# 3D Face Morphing Using MediaPipe and Trimesh: Technical Report

## Introduction

This project explores the morphing of a generic 3D face mesh to match a set of individual facial images using MediaPipe for 2D landmark detection and Trimesh for mesh processing. The objective is to build a pipeline that can take a reference 3D model and deform it based on facial landmarks extracted from an image, creating personalized 3D face meshes. This is particularly useful in applications like virtual avatars, animation, or biometric analysis. Additionally, the project quantifies the accuracy of morphing and provides visualization tools to evaluate deformation quality.

## Implementation

The face morphing pipeline consists of four major stages: **landmark extraction**, **landmark-to-mesh correspondence**, **mesh deformation with smoothing**, and **error visualization**. Each stage is handled by one or more Python modules, supported by libraries such as OpenCV, MediaPipe, NumPy, and Trimesh.

### 1. Landmark Detection

The pipeline begins by loading a user-provided image of a face. This image is passed to MediaPipe's Face Mesh model in static image mode, which returns 3D coordinates (x, y, z) for 468 standardized facial landmarks. These coordinates are relative to the image dimensions, with the z-coordinate derived from facial depth estimation.

To maintain consistency across images of varying size and resolution, the x and y positions are scaled according to the image width and height respectively, while z is scaled by width alone. This ensures all landmark positions lie in a comparable coordinate space.

### 2. Landmark-to-Vertex Mapping

To map 2D image landmarks to the 3D model vertices, an initial alignment is required. A small set of manually chosen MediaPipe landmarks are paired with known vertices on the base mesh that correspond to facial features like the nose tip, eye corners, and lips. These matched pairs are used to compute a **similarity transformation** (scale, rotation, and translation) that brings the image landmarks into alignment with the 3D model space.

Once aligned, each of the 468 transformed landmarks is assigned to the closest vertex in the 3D mesh by calculating the Euclidean distance between the landmark and all mesh vertices. This mapping is saved to disk for reuse, ensuring consistent correspondence for subsequent morphing operations.

### 3. Mesh Normalization and Deformation

Before deformation begins, the base 3D mesh is normalized to **unit space**, which involves translating its center to the origin and scaling it to fit within a unit cube. This step standardizes the mesh across different coordinate systems and ensures that deltas applied later are proportional and stable.

Next, for each new facial image, the pipeline extracts the 3D landmarks, aligns them to the reference space using the same similarity transformation method, and calculates the positional **deltas** between these landmarks and the corresponding reference landmarks. These deltas are then transferred to the mapped mesh vertices.

The deformation step supports customization through two parameters:

* **strength**: a scalar that scales how much each vertex is moved by its corresponding delta. A value of 0.5, for example, means only half of the full deformation is applied.
* **ignore\_z**: a boolean that, if true, zeroes out the z-component of the delta. This allows morphing to follow only the x-y image features, avoiding potential instability or noise in the depth dimension.

### 4. Laplacian Smoothing

To avoid unnatural deformation and ensure smooth transitions between vertices, **Laplacian smoothing** is applied after delta application. This iterative process adjusts the position of each vertex based on the average position of its neighbors, effectively diffusing sharp distortions across the surface. The smoothing is controlled by:

* **lambda**: a scalar weight controlling the strength of the smoothing.
* **iterations**: the number of times the smoothing operation is applied.

The smoothed mesh is then exported in standard .obj format for visualization and further use.

### 5. Error Quantification and Heatmap Visualization

To evaluate how accurately each morphed mesh matches the intended facial shape, the pipeline calculates the **per-landmark error**, defined as the Euclidean distance between each transformed 3D landmark and the mesh vertex it is mapped to. These errors are averaged to compute a single scalar “landmark error” for each image.

Furthermore, the individual errors are stored per landmark and visualized as a **heatmap** on the 3D mesh. Vertex colors are determined based on error magnitudes, allowing users to see which facial regions are morphing accurately and which are not. In addition, histograms and boxplots are generated across all processed images to show error distribution, variability, and outliers.

## Results

The pipeline was tested on a small set of facial images, each generating a personalized 3D mesh. Landmark errors ranged from approximately 0.05 to 0.19 (in normalized unit space). Qualitative visual inspection confirmed that the deformations followed facial contours accurately, particularly around prominent features such as the nose, mouth, and jawline.

Tests with ignore\_z=True and ignore\_z=False showed negligible difference in the final landmark error, implying that the z-coordinate from MediaPipe is either accurate enough to retain or sufficiently noisy to be ignored without significant loss of quality.

The strength parameter had a visible effect on the deformation intensity. A value of 0.2 provided a good tradeoff between recognizable change and mesh stability. Laplacian smoothing with 500 iterations and a lambda of 0.06 successfully reduced local noise without too much loss, especially around high-curvature areas like eyes and lips. However it must be noted that finding more optimal values would improve the results and therefore is advisable.

## ****Conclusion****

This project demonstrates a complete and extensible system for morphing 3D face meshes using 2D images. Key technical contributions include:

* A robust landmark-to-vertex mapping pipeline using similarity transforms
* A tunable mesh morphing procedure with optional depth control
* Laplacian smoothing for quality preservation
* A quantitative and visual evaluation framework

Future work could involve enhancing texture warping, or replacing MediaPipe with 3D-aware models like DECA for better depth accuracy. Nonetheless, the current implementation provides a flexible foundation for 3D face modeling from single images.