A Comprehensive Overview of 3D Face Recognition Techniques: From Conventional Methods to Deep Learning

Shailesh Kumar (224EC6013) ,
Signal and Image Processing
Department of ELectronics and Communication Engineering,
NAtional Institute of Technology, Rourkela

***Abstract*—3D face recognition (3DFR) has gained significant attention due to its robustness to variations in pose, lighting, and expression. This paper presents an in-depth review of key developments in the field, emphasizing recent lightweight deep learning models, synthetic data training, and quality-fusion strategies. We also explore the mathematical and architectural underpinnings of popular methods, analyze public datasets, and examine experimental benchmarks. Lastly, we discuss real-world applications, challenges, and emerging directions.**

***Index Terms*—3D face recognition, point clouds, MobileNetV2, MQFNet, PointNet++, FRNet, deep learning, synthetic data, pose variation.**

I. Introduction

Face recognition is a widely used biometric technology that helps in security, surveillance, and authentication systems. Traditionally, most face recognition systems have used 2D images. These methods work well in controlled environments, but their performance drops when there are changes in lighting, facial expressions, or head poses.

To solve these problems, researchers have started using 3D face recognition. 3D face data includes depth and shape information, making it more robust to lighting and pose variations. Unlike 2D images, 3D models can better capture facial geometry, leading to higher recognition accuracy.

A 3D face model is usually collected using special hardware like 3D scanners or depth cameras. These devices create detailed surface maps using either active methods (like lasers) or passive methods (like stereo cameras). Although this makes data collection harder than with 2D images, the extra geometric detail can greatly improve recognition performance.

Over the years, many 3D face datasets have been created, such as FRGC, Bosphorus, and BU3D-FE. These datasets are essential for training and testing new algorithms.

Earlier research focused on traditional (or conventional) methods that extract features using algorithms like PCA or ICP. More recently, deep learning has become popular for 3D face recognition. Deep learning models can automatically learn features from 3D data by converting it into 2D images or maps.

This paper reviews both conventional and deep learning-based approaches for 3D face recognition. It also compares these methods and discusses the challenges and future directions in the field.

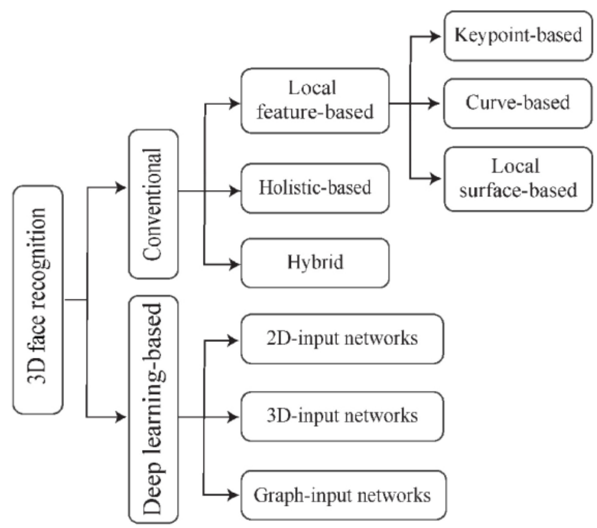

Fig. 1. An example 3D face model and its depth map.

II. 3D Face Database

Large-scale 3D face databases are very important for training and evaluating face recognition algorithms. These datasets contain 3D scans of human faces with variations in expression, pose, and lighting.

There are different types of 3D data formats used:

- **Point Clouds:** A set of points in 3D space representing the face.
- **Meshes:** Points connected with triangles to form a surface.
- **Range Images / Depth Maps:** 2D images with depth information.
- **3D Videos:** Sequences of 3D frames showing facial motion.

3D face data is collected using two main methods:

- **Active Scanning:** Uses structured light or laser beams (e.g., Kinect, Minolta).
- **Passive Scanning:** Uses multiple cameras to estimate depth.

Some well-known 3D face datasets are:

- **FRGC v2.0:** A standard benchmark with thousands of scans.
- **BU3D-FE:** Designed for facial expression analysis.
- **Bosphorus:** Includes occlusions and pose variations.
- **Texas-3D:** Offers annotated facial landmarks.
- **LS3DFace:** The largest dataset with over 30,000 scans.

These datasets help researchers develop and compare 3D face recognition algorithms. However, collecting 3D data is difficult and expensive. More work is needed to build large, diverse datasets for real-world applications.

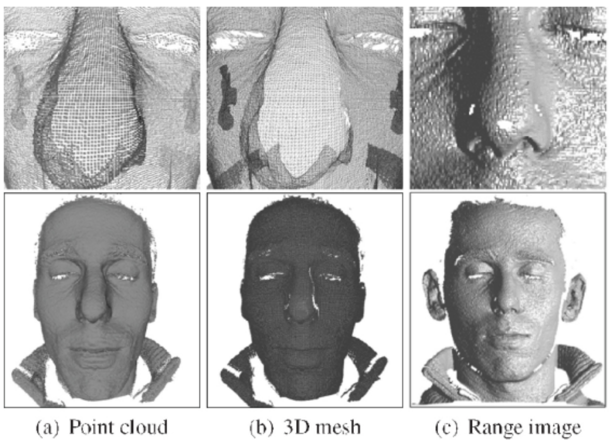

Fig. 2. Example of 3D face scan compared to 2D image. 3D data contains richer structural cues.

III. Conventional Methods

In traditional 3D face recognition systems, the process usually consists of two stages: **training** and **testing**, as shown in Fig. 3. During the training phase, 3D face scans are collected, preprocessed, and used to extract feature vectors. These vectors are stored in a gallery. During the testing phase, a new face scan is processed in the same way, and the extracted features are matched against those in the gallery. If the similarity score is high enough, the identity is confirmed.

The recognition process includes three core components:

- Data preprocessing
- Feature extraction
- Face matching

Each of these steps plays a crucial role in the accuracy and robustness of the system.

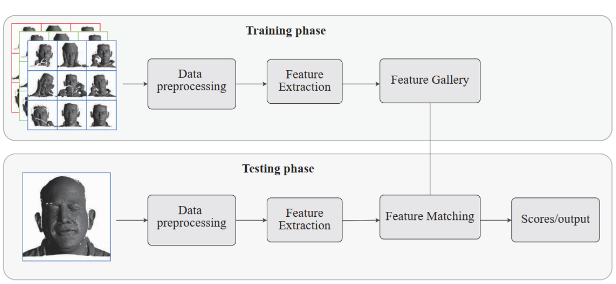

Fig. 3. Pipeline of a conventional 3D face recognition system.

A. A. Data Preprocessing and Matching

Raw 3D face data often includes noise, irrelevant parts (like hair and shoulders), and alignment errors. Preprocessing is needed to clean and standardize the input. Common steps include:

- **Landmark Detection:** Detecting key facial points such as nose tip, eyes, and mouth corners.
- **Face Segmentation:** Isolating the facial region using the detected landmarks.
- **Face Registration:** Aligning all faces to a common pose using techniques like Iterative Closest Point (ICP).

Once preprocessed, the system compares the new face's feature vector with stored vectors using distance metrics like:

- Euclidean distance
- Mahalanobis distance
- Hausdorff distance

If the distance is below a set threshold, the face is recognized; otherwise, it is rejected.

B. B. Local Feature-Based Methods

These methods extract features from small, rigid areas of the face like the nose or eyes. They are robust against expressions and partial occlusion. Local methods are generally categorized into:

- **Keypoint-Based:** Detect keypoints on the face (e.g., using SIFT or MeshDOG) and calculate features based on their spatial relationships.
- **Curve-Based:** Use curves (e.g., radial or level curves) on the facial surface to form descriptors.
- **Local Surface-Based:** Use texture or geometric features from specific facial regions, such as Local Binary Pattern (LBP) or curvature-based descriptors.

Keypoint-based methods are flexible to missing data but require good keypoint selection. Curve-based methods often start curves from the nose tip, using radial or iso-geodesic curves. Surface-based methods divide the face into patches and extract features from each.

Fusion Techniques are often used to improve recognition, such as:

- Feature-level fusion
- Score-level fusion
- Decision-level fusion

These combine multiple sources of information to improve recognition rates.

C. C. Holistic-Based Methods

Holistic methods treat the entire face as a single unit. They are effective when the face is fully visible and frontal. Common techniques include:

- **Principal Component Analysis (PCA):** Reduces dimensionality by finding the main variations in the dataset (also known as Eigenfaces).
- **Linear Discriminant Analysis (LDA):** Maximizes the separability between different individuals.

- **Spherical Harmonic Features (SHF):** Projects 3D faces onto a sphere to extract frequency-based features.
- **ICP-Based Alignment:** Aligns face point clouds and compares them directly.

Although fast and simple, holistic methods are more sensitive to pose changes and occlusions.

D. D. Hybrid Methods

Hybrid methods combine both local and holistic features. For example, they may use local descriptors (like LBP) to capture detailed features, and global descriptors (like PCA) to understand the full face shape.

These methods can achieve higher accuracy but are more complex and require careful alignment. Hybrid methods are especially useful in dealing with real-world data where expressions, occlusion, and pose variation are common.

In summary:

- Local methods are robust and detailed but computationally intensive.
- Holistic methods are fast but fragile to variations.
- Hybrid methods combine strengths but need careful implementation.

IV. Deep Learning-Based 3D Face Recognition

In recent years, deep learning has become the most popular and powerful technique for face recognition. Unlike traditional methods that rely on hand-crafted features, deep learning models—especially Convolutional Neural Networks (CNNs)—can automatically learn and extract useful facial features from raw data. This has led to significant improvements in both speed and accuracy.

A. A. Background: 2D Face Recognition with Deep Learning

Before applying deep learning to 3D face recognition, many successful models were developed for 2D face recognition:

- **DeepFace:** A 9-layer CNN trained on 4 million labeled face images, achieving over 97% accuracy on the LFW dataset.
- **DeepID Series:** These networks learned face features from different parts of the face and improved accuracy using multiple layers and loss functions.
- **GoogLeNet and FaceNet:** Used deeper networks and triplet loss to directly map faces into a compact Euclidean space.
- **VGG-Face:** A widely used model trained on millions of 2D images, often used as a base for transfer learning in 3D applications.
- **ResNet:** Introduced residual connections, allowing networks with more than 100 layers to be trained effectively.

These models showed excellent performance and inspired researchers to adapt similar architectures for 3D face recognition.

B. B. Adapting CNNs for 3D Face Recognition

Since most CNNs are designed for 2D images, 3D data is usually converted into 2D formats, such as:

- **Depth maps:** Projecting the 3D point cloud onto a 2D plane.
- **Surface normal maps:** Encoding the direction of surface normals into RGB images.
- **Curvature maps:** Showing geometric features like shape index or Gaussian curvature.

Once converted, the 3D face data can be passed to CNNs similar to 2D input.

1) 1) VGG-Based Models: Kim et al. used the VGG-Face model and fine-tuned it with 3D depth maps. They applied data augmentation by generating new 3D expressions and poses using a 3D Morphable Model (3DMM). The depth maps were cropped and resized to 224×224 to fit the CNN input. Their model achieved recognition accuracy of 99.2% on Bosphorus and 95.0% on BU3D-FE datasets.

FR3DNet, another VGG-based model, was trained on a much larger dataset of over 3 million synthetic 3D faces. It used three-channel inputs: depth, azimuth, and elevation maps. The model reached 98.64% accuracy on BU3D-FE.

2) 2) ResNet-Based Models: ResNet architectures have also been adapted for 3D face recognition:

- **Pre-ResNet models** were trained with triplet loss and softmax loss, showing excellent results across multiple datasets.
- A method using Voronoi diagrams split the face into 13 parts and used each for feature extraction. It achieved 99.7% on Bosphorus.
- Another method trained ResNet-18 to handle low-quality 3D data, using sparse point clouds and a registration network to reconstruct the full face.

3) 3) Other Architectures: Other CNN variants were also explored:

- **MobileNetV2:** Designed for mobile use, this model is lightweight and fast. It used geometric descriptors like shape index and curvedness to enhance accuracy.
- **Dual-Channel CNNs:** These networks combine 2D texture and 3D depth maps to improve robustness and accuracy.
- **PointNet++:** Directly processes raw 3D point clouds without converting to images, making it efficient for irregular data.

C. C. Key Elements in Deep Learning-Based 3D Recognition

There are three important parts in any deep learning-based system:

1) **Training Dataset:** A large and diverse dataset is needed. Many methods generate synthetic 3D faces to increase the dataset size.
2) **Data Preprocessing:** This includes detecting the nose tip, aligning the face, and converting 3D data into suitable formats.

3) **Network Design:** The choice of architecture (VGG, ResNet, MobileNet, etc.) affects both performance and speed.

D. D. Summary of Performance

Many deep learning methods achieve over 95% recognition accuracy. For example:

- **FR3DNet:** 98.6% on BU3D-FE.
- **Pre-ResNet:** Over 99% on multiple datasets.
- **MobileNetV2-based:** 97.5% on Bosphorus.

Although deep learning-based methods perform very well, challenges still exist:

- Lack of large-scale real 3D datasets.
- Sensitivity to poor alignment and occlusion.
- High computational cost in some methods.

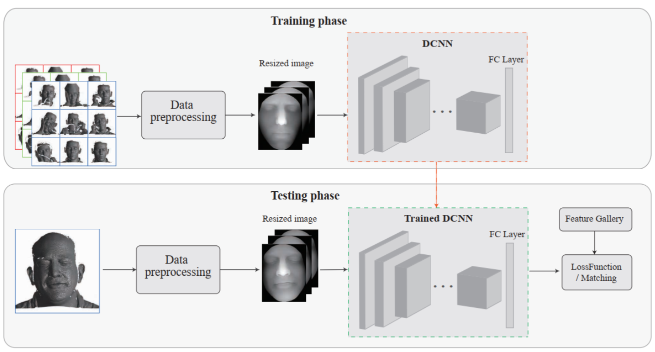

Fig. 4. Overview of a deep learning-based 3D face recognition pipeline.

Future work may focus on building more diverse datasets, improving point cloud processing, and designing efficient architectures for real-time applications.

V. Experimental Results

TABLE I
Recognition Accuracy on Benchmark Datasets (Rank-1)

Method	FRGC v2	Bosphorus	BU3D-FE	UMBDB
Olivetti et al.	95.6%	97.56%	-	-
Lin et al.	-	-	98.1%	98.2%
Zhang et al.	97.3%	99.1%	98.6%	-
Tan et al.	-	-	-	99.2%

VI. Discussion

3D face recognition has improved significantly in terms of accuracy, robustness, and dataset availability. Both traditional and deep learning methods have their own strengths and weaknesses.

A. A. Conventional Methods

Conventional methods are categorized into:

- **Local Feature-Based:** Use small face areas like nose or eyes. They are robust to occlusion and expressions.
- **Holistic-Based:** Use the full face for recognition. They work well for neutral faces but are sensitive to pose and occlusion.
- **Hybrid Methods:** Combine local and global features for better overall performance.

B. B. Deep Learning Methods

Deep learning models, especially CNNs, have higher accuracy and simpler pipelines. They do not require manual feature selection but depend heavily on large training datasets and good preprocessing.

C. C. Challenges and Future Work

Key challenges and directions include:

- **Data Augmentation:** Generating more diverse 3D face samples is essential.
- **Preprocessing:** Better alignment and conversion from 3D to 2D formats can improve performance.
- **Network Design:** Choosing suitable CNN architectures and loss functions helps improve learning.
- **Larger Datasets:** Public 3D datasets need to grow to support deeper networks.

Combining traditional keypoint techniques with CNNs and handling low-quality or mobile-scanned faces are also important future directions.

VII. Related Work

Earlier studies used handcrafted features such as Shape Index Histograms and curvature descriptors. While effective under limited variations, these approaches struggled with generalization.

Recent surveys (e.g., by Bowyer et al. and Smeets et al.) provided early taxonomies but lacked deep learning insight. Jing et al.'s work fills that gap by comprehensively comparing modern neural methods.

VIII. Challenges and Future Directions

- **Dataset Diversity:** Current benchmarks are limited in ethnic and age representation.
- **Cross-Domain Learning:** 2D-3D fusion and domain adaptation are underexplored.
- **Low-Light 3D Capture:** Sensors struggle in darkness.
- **Real-Time Constraints:** Lightweight models are needed for mobile inference.

Emerging Trends:

- Synthetic data and simulation environments
- Transformer-based 3D models
- Federated learning for privacy-preserving 3DFR

IX. Conclusion

3D face recognition continues to evolve with deep learning and novel data processing pipelines. From lightweight MobileNetV2 to robust fusion networks and point-based architectures, the diversity of solutions demonstrates the field's maturity. However, real-world deployment still demands advances in sensor quality, dataset scale, and algorithmic generalization. By addressing these challenges, 3DFR can become a staple in secure and intelligent biometric systems.

References

[1] Y. Jing, X. Lu, and S. Gao, "3D face recognition: A comprehensive survey," *Comput. Visual Media*, vol. 9, no. 4, pp. 657–685, 2023.

[2] S. Olivetti, S. Calderara, and R. Cucchiara, "Geometric Deep Learning using MobileNetV2," *Proc. ICPR*, pp. 1235–1242, 2020.

[3] Y. Lin, B. Fang, and Z. Zhang, "MQFNet: Multi-quality fusion network," *Sensors*, vol. 21, no. 9, pp. 2981–2995, 2021.

[4] Y. Zhang, X. Li, and H. Wang, "PointNet++ based 3D face recognition," *Proc. CVPRW*, Jun. 2021, pp. 145–154.

[5] H. Tan, K. Mu, and Y. Feng, "Lightweight 3D face recognition," *IEEE Access*, vol. 8, pp. 204908–204917, Nov. 2020.