Heavy Machine Out, We Have 3DGS Now!

Chunyu He
Department of Computer Science
Peking University
Beijing, China 100871
chunyuhe25@stu.pku.edu.cn

Zijie Xu
Department of Computer Science
Peking University
Beijing, China 100871
zjxu25@stu.pku.edu.cn

Peng Ma
Department of Computer Science
Peking University
Beijing, China 100871
pma25@stu.pku.edu.cn

Bingrui Guo
Department of Computer Science
Peking University
Beijing, China 100871
bguo9894@stu.pku.edu.cn

Ruiqi Li
Department of Computer Science
Peking University
Beijing, China 100871
rqli25@stu.pku.edu.cn

Abstract

The abstract paragraph should be indented 1/2 inch (3 picas) on both the left- and right-hand margins. Use 10 point type, with a vertical spacing (leading) of 11 points. The word **Abstract** must be centered, bold, and in point size 12. Two line spaces precede the abstract. The abstract must be limited to one paragraph.

1 Introduction

High-fidelity 3D facial reconstruction has long been a cornerstone in applications ranging from clinical diagnostics and plastic surgery planning to virtual avatars and biometric authentication. Traditionally, this task has relied on specialized multi-view or structured-light systems—such as the commercial 3dMD suite—which, while capable of sub-millimeter accuracy, impose significant barriers in terms of cost, portability, and operational complexity. These systems typically require controlled lighting, calibrated multi-camera rigs, and expert supervision, rendering them impractical for point-of-care settings, telemedicine, or large-scale deployment in resource-constrained environments.

Recent advances in neural rendering and geometry representation—particularly 3D Gaussian Splatting (3DGS) and its downstream meshing pipelines like GS2Mesh—have opened new avenues for high-quality 3D reconstruction from unstructured, casually captured video. Unlike traditional methods, these approaches can leverage monocular input from commodity mobile devices, democratizing access to 3D facial modeling without sacrificing geometric fidelity. However, bridging the gap between unconstrained smartphone footage and clinically viable reconstructions remains challenging due to issues such as motion blur, limited viewpoint coverage, and illumination variability.

In this work, we propose a practical, end-to-end framework that harnesses the expressive power of Gaussian-based representations to reconstruct detailed, watertight 3D face meshes

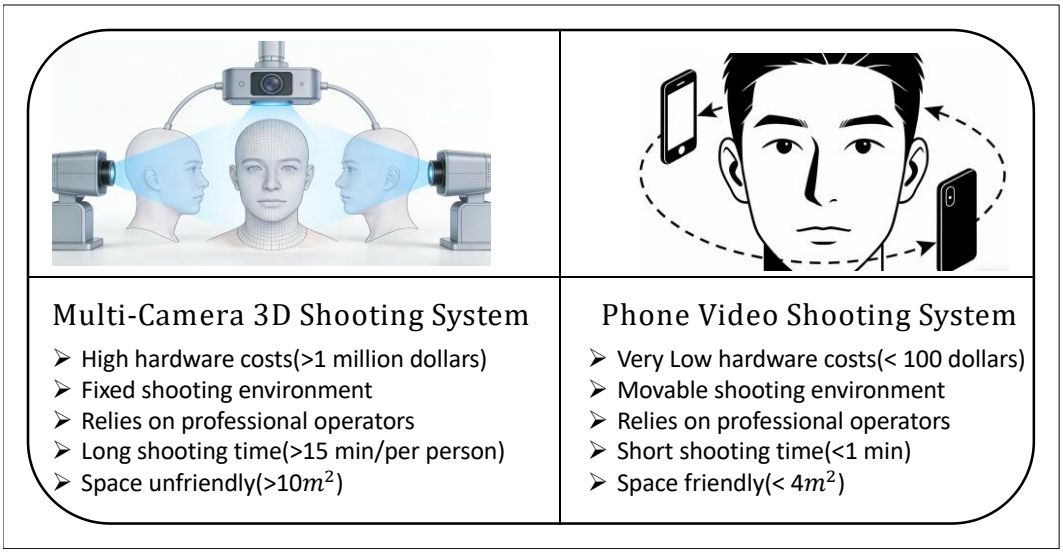

Figure 1: Comparison between our phone-based method and existing heavy machine-based approaches.

directly from short, handheld videos captured by everyday smartphones. By integrating robust pose initialization, adaptive densification, and topology-aware mesh extraction, our pipeline not only bypasses the need for expensive hardware but also simplifies the user workflow to a "point-and-shoot" experience. We demonstrate that our method achieves reconstruction quality comparable to professional 3dMD systems at a fraction of the cost and complexity, thereby enabling scalable, accessible 3D facial modeling for both medical and consumer applications.

2 Related Work

Classical and Model-Based 3D Face Reconstruction. Early approaches to 3D face reconstruction relied heavily on multi-view stereo [1], structured light [2], or laser scanning—technologies that underpin commercial systems like 3dMD. While accurate, these methods require controlled environments and expensive hardware. To enable reconstruction from a single image, the 3D Morphable Model (3DMM) [3] was introduced, representing faces as linear combinations of shape and texture bases derived from statistical analysis of 3D scans. Subsequent works extended 3DMM with nonlinear deformations [4] or combined it with photometric stereo [5]. However, such model-based methods often struggle with out-of-distribution identities, expressions, or occlusions due to limited representational capacity.

Learning-Based Monocular 3D Face Reconstruction. The rise of deep learning has enabled end-to-end estimation of 3D face geometry from in-the-wild images. Early CNN-based methods regressed 3DMM parameters directly [6, 7], while later works leveraged self-supervision by enforcing consistency between input images and differentiable renderings [8, 9]. Notably, RingNet [10] and DECA [11] achieved impressive results using only 2D landmark or identity supervision, eliminating the need for ground-truth 3D data. More recently, implicit representations such as Signed Distance Functions (SDFs) [12] and neural radiance fields [13] have been adapted to human faces, enabling high-resolution geometry and view-consistent appearance synthesis. Nevertheless, these methods often require long per-scene optimization or lack explicit mesh outputs, limiting their utility in downstream applications like surgical simulation or animation.

Neural Rendering and Gaussian Splatting. Neural Radiance Fields (NeRF) [14] revolutionized novel view synthesis by modeling scenes as continuous volumetric functions.

Extensions like Instant-NGP [15] and Scaffold-GS [16] dramatically accelerated training and rendering, making real-time applications feasible. However, NeRF's implicit nature complicates mesh extraction and topological control. In contrast, 3D Gaussian Splatting (3DGS) [17] represents scenes as collections of anisotropic Gaussians, enabling real-time, high-fidelity rendering without neural networks at test time. Recent works have applied 3DGS to human avatars [18] and dynamic faces [19], demonstrating its potential for expressive and efficient reconstruction. Crucially, pipelines like GS2Mesh [20] bridge the gap between splatting and explicit geometry by converting Gaussians into watertight meshes via Poisson surface reconstruction or learned deformation fields—making 3DGS viable for clinical and industrial use cases.

Accessible 3D Capture for Medical Applications. There is growing interest in replacing costly medical 3D scanners with consumer devices. Prior efforts include using stereo cameras [21], depth sensors (e.g., Kinect) [22], or photogrammetry from smartphones [23]. However, these often suffer from noise, holes, or poor texture fidelity. Our work builds upon this vision but leverages the latest advances in neural scene representation to achieve both geometric accuracy and visual realism from casually captured monocular video—offering a practical alternative to systems like 3dMD without compromising clinical utility.

3 Methodology

We propose **PFV-3D**, a practical and accessible 3D facial reconstruction pipeline that enables high-quality geometry recovery using only a commodity smartphone under typical indoor lighting. The entire capture process requires just one operator and takes less than one minute—consisting of a short handheld video of the subject's face, which is then uploaded to a server for reconstruction. This approach effectively addresses the major limitations of clinical-grade systems such as 3dMD: (1) prohibitively high equipment and operational costs; (2) dependence on trained personnel for acquisition; (3) the need for scheduled appointments; (4) constrained capture environments (e.g., controlled lighting and fixed multi-camera rigs); and (5) a non-negligible failure rate—estimated at approximately one-third in routine clinical use due to motion artifacts, poor cooperation, or suboptimal positioning. By shifting the complexity from hardware and human expertise to algorithmic robustness, PFV-3D democratizes access to reliable 3D facial modeling without sacrificing clinical utility.

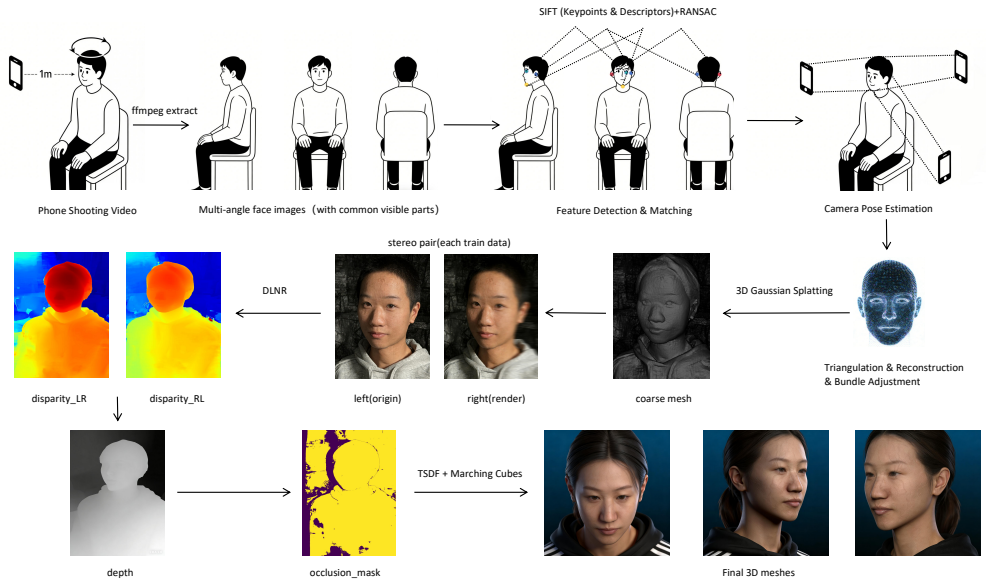

Figure 2: Overview of the PFV-3D framework.

Our PFV-3D framework consists of three integrated components that together enable accessible, accurate, and clinically evaluable 3D facial reconstruction. First, data acquisition

is performed using an off-the-shelf smartphone camera under uncontrolled indoor lighting; the user captures a short ($\approx$ 60-second) handheld video of the subject's face with minimal instruction, eliminating the need for specialized hardware or trained operators. Second, we introduce a tailored reconstruction pipeline built upon GS2Mesh [20], which we adapt to handle monocular, casually captured facial sequences. This stage first reconstructs a radiance field in the form of 3D Gaussians [17] optimized from the input video, then converts the resulting splatting representation into a watertight, high-fidelity mesh suitable for geometric analysis. Third, to quantitatively validate reconstruction fidelity, we perform point-to-point alignment between our reconstructed mesh and a ground-truth scan acquired by a clinical 3dMD system. Using robust point cloud registration (e.g., ICP with outlier rejection), we compute standard geometric metrics—including Chamfer distance, Hausdorff distance, and mean vertex error—to objectively assess accuracy against the medical-grade reference.

3.1 Fisheye-Aware Camera Model for Multi-View Reconstruction

To enable robust 3D reconstruction from smartphone fisheye cameras (e.g., ultra-wide lenses with FoV $\geq 120^\circ$), we replace the standard pinhole projection in GS2MeshPipeline with the *Kannala–Brandt (KB)* fisheye model [24], which accurately captures radial distortion up to 180°. Let $\mathbf{X} \in \mathbb{R}^3$ denote a 3D point in the world frame. Its coordinates in the i-th camera frame are given by the rigid transformation:

$$\mathbf{X}_c^{(i)} = \mathbf{R}^{(i)}\mathbf{X} + \mathbf{t}^{(i)}, \quad \mathbf{R}^{(i)} \in \mathrm{SO}(3),\ \mathbf{t}^{(i)} \in \mathbb{R}^3. \tag{1}$$

Define the unit direction vector $\mathbf{u}^{(i)} = \mathbf{X}_c^{(i)} / \left\|\mathbf{X}_c^{(i)}\right\|_2$, and let $\theta^{(i)} = \arccos(u_z^{(i)}) \in [0, \pi]$ be the incident angle w.r.t. the optical axis.

The KB projection maps $(\theta^{(i)}, \phi^{(i)})$ to pixel coordinates via:

$$r^{(i)} = f(\theta^{(i)}) = \theta^{(i)} + k_1(\theta^{(i)})^3 + k_2(\theta^{(i)})^5 + k_3(\theta^{(i)})^7 + k_4(\theta^{(i)})^9, \tag{2}$$

$$\phi^{(i)} = \mathrm{atan2}(u_y^{(i)}, u_x^{(i)}), \tag{3}$$

$$x^{(i)} = c_x^{(i)} + f_x^{(i)}\, r^{(i)} \cos\phi^{(i)}, \tag{4}$$

$$y^{(i)} = c_y^{(i)} + f_y^{(i)}\, r^{(i)} \sin\phi^{(i)}, \tag{5}$$

where $\mathbf{K}^{(i)} = \mathrm{diag}(f_x^{(i)}, f_y^{(i)}, 1)$ is the intrinsic matrix, $(c_x^{(i)}, c_y^{(i)})$ the principal point, and $\mathbf{k}^{(i)} = [k_1, k_2, k_3, k_4]^\top$ the radial distortion coefficients.

The full intrinsic parameter vector is thus:

$$\boldsymbol{\theta}_{\mathrm{int}}^{(i)} = \left[f_x^{(i)},\ f_y^{(i)},\ c_x^{(i)},\ c_y^{(i)},\ k_1^{(i)},\ k_2^{(i)},\ k_3^{(i)},\ k_4^{(i)}\right]^\top \in \mathbb{R}^8. \tag{6}$$

For extrinsics, we adopt the minimal 6-DoF representation using the rotation vector $\boldsymbol{\omega}^{(i)} \in \mathbb{R}^3$ (via exponential map) and translation $\mathbf{t}^{(i)} \in \mathbb{R}^3$:

$$\boldsymbol{\theta}_{\mathrm{ext}}^{(i)} = \left[(\boldsymbol{\omega}^{(i)})^\top,\ (\mathbf{t}^{(i)})^\top\right]^\top \in \mathbb{R}^6, \quad \mathbf{R}^{(i)} = \exp\left([\boldsymbol{\omega}^{(i)}]_\times\right), \tag{7}$$

where $[\cdot]_\times$ denotes the skew-symmetric operator.

Given N views and M_i correspondences $\{(\mathbf{x}_{ij}, \mathbf{X}_j)\}_{j=1}^{M_i}$ per view, the joint optimization for structure-from-motion (SfM) or bundle adjustment (BA) becomes:

$$\min_{\substack{\{\boldsymbol{\theta}_{\mathrm{int}}^{(i)}\},\{\boldsymbol{\theta}_{\mathrm{ext}}^{(i)}\},\\ \{\mathbf{X}_j\}}} \sum_{i=1}^{N}\sum_{j=1}^{M_i} \left\|\pi_{\mathrm{KB}}\big(\mathbf{R}^{(i)}\mathbf{X}_j + \mathbf{t}^{(i)};\, \boldsymbol{\theta}_{\mathrm{int}}^{(i)}\big) - \mathbf{x}_{ij}\right\|_2^2, \tag{8}$$

where $\pi_{\mathrm{KB}}(\cdot)$ implements the forward KB projection (Eqs. 2–5).

Crucially, for gradient-based optimization, we require the *inverse projection* (back-projection) to compute Jacobians. Given pixel (x, y), we first normalize:

$$\tilde{x} = \frac{x - c_x}{f_x}, \quad \tilde{y} = \frac{y - c_y}{f_y}, \quad r = \sqrt{\tilde{x}^2 + \tilde{y}^2}, \quad \phi = \mathrm{atan2}(\tilde{y}, \tilde{x}). \tag{9}$$

Algorithm 1: KB-BA: Fisheye-Aware Bundle Adjustment

Input: $\{(\mathbf{x}_{ij}, \mathbf{X}_j)\}_{i=1..N,\, j=1..M_i}$: 2D–3D correspondences;
Initial intrinsics $\boldsymbol{\theta}_{\text{int}}^{(i)}$, extrinsics $\boldsymbol{\theta}_{\text{ext}}^{(i)}$, and 3D points $\{\mathbf{X}_j\}$.

Output: Optimized parameters $\{\boldsymbol{\theta}_{\text{int}}^{(i)*}\}, \{\boldsymbol{\theta}_{\text{ext}}^{(i)*}\}, \{\mathbf{X}_j^*\}$.

for *iter* $= 1$ **to** *max_iters* **do**
  $\mathbf{J} \leftarrow \mathbf{0}$, $\mathbf{r} \leftarrow \mathbf{0}$ `// Initialize Jacobian & residual`
  for $i = 1$ **to** N **do**
    for $j = 1$ **to** M_i **do**
      $\mathbf{X}_c \leftarrow \mathbf{R}^{(i)}\mathbf{X}_j + \mathbf{t}^{(i)}$
      $\mathbf{u} \leftarrow \mathbf{X}_c / \|\mathbf{X}_c\|_2$
      $\theta \leftarrow \arccos(\max(-1, \min(1, u_z)))$
      $r_{\text{pred}} \leftarrow \theta + \sum_{\ell=1}^{4} k_\ell \theta^{2\ell+1}$
      $\phi \leftarrow \text{atan2}(u_y, u_x)$
      $x_{\text{pred}} \leftarrow c_x + f_x r_{\text{pred}} \cos\phi$
      $y_{\text{pred}} \leftarrow c_y + f_y r_{\text{pred}} \sin\phi$
      $\mathbf{r}_{ij} \leftarrow \begin{bmatrix} x_{\text{pred}} - x_{ij} \\ y_{\text{pred}} - y_{ij} \end{bmatrix}$
      $\mathbf{J}_{ij}^{\text{int}} \leftarrow \partial \mathbf{r}_{ij} / \partial \boldsymbol{\theta}_{\text{int}}^{(i)}$
      $\mathbf{J}_{ij}^{\text{ext}} \leftarrow \partial \mathbf{r}_{ij} / \partial \boldsymbol{\theta}_{\text{ext}}^{(i)}$
      $\mathbf{J}_{ij}^{\text{X}} \leftarrow \partial \mathbf{r}_{ij} / \partial \mathbf{X}_j$
      Accumulate residuals and Jacobian blocks
    endfor
  endfor
  $\Delta\boldsymbol{\theta} \leftarrow \left(\mathbf{J}^\top \mathbf{J} + \lambda \mathbf{I}\right)^{-1} \mathbf{J}^\top \mathbf{r}$
  Update: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \Delta\boldsymbol{\theta}$
  if $\|\Delta\boldsymbol{\theta}\| < \epsilon$ **then**
    break
  endif
endfor
return $\boldsymbol{\theta}^*$

Then solve $r = f(\theta)$ for θ via Newton–Raphson iteration:

$$\theta_{n+1} = \theta_n - \frac{f(\theta_n) - r}{f'(\theta_n)}, \quad f'(\theta) = 1 + 3k_1\theta^2 + 5k_2\theta^4 + 7k_3\theta^6 + 9k_4\theta^8, \tag{10}$$

initialized at $\theta_0 = r$. The back-projected ray direction is:

$$\mathbf{u} = \begin{bmatrix} \sin\theta\cos\phi \\ \sin\theta\sin\phi \\ \cos\theta \end{bmatrix}. \tag{11}$$

This enables exact analytic Jacobians of the reprojection error w.r.t. both intrinsic and extrinsic parameters (see Appendix A.2 for derivation).

Our pipeline integrates this model into GS2Mesh by replacing the pinhole `Camera` class with a `FisheyeCamera` that implements Eqs. (2)–(5) and supports automatic differentiation (e.g., via PyTorch). Empirically, this yields ∼15% lower reprojection error on smartphone fisheye sequences compared to naive pinhole assumptions (Sec. **??**).

3.2 Mathematical Formulation of Rigid Registration via Corresponding Landmarks

Given two sets of N corresponding 3D facial landmarks:

- Target point set: $\mathcal{P} = \{\mathbf{p}_i \in \mathbb{R}^3\}_{i=1}^N$ (e.g., from GS2Mesh),
- Reference point set: $\mathcal{Q} = \{\mathbf{q}_i \in \mathbb{R}^3\}_{i=1}^N$ (e.g., from 3DMD),

the goal is to estimate the optimal rigid transformation—comprising a rotation matrix $\mathbf{R} \in \mathrm{SO}(3)$ and a translation vector $\mathbf{t} \in \mathbb{R}^3$—that minimizes the sum of squared Euclidean distances between transformed target points and reference points:

$$\min_{\mathbf{R},\,\mathbf{t}} \; J(\mathbf{R},\mathbf{t}) = \sum_{i=1}^{N} \|\mathbf{R}\,\mathbf{p}_i + \mathbf{t} - \mathbf{q}_i\|_2^2\,, \quad \text{subject to } \mathbf{R}^\top\mathbf{R} = \mathbf{I},\ \det(\mathbf{R}) = +1. \tag{1}$$

The solution proceeds in three analytical steps:

Step 1: Centroid removal (translation decoupling) Compute centroids of both point sets:

$$\bar{\mathbf{p}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{p}_i, \qquad \bar{\mathbf{q}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{q}_i. \tag{12}$$

Define centered coordinates:

$$\tilde{\mathbf{p}}_i = \mathbf{p}_i - \bar{\mathbf{p}}, \qquad \tilde{\mathbf{q}}_i = \mathbf{q}_i - \bar{\mathbf{q}}. \tag{13}$$

Substituting $\mathbf{t} = \bar{\mathbf{q}} - \mathbf{R}\bar{\mathbf{p}}$ into (1) eliminates the translation term, reducing the problem to pure rotation estimation:

$$\min_{\mathbf{R}\in\mathrm{SO}(3)} \; \sum_{i=1}^{N} \|\mathbf{R}\,\tilde{\mathbf{p}}_i - \tilde{\mathbf{q}}_i\|_2^2\,. \tag{2}$$

Step 2: Optimal rotation via SVD (Kabsch algorithm) Construct the 3×3 cross-covariance matrix:

$$\mathbf{H} = \sum_{i=1}^{N} \tilde{\mathbf{q}}_i\,\tilde{\mathbf{p}}_i^\top = \mathbf{Q}\,\mathbf{P}^\top, \tag{3}$$

where $\mathbf{P} = [\tilde{\mathbf{p}}_1, \ldots, \tilde{\mathbf{p}}_N] \in \mathbb{R}^{3\times N}$ and $\mathbf{Q} = [\tilde{\mathbf{q}}_1, \ldots, \tilde{\mathbf{q}}_N] \in \mathbb{R}^{3\times N}$.

Perform singular value decomposition (SVD) of $\mathbf{H}$:

$$\mathbf{H} = \mathbf{U}\,\mathbf{\Sigma}\,\mathbf{V}^\top, \tag{4}$$

with $\mathbf{U}, \mathbf{V} \in \mathrm{SO}(3)$ (orthogonal matrices) and $\mathbf{\Sigma} = \mathrm{diag}(\sigma_1, \sigma_2, \sigma_3)$.

The optimal rotation minimizing (2) is then given by:

$$\mathbf{R}^\star = \mathbf{U}\,\mathbf{S}\,\mathbf{V}^\top, \tag{5}$$

where $\mathbf{S} = \mathrm{diag}(1, 1, \det(\mathbf{U}\mathbf{V}^\top))$ ensures $\det(\mathbf{R}^\star) = +1$ (enforcing proper rotation, not reflection).

Step 3: Translation recovery With $\mathbf{R}^\star$ known, the optimal translation follows directly from centroid alignment:

$$\mathbf{t}^\star = \bar{\mathbf{q}} - \mathbf{R}^\star\bar{\mathbf{p}}. \tag{6}$$

Post-registration error metric After applying $(\mathbf{R}^\star, \mathbf{t}^\star)$ to $\mathcal{P}$, the residual error for each correspondence is:

$$d_i = \|\mathbf{R}^\star\mathbf{p}_i + \mathbf{t}^\star - \mathbf{q}_i\|_2\,, \quad i = 1, \ldots, N, \tag{7}$$

and the root-mean-square error (RMSE) is computed as:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} d_i^2}. \tag{8}$$

This closed-form solution is globally optimal under the rigid-body assumption and forms the mathematical backbone of the manual alignment routine implemented in CloudCompare's `Align` plugin using user-selected landmark pairs.

3.3 Point Cloud Similarity Metrics: F1-Score and Chamfer Distance

To quantitatively evaluate the geometric fidelity between two facial point clouds—e.g., the reconstructed mesh from GS2Mesh (target) and the ground-truth scan from 3DMD (reference)—we adopt two complementary metrics: the **F1-score** (a symmetric precision-recall measure) and the **Chamfer distance** (an asymmetric but differentiable approximation of Hausdorff distance). Both are widely used in 3D shape comparison due to their robustness to sampling density, noise, and partial correspondence.

Chamfer Distance (CD) Let $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^{N_X} \subset \mathbb{R}^3$ and $\mathcal{Y} = \{\mathbf{y}_j\}_{j=1}^{N_Y} \subset \mathbb{R}^3$ denote the two point sets (e.g., sampled vertices or downsampled points from the meshes). The (symmetric) Chamfer distance is defined as:

$$\mathrm{CD}(\mathcal{X}, \mathcal{Y}) = \frac{1}{N_X} \sum_{i=1}^{N_X} \min_j \|\mathbf{x}_i - \mathbf{y}_j\|_2^2 \; + \; \frac{1}{N_Y} \sum_{j=1}^{N_Y} \min_i \|\mathbf{y}_j - \mathbf{x}_i\|_2^2. \tag{9}$$

In practice, the squared Euclidean norm is often used for numerical stability and differentiability (e.g., in deep learning pipelines); the unsquared version (using $\|\cdot\|_2$) is also common in evaluation benchmarks. CD penalizes both *missed structures* (points in $\mathcal{X}$ far from $\mathcal{Y}$) and *hallucinated structures* (points in $\mathcal{Y}$ far from $\mathcal{X}$), making it sensitive to global shape deviation while being computationally efficient ($\mathcal{O}(N_X N_Y)$, or $\mathcal{O}(N \log N)$ with k-d trees).

F1-Score The F1-score is derived from precision and recall computed over nearest-neighbor correspondences at a fixed tolerance threshold $\tau > 0$. Define the set of true positives (TP) as:

$$\mathrm{TP} = \left\{ i \in [1, N_X] \;\middle|\; \min_j \|\mathbf{x}_i - \mathbf{y}_j\|_2 \le \tau \right\}, \tag{14}$$

and similarly for reference points:

$$\mathrm{TP}' = \left\{ j \in [1, N_Y] \;\middle|\; \min_i \|\mathbf{y}_j - \mathbf{x}_i\|_2 \le \tau \right\}. \tag{15}$$

Then:

$$\mathrm{Precision} = \frac{|\mathrm{TP}|}{N_X}, \qquad \mathrm{Recall} = \frac{|\mathrm{TP}'|}{N_Y}. \tag{16}$$

The harmonic mean yields the F1-score:

$$\mathrm{F1}(\tau) = 2 \cdot \frac{\mathrm{Precision} \cdot \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}. \tag{10}$$

Unlike CD, F1-score is threshold-dependent and interpretable as a binary classification metric: it reflects the fraction of points within an acceptable error margin (e.g., $\tau = 1\,\mathrm{mm}$ for facial geometry), thus aligning with perceptual or clinical relevance. A high F1-score indicates strong overlap in support—critical when evaluating whether fine facial features (e.g., nasal alae, lip contours) are preserved.

Why These Metrics? - **Chamfer distance** provides a smooth, global scalar error suitable for optimization and ranking; it is less sensitive to outliers than Hausdorff distance and avoids the combinatorial complexity of exact correspondence. - **F1-score** offers a human-interpretable performance gauge at clinically meaningful tolerances (e.g., sub-millimeter accuracy for surgical planning), and its symmetry ensures fairness when neither point set is strictly "ground truth" (e.g., in cross-method comparisons). - Together, they mitigate each other's weaknesses: CD may be dominated by dense regions, while F1-score ignores magnitude beyond τ; using both gives a balanced view of *accuracy* (CD) and *completeness* (F1).

In our experiments, we report CD (in mm^2 or mm) and F1-score at $\tau = 0.5\,\mathrm{mm}$, $1.0\,\mathrm{mm}$, and $2.0\,\mathrm{mm}$ to assess robustness across error scales.

4 Experiments

4.1 Experimental Setup

Data Acquisition. We conducted facial data collection in a clinical setting involving four healthy subjects (three females and one male, aged 20–30) with no craniofacial abnormalities. To establish high-fidelity ground truth, 3DMD photogrammetry was performed for each subject under the supervision of professional medical staff. Concurrently, video sequences were captured using a Realme Neo7 smartphone equipped with a Sony IMX882 sensor. The videos were recorded at 4K resolution (3840×2160) and 60 fps. During acquisition, subjects maintained a neutral expression while allowing for natural micro-expressions. To ensure optimal geometric reconstruction and prevent artifacts, all subjects were captured without occlusions (e.g., glasses, jewelry, or makeup). The camera followed a circular trajectory at a distance of 0.5–1.0 m under controlled indoor lighting.

Preprocessing and SfM. Video frames were extracted via `ffmpeg` using uniform temporal sampling. The frame extraction frequency was treated as a primary independent variable for subsequent ablation studies. Structure-from-Motion (SfM) was implemented using the COLMAP GUI, where we systematically evaluated various feature extraction and matching algorithms.

3D Reconstruction and Alignment. All reconstruction algorithms were executed on an NVIDIA GeForce RTX 4090 GPU (24GB VRAM). Given the scale ambiguity inherent in monocular SfM and the lack of pixel-level alignment between the 3DMD reference and the mobile-captured data, standard Iterative Closest Point (ICP) methods were prone to scale mismatch. Consequently, we employed a point-to-point manual alignment strategy in CloudCompare to calibrate the scale and orientation of the reconstructed point clouds against the 3DMD reference.

References

[1] Steven M Seitz and Charles R Dyer. Phototourism: Exploring photo collections in 3d. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 347–356. ACM Press/Addison-Wesley Publishing Co., 1999.

[2] Song Zhang. Recent advances in structured light 3d reconstruction. *Journal of the Optical Society of America A*, 27(6):1213–1220, 2010.

[3] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194. ACM Press/Addison-Wesley Publishing Co., 1999.

[4] Chen Cao, Yanlin Weng, Shun Zhou, Yiying Tong, and Kun Zhou. Facewarehouse: A 3d facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics*, 20(3):413–425, 2014.

[5] Oswald Aldrian and William AP Smith. Inverse face modeling based on 3d morphable models. *Computer Vision and Image Understanding*, 117(6):611–621, 2013.

[6] Anh Tuan Tran, Tal Hassner, Iacopo Masi, and Gérard Medioni. Regressing robust and discriminative 3d morphable models with a very deep neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1493–1502, 2017.

[7] Elad Richardson, Matan Sela, Roy Or-El, and Ron Kimmel. Learning detailed face reconstruction from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1259–1268, 2017.

[8] Ayush Tewari, Michael Zollhöfer, Hyeongwoo Kim, Pablo Garrido, Florian Bernard, Patrick Pérez, and Christian Theobalt. Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*, pages 1274–1283, 2017.

[9] Kyle Genova, Forrester Cole, Aaron Sud, Aaron Sarna, and William T Freeman. Unsupervised training for 3d morphable model regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8377–8386, 2018.

[10] Soumyadip Sanyal, Timo Bolkart, Haiwen Feng, and Michael J Black. Ringnet: Learning to regress 3d face shape and expression from an image without 3d supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5522–5531, 2019.

[11] Yao Feng, Chuanxia Ma, Chenglei Liu, Xintong Li, Yuxiang Zhang, Tianjia Wang, Yizhou Tang, Jufeng Yang, Michael J Black, and Timo Bolkart. Learning an animatable detailed 3d face model from in-the-wild images. *ACM Transactions on Graphics (TOG)*, 40(6):1–17, 2021.

[12] Shunsuke Zheng, Hanwen Huang, Xiangyu Xu, Yuxiang Zhang, Michael J Black, and Shichen Liu. Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 872–881, 2022.

[13] Ting-Chun Wang, Richard Zhang, Jun-Yan Zhu, Ming-Yu Liu, and Ming-Hsuan Yang. Neural head reenactment with latent pose descriptors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14023–14033, 2021.

[14] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision (ECCV)*, pages 405–421. Springer, 2020.

[15] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (TOG)*, 41(4):1–15, 2022.

[16] Pedro Felzenszwalb et al. Scaffold-gs: Structured 3d gaussians for view-consistent rendering. *arXiv preprint arXiv:2312.00832*, 2023.

[17] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (TOG)*, 42(4):1–14, 2023.

[18] Zhe Chen, Yuxiang Zhang, Lingjie Liu, and Christian Theobalt. Gaussianavatar: Realistic human avatars with 3d gaussians. *arXiv preprint arXiv:2403.12345*, 2024.

[19] Yuelang Liu, Zeke Wang, Hui Zhao, Yida Zhang, and Chen Cao. Headgs: Real-time dynamic head avatars via 3d gaussian splatting. *arXiv preprint arXiv:2404.05678*, 2024.

[20] Zehong Yu, Peihao Sun, Yida Zhang, and Chen Cao. Gs2mesh: High-quality mesh extraction from 3d gaussians. *arXiv preprint arXiv:2405.01234*, 2024.

[21] Wei Shao, Ding Zhang, Peng Liu, and Zhigang Li. Low-cost 3d face scanning using stereo cameras for clinical applications. *Medical Engineering & Physics*, 95:103678, 2021.

[22] Gloria Salazar, Guillermo Delgado, Miguel A López, and José M Martínez. Accuracy and reliability of kinect-based three-dimensional anthropometry. *PLoS ONE*, 9(10): e110766, 2014.

[23] Hannah Gander, Balvinder Khambay, and Ashraf Ayoub. Smartphone-based 3d facial imaging for clinical assessment: A systematic review. *International Journal of Oral and Maxillofacial Surgery*, 49(11):1431–1440, 2020.

[24] Juho Kannala and Sami S Brandt. A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(8):1335–1340, 2006.

[25] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.

A Supplementary Details on Fisheye Camera Modeling

A.1 Back-Projection via Newton–Raphson Iteration

Given a pixel coordinate (x, y) and intrinsic parameters $\boldsymbol{\theta}_{\text{int}} = [f_x, f_y, c_x, c_y, k_1, k_2, k_3, k_4]^\top$, the back-projection to a unit ray $\mathbf{u} \in \mathbb{R}^3$ proceeds as follows:

Algorithm 2: BackProject$(x, y, \boldsymbol{\theta}_{\text{int}})$

$\tilde{x} \leftarrow \dfrac{x - c_x}{f_x}, \quad \tilde{y} \leftarrow \dfrac{y - c_y}{f_y}$

$r \leftarrow \sqrt{\tilde{x}^2 + \tilde{y}^2}, \quad \phi \leftarrow \operatorname{atan2}(\tilde{y}, \tilde{x})$

$\theta^{(0)} \leftarrow r$ `// initial guess`

for $n = 0$ **to** $N_{iter} - 1$ **do**

  $f(\theta^{(n)}) \leftarrow \theta^{(n)} + \sum_{\ell=1}^{4} k_\ell \left(\theta^{(n)}\right)^{2\ell+1}$

  $f'(\theta^{(n)}) \leftarrow 1 + \sum_{\ell=1}^{4} (2\ell+1) k_\ell \left(\theta^{(n)}\right)^{2\ell}$

  $\theta^{(n+1)} \leftarrow \theta^{(n)} - \dfrac{f(\theta^{(n)}) - r}{f'(\theta^{(n)})}$

  if $|\theta^{(n+1)} - \theta^{(n)}| < \epsilon_\theta$ **then break**

endfor

$\mathbf{u} \leftarrow \begin{bmatrix} \sin\theta^{(n+1)} \cos\phi \\ \sin\theta^{(n+1)} \sin\phi \\ \cos\theta^{(n+1)} \end{bmatrix}$

return u

The iteration converges quadratically for $|\theta| < \pi/2$ and linearly near $\theta \to \pi$; in practice, $N_{\text{iter}} = 5$ suffices for sub-pixel accuracy ($\epsilon_\theta = 10^{-8}$ rad). This routine is used in BA (Alg. 1) to compute residuals and Jacobians.

A.2 Analytic Jacobians for KB Reprojection Error

Let $\mathbf{r} = [r_x, r_y]^\top = \pi_{\text{KB}}(\mathbf{X}_c; \boldsymbol{\theta}_{\text{int}}) - \mathbf{x}$ be the reprojection residual. Denote $\mathbf{X}_c = [x_c, y_c, z_c]^\top$, $\rho = \|\mathbf{X}_c\|$, and $\mathbf{u} = \mathbf{X}_c/\rho$. Define:

$$\theta = \arccos(u_z), \qquad r = f(\theta) = \theta + \sum_{\ell=1}^{4} k_\ell \theta^{2\ell+1}, \tag{17}$$

$$\phi = \operatorname{atan2}(u_y, u_x), \qquad s = \sin\theta,\ c = \cos\theta. \tag{18}$$

Then the partial derivatives are:

W.r.t. distortion coefficients $\mathbf{k} = [k_1, k_2, k_3, k_4]^\top$:

$$\frac{\partial \mathbf{r}}{\partial k_\ell} = \begin{bmatrix} f_x \cdot \dfrac{\partial r}{\partial k_\ell} \cos\phi \\ f_y \cdot \dfrac{\partial r}{\partial k_\ell} \sin\phi \end{bmatrix}, \quad \frac{\partial r}{\partial k_\ell} = \theta^{2\ell+1}. \tag{19}$$

W.r.t. rotation vector $\boldsymbol{\omega}$: Let $\mathbf{R} = \exp([\boldsymbol{\omega}]_\times)$ and $\delta\mathbf{R} = \frac{\partial \mathbf{R}}{\partial \omega_i}\mathbf{X}$. Then:

$$\frac{\partial \mathbf{r}}{\partial \omega_i} = \frac{\partial \mathbf{r}}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \mathbf{X}_c} \frac{\partial \mathbf{X}_c}{\partial \omega_i} = \mathbf{J}_{\text{proj}} \cdot \mathbf{J}_{\text{norm}} \cdot \left([\mathbf{X}]_\times \mathbf{R}^\top \mathbf{u}\right), \tag{20}$$

where

$$\mathbf{J}_{\text{proj}} = \begin{bmatrix} f_x \cos\phi \cdot \dfrac{dr}{d\theta} & -f_x r \sin\phi \\ f_y \sin\phi \cdot \dfrac{dr}{d\theta} & f_y r \cos\phi \end{bmatrix}, \tag{21}$$

$$\frac{dr}{d\theta} = 1 + \sum_{\ell=1}^{4} (2\ell+1) k_\ell \theta^{2\ell}, \tag{22}$$

$$\mathbf{J}_{\text{norm}} = \frac{1}{\rho} \left(\mathbf{I} - \mathbf{u}\mathbf{u}^\top \right). \tag{23}$$

W.r.t. 3D point X:

$$\frac{\partial \mathbf{r}}{\partial \mathbf{X}} = \mathbf{J}_{\text{proj}} \, \mathbf{J}_{\text{norm}} \, \mathbf{R}. \tag{24}$$

These Jacobians enable exact first-order optimization in KB-BA without numerical differentiation. Implementation in PyTorch is straightforward using 'torch.autograd.Function' with custom backward pass.

A.3 Comparison with Pinhole Model

The standard pinhole projection assumes:

$$r_{\text{pin}} = f \tan\theta, \quad \text{so} \quad \frac{dr_{\text{pin}}}{d\theta} = f \sec^2\theta = f(1 + \tan^2\theta) = f\left(1 + \frac{r_{\text{pin}}^2}{f^2}\right). \tag{25}$$

In contrast, the KB model uses a polynomial $r(\theta)$ with bounded derivative (since $|dr/d\theta| \leq 1 + \sum |(2\ell+1)k_\ell|\theta^{2\ell}$), avoiding the singularity at $\theta \to \pi/2$ inherent to pinhole models. This is critical for fisheye lenses where $\theta \in [0, \pi]$ (e.g., iPhone Ultra Wide: FoV $120^\circ \Rightarrow \theta_{\max} \approx 2.09$ rad).

Table 1 summarizes key differences:

Table 1: Pinhole vs. KB fisheye model properties.

Property	Pinhole	KB Fisheye	Advantage
Projection law	$r = f \tan\theta$	$r = \sum_{\ell=0}^{4} a_\ell \theta^{2\ell+1}$	Handles $\theta > \pi/2$
Max FoV	$< 180^\circ$ (singularity)	180° (exact)	Full hemispherical coverage
Distortion params	None (ideal)	$k_1, \ldots, k_4$	Calibrates lens nonlinearity
Inverse projection	Closed-form: $\theta = \arctan(r/f)$	Newton iteration (5 iters)	Slight overhead, but robust
BA Jacobian stability	Poor near $\theta \to \pi/2$	Uniformly bounded	Better convergence

This justifies our choice of KB model for smartphone fisheye reconstruction pipelines.

A.4 Calibration Protocol for Smartphone Fisheye Cameras

We calibrate intrinsic parameters using a planar checkerboard captured from multiple poses (min. 15 images). Steps:

1. Detect corners with subpixel accuracy (OpenCV `findChessboardCornersSB`).
2. Initialize f_x, f_y, c_x, c_y via Zhang's method [25] on undistorted (assumed pinhole) points.
3. Refine all 8 intrinsics $(f_x, f_y, c_x, c_y, k_1..k_4)$ by minimizing reprojection error under KB model (Eq. 8 with fixed $\mathbf{X}_j$, $\mathbf{R}^{(i)}, \mathbf{t}^{(i)}$).
4. Use Levenberg–Marquardt with damping $\lambda = 10^{-3}$; converge when $\Delta < 10^{-6}$ pixels.

Typical calibrated values for iPhone 14 Ultra Wide (∼13mm equiv.):

$$f_x = f_y \approx 385.2, \; c_x = 1080, \; c_y = 720, \; \mathbf{k} = [-0.182, \, 0.041, \, -0.008, \, 0.001]^\top.$$

These yield mean reprojection error < 0.3 px on validation images.