

3D Character Reconstruction from Hand-drawn Model Sheets - Supplementary Document -

Hyejeong Yoon[✉] Wonjong Jang[✉] Yoonha Hwang[✉] Seungyong Lee[†][✉]

POSTECH, South Korea

In this supplementary document, we provide more details of our method and additional experimental results. Section 1 details the formulations of the regularization terms in deformable per-pixel camera ray optimization and the edge weights for palette-based mesh coloring. Section 2 presents the implementation details including hyperparameter settings. Finally, Section 3 presents the detailed distribution of ratings from our user study.

1. Additional Method Details

1.1. Deformable Per-Pixel Camera Ray: Regularizers

We regularize the per-view deformation using first- and second-order smoothness terms on the B-spline control point grid, together with a Jacobian-based penalty that prevents grid flips.

$$\mathcal{L}_{\text{reg}} = \mathcal{L}_{\text{mem}} + \mathcal{L}_{\text{bend}} + \mathcal{L}_{\text{jac}}, \quad (1)$$

$$\mathcal{L}_{\text{mem}} = \frac{1}{Kn^2} \sum_{k=1}^K \sum_{i,j} \left(\left\| \partial_x \mathbf{C}_{ij}^k \right\|_2^2 + \left\| \partial_y \mathbf{C}_{ij}^k \right\|_2^2 \right), \quad (2)$$

$$\mathcal{L}_{\text{bend}} = \frac{1}{Kn^2} \sum_{k=1}^K \sum_{i,j} \left(\left\| \partial_{xx} \mathbf{C}_{ij}^k \right\|_2^2 + 2 \left\| \partial_{xy} \mathbf{C}_{ij}^k \right\|_2^2 + \left\| \partial_{yy} \mathbf{C}_{ij}^k \right\|_2^2 \right), \quad (3)$$

$$\mathcal{L}_{\text{jac}} = \frac{1}{K|\Omega|} \sum_{k=1}^K \sum_{\mathbf{p} \in \Omega} \max(0, -\det(I + \nabla \mathbf{D}^k(\mathbf{p}))). \quad (4)$$

The membrane and bending terms encourage smooth and low-curvature deformations of the displacement field, while the Jacobian term penalizes negative Jacobian determinants to prevent foldovers and self-intersections.

1.2. Palette-based Mesh Coloring: Edge Weight

The edge weight $w_{t,t'}$ integrates three complementary cues to balance smoothness with boundary preservation:

- *Geometric adjacency weight* w_{geo} : To encourage smooth color propagation over locally planar regions, w_{geo} assigns larger weights to pairs of adjacent triangles in near-planar configurations. We employ standard cotangent weights from geometry processing, computed as $w_{\text{geo}}(t, t') = \frac{1}{2}(\cot \angle(a, v_0, v_1) +$

$\cot \angle(b, v_1, v_0))$, where (v_0, v_1) is the shared edge and a, b are opposite vertices.

- *Line preservation weight* w_{line} : To encourage artist-drawn 2D sketch lines to serve as color boundaries, w_{line} discourages color propagation across line pixels in the decomposed line layer. We set the weight to 0.5 when an edge crosses line pixels in any view, and to 1.0 otherwise.
- *Curvature adaptation weight* w_{curv} : To encourage sharp geometric features to serve as color boundaries, w_{curv} decreases as the normal difference between adjacent triangles increases. We set $w_{\text{curv}}(t, t') = \exp(-\theta(t, t')^2 / 2\sigma_{\text{curv}}^2)$, where $\theta(t, t') = \arccos(\mathbf{n}_t \cdot \mathbf{n}_{t'})$ measures the angle between triangle normals.

The final edge weight is computed as $w_{t,t'} = \text{clip}(w_{\text{geo}} \cdot w_{\text{line}} \cdot w_{\text{curv}}, 0.5, 1.0)$ to ensure numerical stability.

2. Additional Implementation Details

Images were processed at their original resolution during the drawing decomposition and subsequently resized to 1024×1024 for mesh deformation, camera ray optimization, and appearance reconstruction stages. For global color palette construction, we used the HDBSCAN method [MHA17] in the CIELAB color space with a minimum cluster size of 200, and then iteratively merged clusters with the centroid distance less than 15. For the per-view 2D flat-color segmentation, we set the graph-cut regularization weight λ to 1.0. Specifically, we assign a lower edge weight (0.1 ~ 0.5) to pixel pairs involving line pixels of the decomposed line layer to encourage color transitions along these lines, while setting it to 1.0 for non-line pixel pairs. For rigid view alignment and mesh deformation, we employed 256×256 rendering resolution with the Adam optimizer using a step size of 0.01 for 100 iterations.

For deformable per-pixel camera ray optimization, we used 1024×1024 resolution with a 32×32 B-spline control grid and the Adam optimizer with a step size of 0.0001. We employed a two-stage weighting schedule: $\lambda_{\text{cons}} = 0.0$, $\lambda_{\text{reg}} = 10.0$ for the initial 500 iterations to prioritize image alignment, followed by $\lambda_{\text{cons}} = 0.1$, $\lambda_{\text{reg}} = 1.0$ for the subsequent 500 iterations to enforce cross-view consistency. For the label consistency loss, we set the soft extension parameter τ (Eq. (11) in the main paper) to 50 pixels. For palette-based mesh coloring, we set the regularization weight λ (Eq. (15) in the main paper) to 6 and the curvature parameter σ_{curv}

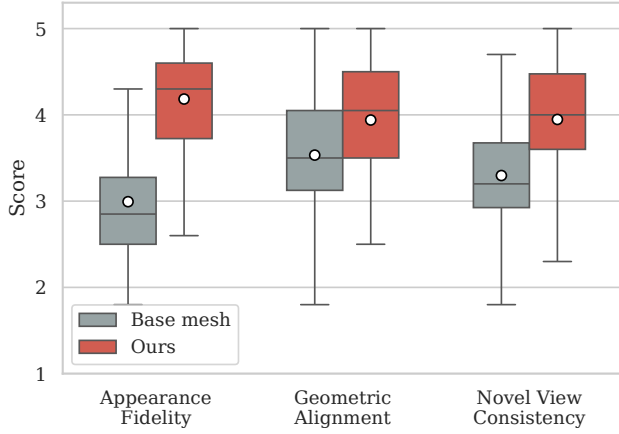


Figure 1: Quantitative results of the user study comparing the base mesh generated by TRELLIS [XLX*25] and our method. The boxes represent the interquartile ranges (25th to 75th percentiles), and the horizontal line inside indicates the median. The separate marker denotes the mean; outliers are omitted for clarity. Our method shows statistically significant improvements across all metrics.

to 30° . For boundary refinement, we performed Laplacian smoothing with 5 iterations and the smoothing weight $\lambda_{lap} = 0.5$.

3. Additional User Study Result

We provide the full distribution of user ratings in Figure 1. The box plots illustrate the spread of scores for appearance fidelity, geometric alignment, and novel view consistency. The overall distributions indicate that our method consistently yields higher scores than the baseline across all metrics. For appearance fidelity and novel view consistency, the interquartile ranges of our method and the baseline show almost no overlap, indicating a distinct difference in perceived quality. For geometric alignment, while the ranges partially overlap, our median and mean scores are clearly above the base mesh.

References

- [MHA17] MCINNES, LELAND, HEALY, JOHN, and ASTELS, STEVE. “hdbscan: Hierarchical density based clustering”. *The Journal of Open Source Software* 2.11 (2017), 205 1.
- [XLX*25] XIANG, JIANFENG, LV, ZELONG, XU, SICHENG, et al. “Structured 3D Latents for Scalable and Versatile 3D Generation”. *Proceedings of the Computer Vision and Pattern Recognition Conference*. 2025, 21469–21480 2.