Skullptor: High Fidelity 3D Head Reconstruction in Seconds with Multi-View Normal Prediction

Supplementary Material

Dataset	Normal Estimator	Depth (mm) ↓	Angular Err. ↓	Avg. Normal Grad. Err. ↓
Multiface	Sapiens 0.3B	5.28	8.75	0.177
	Sapiens 2B	3.79	6.76	0.167
	DAViD	3.74	6.73	0.163
	Skullptor (Ours)	**3.20**	**6.45**	**0.157**
NPHM	Sapiens 0.3B	6.38	8.23	0.124
	Sapiens 2B	6.76	7.91	0.121
	DAViD	2.65	**5.91**	0.116
	Skullptor (Ours)	**2.39**	6.07	**0.112**

Table S1. Ablation study on normal estimator impact on final mesh quality (10 views).

S1. Additional Results

Additional qualitative results for our method, Skullptor, including 4D sequences and extended comparisons with state-of-the-art methods for normal estimation and mesh reconstruction, are available in the supplemental webpage.

S1.1. Impact of the Normal Estimator

To validate the importance of our multi-view normal estimator (Sec. 3.1), we replace it with several monocular baselines (Sapiens 0.3B/2B, DAViD) within our full pipeline. For these monocular methods, we independently predict normals for each of the 10 input views and then run the same inverse rendering optimization to recover the geometry. Since monocular predictors process each view independently without enforcing multi-view consistency, they produce normals that can be geometrically inconsistent across viewpoints, leading to degraded reconstruction quality. As shown in Table S1, our multi-view estimator consistently produces superior geometry. Notably, while Sapiens 2B performed well on the normal comparison evaluation (Table 1), its final geometry is poor (6.76 mm error on NPHM). We attribute this to its tendency to produce overly smooth normals, an observation supported by the high average normal gradient error it achieved in Table 1, which leads to a loss of fine surface details during optimization. This discrepancy demonstrates that multi-view processing during the normal prediction stage translates to more accurate and coherent final geometry.

To complement these quantitative findings, we provide visual comparisons in Figure S1 illustrating how different normal estimators affect the remeshing stage. Monocular predictors exhibit noticeable identity drift and loss of structural fidelity in the reconstructed mesh due to the lack of shared geometric information during view processing. In contrast, our multi-view approach maintains high-frequency details and identity consistency by leveraging cross-view features during the initial estimation phase.

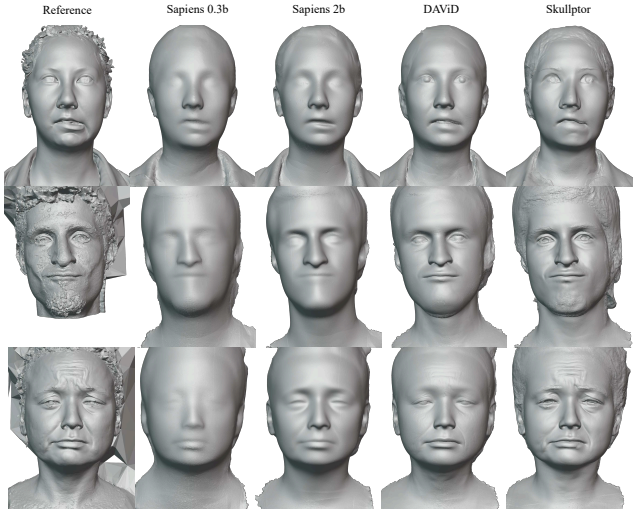

Figure S1. Qualitative impact of normal estimation quality on mesh reconstruction.

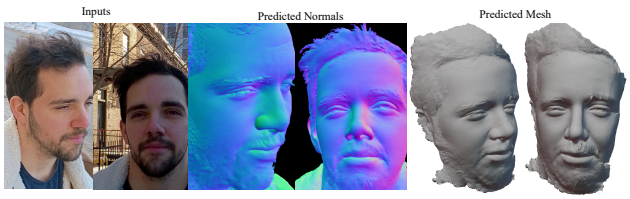

Figure S2. Skullptor results on 12 phone-captured images.

S1.2. Generalization to In-the-Wild Capture

Although our primary evaluation focuses on lightstage capture setups, DAViD's backbone provides robust priors from 300K diverse images that our fine-tuning specializes rather than overwrites. As a result, Skullptor generalizes over four distinct capture setups (TripleGangers, Multiface, NPHM, and ours) with varying camera configurations, lighting, and subjects. Additionally, we include an in-the-wild result in Fig. S2, suggesting potential for generalizing beyond controlled environments.

S2. Implementation Details For Multi-View Normal Prediction

In this section, we provide implementation details for the normal prediction (Sec. 3.1 in the main paper).

S2.1. Training Dataset

Our training dataset is curated from high-quality Triplegangers facial scan assets to ensure diversity across ethnicity, age, and gender. Each scan in the original dataset was captured using a professional light stage setup comprising 55 synchronized high-end cameras, followed by a photogrammetry reconstruction pipeline with manual cleaning to ensure geometrical accuracy. Each subject is captured while performing 20 static facial expressions, resulting in 1,000 unique expression instances across the dataset. The distribution of subjects in our training dataset is given in Table S2.

Table S2. Subject distribution in the Triplegangers training dataset.

	Male	**Female**
Number of subjects	18	32
Age range	18–64	18–75
Mean age	37.2	42.8
Age < 25	9 (50%)	14 (44%)
Age 25–65	8 (44%)	12 (38%)
Age > 65	1 (6%)	6 (19%)

Training-validation split. We reserve 5 subjects (100 expressions) for validation and use the remaining 45 subjects (900 expressions) for training. This subject-level split ensures that the model's generalization is evaluated on entirely unseen identities.

Mesh normalization. To standardize the input geometry across subjects and expressions, we apply Procrustes analysis using 3D facial landmarks to align each scan to a normalized template facial mesh. Specifically, we center each mesh at the origin and scale it to fit within a unit sphere. This normalization ensures consistent scale and orientation across all training samples. Our foreground segmentation allows us to get the full upper part and neck, not only the face.

Rendering setup. For every subject's expression, we render both RGB images and rasterized surface normal maps from 48 virtual cameras at 512×512 resolution using nvdiffrast. The rasterized normal maps are transformed into each camera's local coordinates by multiplying them with the rotation of its model-view matrix. Camera viewpoints are randomly sampled for every expression-subject pair to maximize viewpoint diversity during training. In total, the training set consists of 43200 images.

S2.2. Camera View Generation

As mentioned above, we generate 48 diverse camera views per expression for every subject. The camera viewpoints are selected by randomly sampling from a parameterized camera distribution that ensures comprehensive coverage of the facial region. For the 48 cameras associated with an expression, we construct a camera transformation matrix using the following random sampling of parameters as listed in Table S3. The camera transformation matrix is constructed by combining rotations around the X-axis (pitch) and Y-axis (yaw), followed by translation and scaling operations. Random sampling allows the model to learn from a wide range of viewing angles and distances.

Table S3. Camera Sampling and Projection Parameters

Camera Sampling	
Pitch angle θ_{pitch}	$\mathcal{U}(-35°, 35°)$
Yaw angle θ_{yaw}	$\mathcal{U}(-90°, 90°)$
Initial radius $\mathbf{r}$	2.0
Extra 3D translation $\mathbf{t}$	$\mathcal{U}(-0.2, 0.2)^3$
Scale factor z	$\mathcal{U}(1.0, 1.8)$
Projection Matrix	
Image resolution $w = h$	512 pixels
Focal lengths $f_x = f_y$	512 pixels
Principal point $c_x = c_y$	256 pixels
Depth range (near n, far f)	0.001 – 1000

S2.3. Training Procedure

Each epoch iterates over all training expressions, selecting six random viewpoints from the 48 pre-rendered cameras for each expression. This sampling strategy encourages the model to integrate information across diverse view combinations, reducing the risk of overfitting to particular camera configurations.

We use the AdamW optimizer with differential learning rates: 1×10^{-4} for the newly added multi-view attention components and 1×10^{-5} for the DAViD backbone. A weight decay of 1×10^{-4} is used while training the model for 100 epochs with a batch size of 6 views per sample. The model is trained on an NVIDIA Quadro RTX 8000 GPU using PyTorch, with a total training time of 82 hours on 900 training expressions (45 subjects) and 100 validation expressions (5 subjects).

S2.4. Inference

Our architecture extends the monocular foundation model DAViD with a multi-view cross-attention mechanism to aggregate information across different input viewpoints in each iteration. Although the model is trained exclusively

with 6-view batches, it generalizes seamlessly to arbitrary numbers of views during inference. This generalization capability stems from the inherent properties of cross-attention. Each target view attends to all available context views through learned query-key-value projections. The key and value matrices encode information from each context view into a fixed embedding dimension, which the attention mechanism aggregates regardless of the number of input views. During training with randomly sampled 6-view subsets, the model learns to flexibly combine information from diverse viewpoint configurations, making it naturally robust to different numbers of views at test time.

At inference, the model accepts any number of input views (from 3 to 26 in our experiments) without architectural modifications or retraining. The cross-attention operation dynamically adjusts to the available views, computing attention weights that reflect the geometric consistency of each viewpoint for reconstructing the target normal map.

S3. Implementation Details For Mesh Optimization

In this section, we provide implementation details for the Mesh optimization (Sec. 3.2 in the main paper).

For optimization, we first pre-compute the per-pixel weight matrix from the normal maps (obtained from our normal prediction model) using Eq. 9 of the main paper. The target normal maps are then masked using the foreground segmentation model from DAViD to retain the complete head region, including the face, neck, and upper torso.

We use the optimization framework from Continuous Remeshing to refine the mesh geometry. The optimization proceeds for 300 steps with a learning rate of 0.3. At each iteration, we rasterize the normal maps from the current mesh state using nvdiffrast (Eq. 6), compute the normal loss (Eq. 8) along with the Laplacian regularization term (Eq. 7) with $\lambda_{\text{lap}} = 0.1$), and update the mesh vertices. At the end of each optimization step, we apply adaptive remeshing operations from Continuous Remeshing. These operations consist of edge splits, collapses, and flips that dynamically adjust mesh resolution to local geometric complexity while preventing degeneracies such as collapsed faces and self-intersections.

S3.1. Initialization

For the optimization to converge successfully, the initial sphere must be positioned close to the target mesh location in 3D space. To estimate this location, we leverage the known camera extrinsics. We cast rays from each camera center through its optical axis and compute the point in 3D space that minimizes the sum of distances to all rays. This point serves as the center of the initial sphere. While this approach is most accurate when all cameras are focused on the same region, it remains robust even when camera viewing directions vary, provided the cameras roughly face a common area of interest.

During optimization, we observed that initializing from a neutral facial template rather than a sphere does not improve convergence or final quality (Figure S3). Both converge to comparable results in 300 iterations.

S3.2. Mesh Resolution

The optimization controls granularity by specifying the maximum vertex count and minimum edge length. We observed no quality gains beyond 500k vertices, suggesting resolution ceases to be a limiting factor at that scale. Skullptor's evaluation used this vertex count, which is the same order of magnitude as 2DGS/SuGaR ($\sim$450k) and Meshroom ($\sim$250k).

S4. Evaluation Protocol on NPHM and Multiface

Because the NPHM dataset provides only raw textured meshes without corresponding camera images, we generate a synthetic multi-view dataset by rendering each mesh from 23 virtual cameras. The cameras are uniformly sampled on a sphere around the head to provide full 360 degrees of face coverage, with the head centered at the origin. We render 2048×2048 color images and corresponding camera-space normal maps using nvdiffrast, without shading (texture only). These renders form the input to all reconstruction baselines. High image resolution is necessary for fair comparison, as Meshroom, 2DGS, and SuGaR exhibit degraded performance at lower input resolutions.

As Multiface provides original lightstage-captured images at 1334×2048 resolution, we directly use the real multi-view RGB frames as input during evaluation. The dataset contains synchronized multi-view video sequences of subjects performing a variety of expressions. From the camera array, we select 26 viewpoints that provide full facial coverage and feed their captured images to all reconstruction methods.

For our method, the input images are processed to match the 512×512 training resolution of our multi-view normal predictor. For NPHM, we downsample the 2048×2048 rendered images. For Multiface, we resize and pad the 1334×2048 captured images to obtain 512×512 inputs. All baselines operate directly on the original image resolution (2048×2048 for NPHM renders and the native resolution for Multiface captures of 1334×2048).

To evaluate reconstruction quality, we measure performance on novel viewpoints not used during reconstruction. We render 12 additional frontal-facing cameras, explicitly excluding any back-of-head viewpoints, to test generalization to unseen poses. For each method, we render normal and depth maps from its reconstructed mesh at 512×512 from these novel viewpoints and compare them against

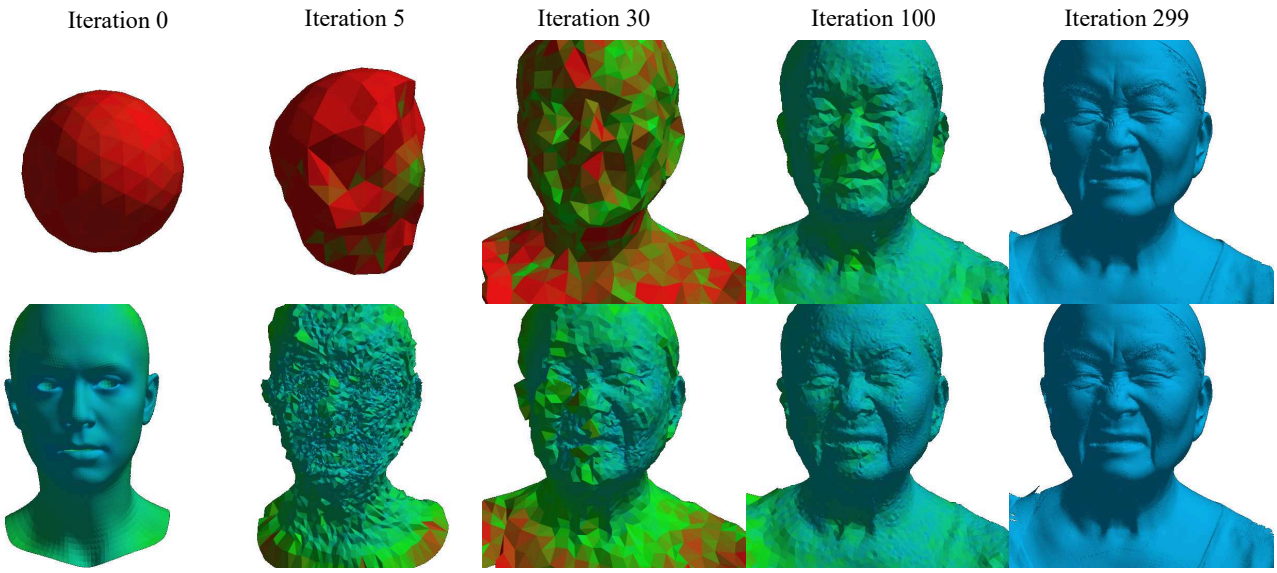

Figure S3. Effect of initialization on mesh optimization. Vertex color indicates local edge length (Red short large edges, blue for small edges). **Top:** Sphere initialization. **Bottom:** Template mesh initialization. Facial characteristics from the template mesh are discarded, and both initializations converge to comparable results in 300 iterations.

ground-truth renderings from the corresponding ground-truth mesh under identical camera poses.

To ensure consistency across all methods, every reconstructed mesh is first aligned to a canonical coordinate system using Procrustes analysis (as described in Sec. 3.2). Depth and normal comparisons are performed after this canonical alignment for all methods, which is necessary because some baselines may output meshes in arbitrary coordinate frames.

Evaluation focuses on the frontal facial region, which is the primary area of interest for head-level fidelity. We extract per-view facial masks using Facer, removing hair, neck, and clothing. These masks are applied both to predictions and ground truth to ensure that metrics reflect facial geometry alone.