Depth-Based Parametric Face Reconstruction from RGB-D Data

3D Scanning & Motion Capture – Course Project Proposal

Atakan Sucu	Emre Kalkan	Arda Senyürek	Daye Fubara
atakan.sucu@tum.de	emre.kalkan@tum.de	arda.senyurek@tum.de	daye.fubara@tum.de
03817597	03814232	03804231	03811820

Abstract

We propose a depth-focused 3D face reconstruction system based on a parametric morphable model. Identity will be estimated from a reference frame, and expression and pose will be tracked across a sequence by minimizing sparse landmark alignment and dense depth residuals. The design keeps depth as the primary supervision signal to ensure robustness and feasibility. Optional RGB refinement or expression transfer may be explored once the core pipeline is complete. The final system will provide temporally stable reconstructions with quantitative and qualitative evaluation.

1. Introduction

Parametric morphable face models, originating from the seminal work of Blanz and Vetter [1], provide a compact and expressive representation for identity and facial shape variation. This project aims to build a 3D face reconstruction pipeline that uses depth as the primary signal, enabling stable alignment even in the presence of illumination changes or texture variation.

Expression tracking and parametrization are also relevant in performance capture and facial reenactment tasks, such as Face2Face [2], which motivate the optional exploration of cross-identity expression transfer.

Dense geometric alignment from RGB-D observations is a well-studied direction in non-rigid reconstruction, with real-time methods such as Zollhöfer et al. [3] demonstrating the effectiveness of depth residuals for capturing dynamic deformations. We build on similar principles but use a low-dimensional morphable face model to maintain feasibility and interpretability.

2. Technical Approach

2.1. Parametric Face Model

We use:

$$M_{\text{geo}}(\alpha, \delta) = \bar{M} + U_\alpha \alpha + U_\delta \delta,$$

where α encodes identity and δ encodes expression. Identity is estimated once; expression and pose are updated per frame.

2.2. Camera and Depth Rendering

For parameters (α, δ, R, t), we:
1. transform vertices to camera coordinates,
2. apply perspective projection,
3. rasterize a synthetic depth map D_{rend}.

2.3. Energy Formulation

The optimization objective:

$$E(P) = E_{\text{sparse}} + E_{\text{depth}} + E_{\text{reg}}.$$

Sparse 2D landmark term:

$$E_{\text{sparse}} = \sum_i \|\pi(Rv_i + t) - \ell_i\|^2.$$

Dense depth alignment:

$$E_{\text{depth}} = \sum_{p \in \Omega} \|D_{\text{obs}}(p) - D_{\text{rend}}(p)\|^2.$$

Regularization:

$$E_{\text{reg}} = \lambda_\alpha \|\alpha\|^2 + \lambda_\delta \|\delta\|^2.$$

2.4. Optimization

Pose is initialized using PnP or Procrustes. Gauss–Newton or Levenberg–Marquardt updates optimize expression and pose per frame. Optional temporal smoothing may be applied.

2.5. Outputs

- Identity mesh,
- Per-frame expression and pose,
- Depth renderings and overlays,
- Quantitative error plots.

3. Requirements

Datasets. RGB-D sequences recorded via Azure Kinect or RealSense.

Libraries. C++, Eigen, OpenCV, minimal depth renderer, Python visualization.

Hardware. Standard laptop CPU/GPU.

4. Evaluation

We will evaluate the system using a combination of quantitative metrics and qualitative visual comparisons. The goal is to assess geometric accuracy, tracking stability, and optimization behaviour.

4.1. Quantitative Metrics

Depth reconstruction error. We compute the per-pixel depth error between observed and rendered depth:

$$\text{RMSE}_{\text{depth}} = \sqrt{\frac{1}{|\Omega|} \sum_{p \in \Omega} \left(D_{\text{obs}}(p) - D_{\text{rend}}(p)\right)^2}.$$

This measures how closely the fitted mesh aligns with the sensor geometry.

Landmark reprojection error. To validate sparse alignment and pose estimation, we report the average 2D landmark error:

$$\text{Err}_{\text{lm}} = \frac{1}{N} \sum_{i=1}^{N} \|\pi(Rv_i + t) - \ell_i\|.$$

Energy convergence. We track the evolution of

$$E = E_{\text{sparse}} + E_{\text{depth}} + E_{\text{reg}},$$

to analyze convergence across Gauss–Newton/LM iterations and verify numerical stability.

Runtime. We measure per-frame optimization time and total runtime to assess the scalability of the implementation.

4.2. Qualitative Evaluation

Qualitative assessment includes:

- side-by-side comparisons of D_{obs} and D_{rend},
- visual inspection of mesh alignment from multiple viewpoints,
- evaluation of temporal smoothness across frame sequences (reduced jitter),
- example outputs of reconstructed identity and tracked expressions.

If expression transfer is implemented, we additionally inspect:

- visual consistency of transferred expressions,
- preservation of target identity geometry.

This evaluation protocol provides a balanced assessment of reconstruction accuracy, system robustness, and temporal behaviour.

5. Milestones

Week 1. Review core literature on morphable models and depth-based fitting, study previous course projects, and finalize the system design. Inspect RGB-D sensors and implement basic data-loading utilities.

Week 2. Integrate the PCA face model, validate coefficient evaluation, and implement landmark detection. Perform initial sparse alignment and establish a reliable pose initialization pipeline.

Week 3. Develop the minimal depth renderer, including projection, rasterization, and visibility handling. Begin computing dense depth residuals and validate consistency between observed and rendered depth.

Week 4. Assemble the full optimization loop (Gauss–Newton / LM) and test single-frame reconstruction. Tune regularization weights and evaluate stability of depth-based fitting.

Week 5. Extend the system to sequence processing by tracking expression and pose over time. Introduce optional temporal smoothing and generate qualitative reconstructions.

Week 6. Perform quantitative evaluation (landmarks, depth error, convergence) and prepare visual comparisons. Finalize the written report and presentation materials.

6. Team Members

- Atakan Sucu
- Emre Kalkan
- Daye Fubara
- Arda Senyürek

References

[1] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. *SIGGRAPH*, 1999. 1

[2] Justus Thies, Michael Zollhöfer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *CVPR*, 2016. 1

[3] Michael Zollhöfer, Patrick Stotko, Christian Theobalt, et al. Real-time non-rigid reconstruction using an rgb-d camera. In *SIGGRAPH*, 2014. 1