specular object under strong directional light, resulting in areas of deep shadow. The initial geometry, shown as "Geometry before refinement," offers a coarse yet largely complete representation of the bunny figure.

Our refinement process demonstrates a clear benefit in the well-illuminated regions. On the figure's head and belly,

for example, the surface is successfully smoothed, and details are sharpened, showcasing the method's effectiveness. In contrast, the regions obscured by shadow—specifically the lap, the underside of the chin, and between the limbs—present a more challenging scenario. The scarcity of reliable photometric cues in these areas makes it difficult for the algorithm, which leverages multi-view consistency, to resolve the geometry with the same level of confidence. This can lead to the introduction of localized surface artifacts. This observation suggests that integrating priors or specialized shadow-handling techniques could be a promising direction for future work to further enhance robustness in extreme lighting.

9.2. Global Low-Light

The "DutchOven" from the DTC dataset illustrates a different set of challenges associated with globally low-light conditions. Here, the input shows a dark, matte object with diffuse, dim illumination, leading to low contrast across the entire surface. The "Geometry before refinement" is of a modest quality, exhibiting a noisy surface where details, like the triangular patterns on the lid, are not yet fully resolved.

In this low signal-to-noise context, the refinement process achieves limited additional improvement over the initial geometry. As shown in "Geometry after refinement," the surface texture remains noisy, and the geometric details on the lid become less defined. This is because the refinement process finds it challenging to distinguish faint surface features from sensor noise in the low-contrast input images. This case highlights that a sufficient level of image quality and contrast is beneficial for achieving optimal results, a characteristic common to many photometric refinement techniques. It also suggests that our method could be further enhanced by coupling it with advanced image pre-processing, such as denoising or contrast enhancement, for inputs captured in such demanding conditions.