SimpliGuard: Robust Mesh Simplification In the Wild

Supplementary Material

Table 2: The distribution of the number of triangles in our collected dataset.

Tris	Count	Min	Max	Median
0 ∼ 50k	52	30.0k	50.0k	37.2k
50k ∼ 100*k*	213	50.0k	100.0k	82.9k
100k ∼ 150*k*	216	100.0k	149.7k	100.3k
150k ∼ 200*k*	88	150.0k	200.0k	174.4k
200k ∼ 400*k*	173	200.0k	400.0k	257.4k
400k ∼ 800*k*	71	405.7k	800.0k	561.5k
≥ 800*k*	71	800.0k	4359.4k	1174.5k

6 THE PROOF OF THE CURVATURE BASED QEM

We adopt the approach proposed in Hoppe [1999] to represent the projected $\hat{c}_i$ as a linear equation for v_i:

$$\hat{c_i} = \mathbf{g}^{\mathbf{T}} v_i + q$$

where $\mathbf{g}$ and q can be solved as follows. Firstly, for a given face $f = (v_1, v_2, v_3)$, the corresponding attributes $\hat{c} = (\hat{c_1}, \hat{c_2}, \hat{c_3})$ can be expressed using the aforementioned formula. Then, for any attribute $\hat{c}$ belonging to a point $\mathbf{v} \in \mathbf{R}^3$, it is equal to the attribute c' of its projection onto the plane of f, which can be expressed as $\mathbf{N}^{\mathbf{T}}\mathbf{g}$ = 0. Thus, $\mathbf{g}$ and q can be computed by:

$$\begin{pmatrix} \mathbf{v}_1^{\mathbf{T}} & 1 \\ \mathbf{v}_2^{\mathbf{T}} & 1 \\ \mathbf{v}_3^{\mathbf{T}} & 1 \\ \mathbf{N}^{\mathbf{T}} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{g} \\ \\ q \end{pmatrix} = \begin{pmatrix} \mathbf{c_1} \\ \mathbf{c_2} \\ \mathbf{c_3} \\ 0 \end{pmatrix}$$

The quadric of the curvature of v_i, $Q_c(\mathbf{v_i}, \mathbf{f_i})$, can be written as: $Q_c(\mathbf{v_i}, \mathbf{f_i}) = (\mathbf{\hat{c}_i} - \mathbf{c_i})^2 = (\mathbf{g}_j^T \mathbf{v_i} + d_j - \mathbf{c_i})^2$. Similar to the original QEM, we can express it as follows:

$$Q_c(\mathbf{v_i}, \mathbf{f_i}) = \left(\left(\begin{array}{c|ccc} \mathbf{g}\mathbf{g}^T & \ddots 0 \ddots & -\mathbf{g} & \ddots 0 \ddots \\ \hline \ddots 0 \ddots & \ddots 0 \ddots & 0 & \ddots 0 \ddots \\ -\mathbf{g}^T & \cdots 0 \cdots & 1 & \cdots 0 \cdots \\ \ddots 0 \ddots & \ddots 0 \ddots & 0 & \ddots 0 \ddots \end{array} \right), \begin{pmatrix} q\mathbf{g} \\ \hline 0 \\ -q \\ 0 \end{pmatrix}, q^2 \right)$$

The final $Q_{\text{total}}(\mathbf{v_i}, \mathbf{f_i}) = Q(\mathbf{v_i}, \mathbf{f_i}) + \alpha Q_c(\mathbf{v_i}, \mathbf{f_i})$, where α is a hyperparameter used to balance the weights of the original QEM and the curvature constraint.

7 THE DISTRIBUTION OF THE NUMBER OF TRIANGLES IN OUR DATASET

Table 2 shows the distribution of the number of triangles in our dataset.

8 EXPERIMENTAL SETUP

We utilize nvdiffrast [Laine et al. 2020] as the differentiable rendering framework. For the first 100 iterations, we only optimize the 3D structure, while after 100 iterations, we concurrently optimize both the 3D structure and texture maps. In total, the framework is trained for 3,000 iterations. The initial learning rate is set to 3e-5, which is reduced to 1e-5 and 3e-6 at the 1,000th and 2,000th iterations, respectively.

The training resolution is set to 512x512, and the texture size is set to 1024x1024. To improve texture quality with a fixed rendering resolution, we normalize the mesh within a sphere of radius 1. Then the following camera motion strategy is introduced: we randomly sample camera coordinates on a sphere with a radius of 1.5 for the first 1,000 iterations. In the subsequent 2,000 iterations, we uniformly sample radius between 0.5 and 1.5 and then randomly sample camera coordinates on the corresponding spheres. This approach enables us to enhance the quality of the learned textures without increasing the rendering resolution.

We train Nvdiff for 100,000 iterations with a training rendering resolution of 512x512. The resolution of the texture is 1024x1024, and the learning rate is set to 0.03. For QEM, QEM++, Blender and simplygon, we use their default parameters.