# 3D Face Reconstruction and Facial Expression Transfer

## Contents

# 1 Understanding 3D Face Reconstruction

## 1.1 Principles and Components of 3D Face Reconstruction

3D face reconstruction is the process of creating a 3D version of a face from one or more 2D pictures. This process involves several important steps and components:

- **Landmark Detection**: This step involves finding key points on a person's face, such as the position of the eyes, nose, mouth, and chin in a 2D image. These landmarks help guide the reconstruction of the 3D face.
- **3D Model Fitting**: 3D Model Fitting: A basic 3D face shape (often called a "template" or "model") is adjusted to match the positions of the detected landmarks from the 2D image. Think of it like taking a 3D head shape and stretching or squishing it until it fits the person's unique features.
- **Texture Mapping**: This step takes the color and details (like skin tone, freckles, or facial hair) from the 2D image and applies them to the 3D model. This helps the 3D face look like the original person.
- **Pose and Expression Estimation**:The system determines the person's facial expression (e.g., smiling, frowning) and the angle of their face (whether they are looking straight ahead, to the side, up, or down). This makes the 3D model feel more realistic and natural.

The most common model used is the **3D Morphable Model (3DMM)**.It's a flexible model that can change its shape and texture by adjusting different parameters, allowing it to recreate almost any face. If the person is smiling or frowning, the 3DMM with Expressions can adjust to show these movements too, making the model look more life like. Think of 3DMM like a clay model of a face that can be shaped to look like any person by moving and adjusting parts of the face (like making the nose longer or the chin wider). The texture is then "painted" onto this model to give it color and detail, making the 3D face look like the real person from the 2D image.
**This process allows us to recreate any face in 3D from just a picture!**

## 1.2 Importance of Accurate 3D Face Models

Accurate 3D face models are important for tasks like facial expression transfer because they:

- Help capture detailed facial movements.
- Ensure the expressions transferred look realistic.
- Allow adjustments for differences in head angles or lighting, which are common in 2D images.

# 2 Choosing a 3D Face Reconstruction Model

When choosing a 3D face reconstruction model, the selection depends on the requirements of the task. Some models prioritize speed, while others focus on accuracy and realistic facial reenactment. In this section, we compare three popular models used for 3D face reconstruction and expression transfer: **FaceMesh**, **3DDFA**, and **Deep3DFace**.

## 2.1 FaceMesh

FaceMesh detects **468 facial landmarks** from 2D images and provides limited 3D information. The model is optimized for speed and real-time applications, which makes it ideal for live video processing, but it lacks the detailed reconstruction needed for complex facial expressions or head movements.

**Strengths**:

- **Speed**: FaceMesh is designed for real-time applications, enabling quick facial landmark detection for use in live video feeds, and other fast-processing applications.
- **Efficiency**: Because it doesn't capture fine details, FaceMesh can run smoothly on devices with lower computational power, such as smartphones.

**Limitations**:

- **Limited Detail**: FaceMesh provides only basic 3D shape information, which is sufficient for simple applications, but it cannot accurately capture complex facial expressions or fine head movements.
- **Not Suitable for Complex Tasks**: If detailed expression analysis or reenactment is needed, FaceMesh may not provide the required level of detail.

**Example Use**: FaceMesh is ideal for real-time applications like Snapchat filters, where speed is important, and simple face tracking is enough.

### 2.2 3DDFA (3D Dense Face Alignment)

The 3DDFA model uses **deep learning** to generate a dense set of 3D landmarks, enabling more accurate facial expression capture and handling more complex situations such as partial face turns or difficult lighting conditions.

**Strengths**:

- **Accuracy**: 3DDFA can handle complex expressions and head poses, and works well even in poor lighting conditions. This makes it more suitable for detailed facial analysis.
- **Versatility**: It can still detect and recreate 3D landmarks even if the face is rotated or the lighting is uneven.

**Limitations**:

- **Slower than FaceMesh**: Since 3DDFA captures more detail, it requires more time to process each image compared to FaceMesh. It might not be suitable for low-powered devices in real-time applications.
- **Requires More Power**: 3DDFA needs higher computational power to run efficiently, particularly on mobile devices.

**Example Use**: 3DDFA is ideal for facial recognition systems and emotion detection tasks where accuracy is crucial and some processing time can be sacrificed.

### 2.3 Deep3DFace

Deep3DFace is the most advanced model in this comparison, using a **deep neural network** to provide highly detailed 3D face reconstructions, including accurate facial expressions and fine details like skin texture. This model excels in producing realistic faces and is suited for tasks that require high accuracy.

**Strengths**:

- **High Detail and Realism**: Deep3DFace captures fine details of the face.
- **Expression Transfer**: Deep3DFace excels at transferring facial expressions from one face to another with precision, making it perfect for applications like video game avatars or virtual reality.

**Limitations**:

- **Processing Time**: Deep3DFace takes the longest to process among the three models due to the high level of detail it captures, making it less suitable for real-time applications.
- **High Computational Requirements**: This model requires a powerful system with advanced GPU capabilities to run efficiently, particularly when processing large datasets.

**Example Use**: Deep3DFace is ideal for applications, where realistic and detailed 3D face modeling is required, and processing time is less of a concern.

**In summary**:

- If speed and real-time performance are the priority, **FaceMesh** is the best option.
- If more accuracy in facial expressions is required, **3DDFA** offers a balanced choice between speed and detail.
- For the highest quality and realism in 3D face reconstruction and expression transfer, **Deep3DFace** is the best model, though it requires more processing power.

## 3 Facial Expression Transfer

Facial expression transfer involves capturing expressions from one face (source) and applying them to another (target) while keeping the target face's identity. This process requires accurately aligning facial features and transferring expressions without causing distortions.

### 3.1 Capturing and Encoding Facial Expressions

- **FaceMesh-based Approach**: The FaceMesh model from MediaPipe detects 468 facial landmarks on a 2D image. These landmarks correspond to important areas of the face, such as the eyes, nose, mouth. While FaceMesh provides some 3D information, it primarily relies on 2D projections of the face. The facial expressions are encoded by analyzing how these landmarks move from one frame (source) to another (target). For example, if a person smiles, the position of the mouth landmarks changes, and this change is tracked and encoded.
- **3DDFA (3D Dense Face Alignment)**: 3DDFA uses a deep learning-based approach to detect a dense set of 3D facial landmarks. Unlike FaceMesh, 3DDFA works in full 3D space, capturing both the facial structure and the expressions more accurately. This model encodes expressions by adjusting the shape and position of the 3D landmarks to match the source face. The model identifies how different facial features change during an expression and encodes this movement in 3D space.
- **Deep3DFace**: Deep3DFace reconstructs the entire 3D face mesh **A mesh is a network of vertices (points) and edges (lines connecting the points) that define the surface geometry of the face.**, including both the face's shape and texture. It encodes expressions through shape deformations, capturing small changes in the face, like wrinkles or muscle movements.

## 3.2 How Expression Transfer Happens

Facial expression transfer involves taking the facial expressions from one person (the **source face**) and applying them to another person's face (the **target face**). The goal is to keep the target face looking like itself while showing the same expressions as the source face. Different models handle this process in different ways.

## 3.3 FaceMesh-based Approach

In the **FaceMesh-based approach**, the face is divided into small triangular regions using a technique called **Delaunay triangulation**. These triangles represent different parts of the face, such as the forehead, cheeks, or around the eyes.

Here's how the process works:

- **Triangular Division**: The face is split into triangles based on facial landmarks (points like the corners of the eyes, mouth, and nose).
- **Affine Transformation**: The triangles on the source face are warped (stretched or adjusted) to match the triangles on the target face. This warping uses a method called **affine transformation**, which allows the triangles to be scaled, rotated, and shifted to fit the target face.
- **Blending**: Once the triangles are aligned, they are blended onto the target face, which completes the expression transfer.

However, since FaceMesh primarily works in **2D space**, there can be distortions if the source and target faces have different poses (e.g., one face looking forward, the other slightly turned). This means that sometimes the transferred expression may not look perfectly natural, especially if the faces don't align well in terms of shape or position.

## 3.4 3DDFA (3D Dense Face Alignment)

The **3DDFA** model works differently by using a **3D Morphable Model (3DMM)** to capture both the structure of the face and its expressions. This model fits a 3D face shape to both the source and target faces, making sure that the transfer happens accurately.

Here's how it works:

- **3D Fitting**: 3DDFA fits a flexible 3D face model (3DMM) to both the source and target faces. This model can change its shape to match the unique features of each person's face.
- **Expression Transfer**: Once the 3D model is fitted, the source face's expressions (like smiling or frowning) are transferred to the target face by adjusting the target's 3D face shape to match the expression.
- **Texture Transfer**: Along with the shape, the texture (the appearance of the face, like skin tone and details) is also transferred, ensuring the final expression looks natural and realistic.

Because 3DDFA works in **3D space**, it handles different facial angles and lighting much better than the FaceMesh-based approach. It's also less likely to create distortions, making the transferred expression look more natural.

## 3.5 Deep3DFace

The **Deep3DFace** model offers the most advanced method for transferring facial expressions because it works with the entire **3D face mesh**, capturing both the face's shape and fine details like wrinkles or muscle movements.

Here's how it works:

- **3D Face Mesh Reconstruction**: Deep3DFace builds a detailed 3D mesh of both the source and target faces. This mesh represents the shape of the face in full 3D, with thousands of tiny points connecting to form a surface.
- **Expression Warping**: The expression from the source face is warped onto the target face. This means the shape of the target's 3D mesh is adjusted to match the facial expressions of the source face. This warping ensures that small movements, like the muscles around the mouth or the subtle changes in the eyes during a smile, are transferred accurately.
- **Texture Mapping**: Deep3DFace also transfers the texture of the face from the source to the target, which includes things like skin tone, lighting effects, and fine details like freckles or wrinkles. This makes the final result look incredibly realistic.

Since Deep3DFace works with the entire **3D face mesh**, it captures even the smallest details and movements, making the expression transfer much more accurate and lifelike. It's especially useful for tasks that need high realism.

# 4 Realism and Expressiveness

## 4.1 Ensuring Realism

- **FaceMesh-based Approach**: This method may cause distortions if the source and target faces have different head poses or facial structures.
- **3DDFA**: By fitting a 3D Morphable Model, 3DDFA makes expression transfers smoother and more realistic.
- **Deep3DFace**: This model captures subtle details like wrinkles and skin shading, making the expression transfer highly realistic.

## 4.2 Challenges in Realism

- **Lighting Mismatch**: Different lighting on the source and target faces can make the transfer look unnatural.
- **Pose Variations**: If the source and target faces are positioned differently (e.g., one facing forward, the other turned), the transferred expression may be distorted.
- **Facial Feature Differences**: Differences in the shape and size of features like the eyes, nose, or mouth can cause misalignment.
- **Expression Intensity Variations**: The intensity of the expression might not match the target face's natural style, making the expression look exaggerated.

  **Solutions**: More advanced models like Deep3DFace can overcome these challenges by taking lighting, head position, feature shapes, and expression intensity into account, resulting in more realistic and natural expression transfers.

# 5 Evaluating Expression Transfer Quality

To measure the quality and accuracy of the facial expression transfer, we use different metrics that compare the source and target faces. These metrics help determine whether the transferred expression looks natural and realistic.

## 5.1 Illustrating the Three Key Metrics

We used the following three metrics to evaluate the quality of expression transfer:

- **RMSE (Root Mean Square Error) for 2D Landmarks**:
  - **What it measures:** RMSE calculates the difference between the positions of key facial landmarks (such as the eyes, nose, and mouth) on the source face and the target face after the expression transfer.
  - **Why it matters:** A low RMSE value indicates that the positions of these landmarks are similar, meaning the expression transfer is accurate.
- **MAE (Mean Absolute Error) for 3D Landmarks**:
  - **What it measures:** MAE calculates the average difference in the 3D positions of the facial landmarks between the source and target faces. This metric assesses how well the 3D structure has been transferred.
  - **Why it matters:** Lower MAE values suggest that the 3D structure of the face has been accurately transferred, maintaining the face's shape during expression transfer.
- **SSIM (Structural Similarity Index)**:
  - **What it measures:** SSIM compares the overall visual appearance of the source and target faces after expression transfer. It takes into account factors like brightness, contrast, and texture to determine the similarity between the two images.
  - **Why it matters:** A higher SSIM value (close to 1) indicates that the target face looks visually similar to the source face, meaning the expression transfer appears realistic.

## 5.2 Experiments and Results

The following experiments were conducted using our implementation, which is based on the **FaceMesh** model for expression transfer. We performed two test cases, each of which includes the following steps:

- A **source image**, containing the face with the expression to be transferred.
- A **target image**, containing the face that will receive the expression.
- A **source image after landmark detection**, showing detected facial landmarks.
- A **target image after landmark detection**, showing detected facial landmarks.

- The **3D landmarks of the source face**.
- The **3D landmarks of the target face**.
- A **final output image**, showing the target face with the transferred expression.

### 5.2.1 Test Case 1

In the first test case, the following steps were performed:

Figure 1: Test Case 1: Source Image (left), Target Image (right)

Figure 2: Test Case 1: Source Image with Detected Landmarks (left), Target Image with Detected Landmarks (right)

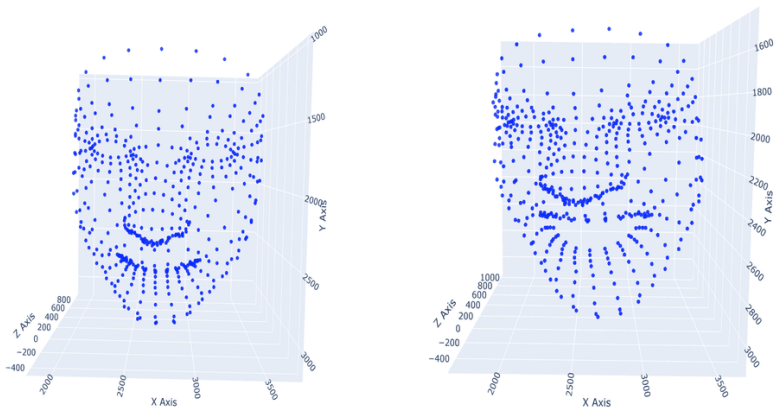


Figure 3: Test Case 1: 3D Landmarks of Source Image (left), 3D Landmarks of Target Image (right)

Figure 4: Test Case 1: Final Output Image with Transferred Expression

**Metric Results for Test Case 1:**

- **RMSE (2D Landmarks)** : 168.71340692195648
- **MAE (3D Landmarks)** : 110.82417733213424
- **SSIM (Structural Similarity Index)** : 0.6687974867383045

#### 5.2.2 Test Case 2

In the second test case, the same steps were repeated with a different set of source and target images:


Figure 5: Test Case 2: Source Image (left), Target Image (right)

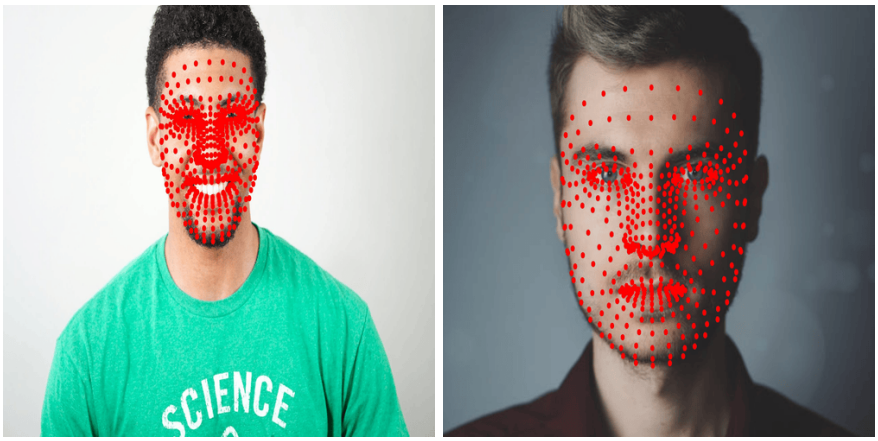


Figure 6: Test Case 2: Source Image with Detected Landmarks (left), Target Image with Detected Landmarks (right)

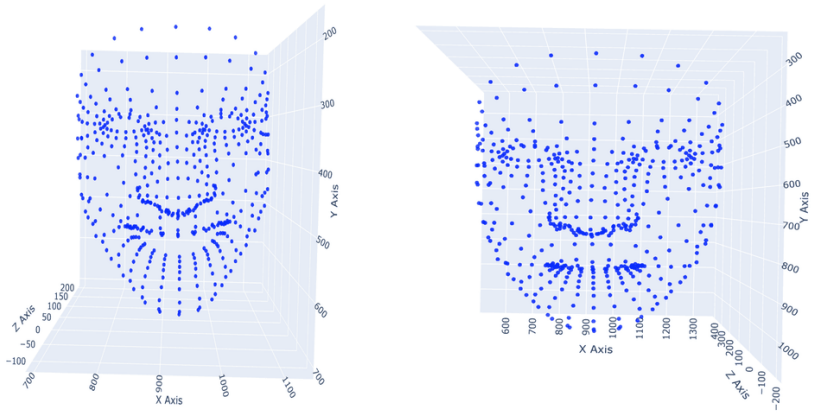


Figure 7: Test Case 2: 3D Landmarks of Source Image (left), 3D Landmarks of Target Image (right)

Figure 8: Test Case 2: Final Output Image with Transferred Expression

**Metric Results for Test Case 2:**

- **RMSE (2D Landmarks)**: 187.73667398961706
- **MAE (3D Landmarks)**: 116.78469973386001
- **SSIM (Structural Similarity Index)**: 0.5454108574214 28

# 6 Challenges and Improvements

## 6.1 Challenges Encountered

During the implementation of the facial expression transfer system, several challenges were encountered, particularly when trying to work with advanced models like **3DDFA** and **Deep3DFace**. While these models provide highly accurate results, they come with significant difficulties:

- **Complexity of Setup**: Both 3DDFA and Deep3DFace require cloning complex repositories with numerous dependencies. Ensuring compatibility with the required libraries, such as TensorFlow and PyTorch, can be tricky and time-consuming. Additionally, some of these repositories are not well-tested on Windows systems, further complicating the setup process on non-Linux environments.
- **Heavy Computational Requirements**: Models like **Deep3DFace** specifically recommend using **GPU rendering** to ensure efficient processing, as running on standard CPUs can be too slow. Additionally, Deep3DFace is optimized for **Linux operating systems**, particularly during the rendering phase, which may cause issues for users on other platforms like Windows.

Given these challenges, we opted to use **FaceMesh** for our implementation. FaceMesh is lightweight, requires minimal setup, and can run efficiently on most machines without the need for specialized hardware like GPUs. While FaceMesh lacks some of the accuracy and detail of 3DDFA and Deep3DFace, it offers a much simpler and more accessible solution, making it suitable for real-time applications and basic expression transfer tasks.

### 6.2 Proposed Improvements and Extensions

While **FaceMesh** offers a simple and efficient solution for basic expression transfers, integrating more advanced models like **3DDFA** or **Deep3DFace** could significantly improve the accuracy and realism.

- **3DDFA** would provide more precise 3D facial modeling, better capturing fine details and complex head poses.
- **Deep3DFace** could enhance the texture mapping, adding realism with finer features like wrinkles and skin texture.

## 7 Conclusion

In this report, we discussed the process of 3D face reconstruction and facial expression transfer. Below is a summary of the key points:

- **FaceMesh**:
  - Fast and suitable for real-time applications.
  - Limited accuracy in capturing detailed facial expressions and 3D poses.
- **3DDFA**:
  - Provides a good balance between accuracy and speed.
  - Suitable for more accurate facial expression transfers, even in complex lighting and poses.
- **Deep3DFace**:
  - Delivers the most realistic and detailed 3D reconstructions.
- **Our Implementation**:
  - We used **FaceMesh** for simplicity and to run efficiently on standard devices.
- **Future Improvements**:
  - Using models like 3DDFA or Deep3DFace can improve accuracy and realism.

## 8 References


- **3DDFA GitHub Repository**: A PyTorch implementation of 3DDFA for 3D dense face alignment, which provides accurate face alignment even in cases of large pose variations.
  GitHub: 3DDFA
- **Deep3DFace (TensorFlow) GitHub Repository**: This repository provides a TensorFlow implementation of Deep3DFace, a model that reconstructs 3D faces from 2D images with high fidelity.
  GitHub: Deep3DFace (TensorFlow)
- **3DFMA - 3D Face Morphable Model**: An article explaining the fundamentals of 3D Morphable Models (3DMM) and how they are used in 3D face reconstruction and manipulation.
  Article: 3D Morphable Model
- **3D Face Reconstruction: Make a Realistic Avatar from a Photo**: This article provides insights into creating realistic 3D face avatars from photos, explaining the steps and techniques involved in the process.
  Article: 3D Face Reconstruction