Recovering Occluded 3D Face Models for 3D Face Recognition

Submitted by:
Macaulay Sadiq
Antenne Sciences et Technique, Centre Universitaire Condorcet,
Universite de Bourgogne, France

Supervisors:
Dr. Djamila Aouada
Dr. Anis Kacem
Dr. Konstantinos Papadopoulos
Interdisciplinary Centre for Security, Reliability and Trust (SnT),
University of Luxembourg

Academic Supervisor:
Professor Lew Yan Voon
Département GEII, University of Bourgogne, IUT Le Creusot,
France

A Thesis Submitted for the Degree of
MSc in Computer Vision (VIBOT)
· 2020 ·

Abstract

Face recognition is an approach that use facial textures, shapes and patterns to analyse unique features on faces for identification purposes. Since the mid-1960s, a significant amount of work has been conducted towards recognizing humans based on facial features. In recent times, face recognition has attracted attention in multiple applications such as law enforcement, access control on mobile platforms, industrial systems and in various commercial domains. While the conventional approaches of face recognition use 2D images in which the geometry information of the face are described with light reflection properties, several challenges have emerged. Most significant challenges are the variation of lightning conditions, facial expressions, and occlusions. With the development of 3D-sensors, most recent approaches on face recognition have been directed toward the use of 3D face data to analyze the geometric information on the face. This approaches are called 3D face recognition. 3D face recognition have yielded more promising results in extreme conditions for 2D face recognition. However, a challenging aspect of 3D face recognition is how to synthesize a face models in other to extract its local features. In the case of face occlusion, the discriminative information relating the occluded part must be identified for an effective recovery the face model for recognition. Face occlusion occurs, when a face model is partly covered, for example, with the wearing of glasses, scarf, and long hairs. Since, only local part of a face is affected with face occlusion, identifying the occluded part would require a method of extracting the local features on a face model. With the analysis of these local features, we can identify the discriminative information of occlusion on the face. This information, can then be used as a basis to restore the occluded face model for effective face recognition.

In this research work, we focus on the approach to synthesize 3D face models in order to learn local facial feature representation. We propose an approach to learn this feature representation from real-world 3D face data of unodered point-set. This point-set data do not assume a particular order and so intuitive features can be leaned easily. With a localized facial features representation, information of the occluded part on the face can be extracted. This information will allow an intuitive modification of the local structures of occluded part on the face without affecting the unrelated part on the face model. Hence, a more accurate face match can be predicted. Our contributions are twofold. First, we propose an approach to use a neural network architecture which takes an unordered point-set of 3D faces as input and encodes intuitive feature representation of the point set. Then, we introduce a layer to the network which specifically learns the local pattern to identify discriminate features of occlusion on the face. We evaluate the network losses using the Chamfer and Earth Mover's distance loss estimations. Extensive experiments and analyses are conducted for validating the proposed approach, showing its effectiveness with respect to the state-of-the-art.

Contents

List of Figures

List of Tables

Acknowledgments

This work was funded by the National Research Fund (FNR), Luxembourg, under the project reference CPPP17/IS/11643091/IDform/Aouada.

I will also like extend my appreciations to Dr. Djamila Aouada, Dr. Anis Kacem for the privilege and research opportunity with the Computer Vision, Imaging and Machine Intelligence (CVI2) research group at SnT-Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg. It has been a great honour to be supervised by you. My expertise in Computer Vision research has been remarkably impacted under your supervision despite the work difficulties that came as a result of the COVID-19 pandemic.

I express my sincere gratitude my course coordinator in the person of Lew-Fock-Chong Lew-Yan-Voon for the opportunity of choosing the research topic "Recovering Occluded 3D Face Model for 3D face Recognition" for my masters thesis. Also to to my course professors for their tutorship and the academic influence on my pursuit for a degree in Computer Vision. More importantly, my course director professor Fofi for the opportunity of a masters degree in Computer Vision at the University of Bourgogne, France.

Many thanks to colleagues, friends and course-mates at Centre Universitaire Condorcet – Le Creusot in the like of Inder Pal, Deogratias Lukamba and Nahid Nazifi for the trust and morale towards me.

Most importantly, to my parents and siblings, thank you for your support that has kept me going through my career pursuits.

Chapter 1

Introduction

Face recognition is an active research area in computer vision that focuses on developing algorithms for analyzing unique patterns of facial features from images or face scans for identification and verification of a person. Over time, it gained popularity in a wide area of applications such as security systems, for access control both on mobile platforms and in industrial applications, in law enforcement agencies and many more.

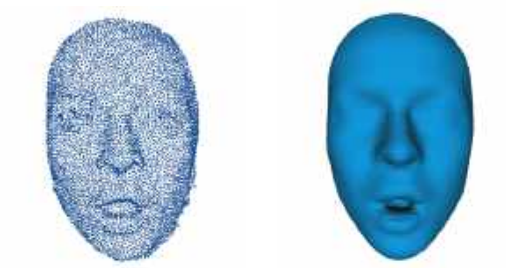
One of the main challenges in face recognition is the occlusion of faces. Occlusions usually occur when the face is partly covered with hair, wearing of glasses and scarfs. As a result, analyzing unique features on the face for recognition becomes challenging.

In the conventional 2D face recognition approaches, the geometric information on the face is described with the reflection of light on the face which is subject to the lighting conditions. Unlike these approaches, the geometric information contained in 3D facial data could substantially improve recognition accuracy under difficult conditions like in the case of face occlusion [32]. 3D face data are usually acquired with stereovision systems or compact 3D sensors. The common representation of 3D models are in point cloud data or mesh data representation, see Figure 1.1.

The focus of this research work is directed towards restoring occluded faces using 3D face model for effective face recognition. In order to achieve this, we must develop an approach

to synthesize the 3D face model in order to identify local patterns of occlusion on the face. These local pattern will allow us to compute an method of recovering the occluded face without affecting the visible part of the face. In the following section, we discuss some challenges that are related with the synthesis of 3D face models.

1.1 Challenges



point cloud mesh

Figure 1.1: 3D representation of face model:
point cloud data are represented with 3D point coordinate in space while mesh data are represented with both point coordinate as vertices and an encoded interaction of the coordinate to form a surface

The processing of 3D mesh data requires complex geometric computations for synthesis and analysis. In many cases, 3D meshes require a low dimensional parameterization with conventional Principal Component Analysis (PCA) [17] to extract meaningful features from the 3D models. The representation with PCA is limited to a global feature of the 3D model. This results to an inaccurate reconstruction of the model in the case of deformations, as irrelevant part of the deformed object are also affected. Some recent approaches have benefited from Sparse Localized Deformation Component (SPLOCS) [18], [13], [27] in which intuitive control of sparsely localized deformable features is possible. Both methods are subject to a linear representation for the deformable features and will fail in the case of non-linear or large scale deformation similarly to the case of partial occlusion on a face model. To address this problem, [1] proposed an effective graph-based neural auto-encoder. [20] proposed a different approach of mesh-based auto-encoder by introducing sparse regularization to the framework of SPLOCS for

a localized deformation representation. The limitation of these methods is that, they require a point-to-point correspondence for the alignment of mesh models. In order to restore face occlusion from 3D models of real-world data, we must be able to provide an alignment for the face model to effectively identify and reconstruct the occluded part on the face.

1.2 Contributions

We investigate the process of generating point features from raw sensor data of 3D face scans. These sensor data consists of a point cloud without any assumptions on its order. It is, therefore referred to as unordered point-set. Unordered point-set allows easy manipulation of its point data when it comes to geometric transformation, without affecting the neighbouring points in contrast to 3D mesh data. As a result, we can intuitively learn the point features from unordered point-set in other to identify occluded parts on a faces model.

We take advantage of the idea of PointNet [11] auto-encoder neural network architecture to develop a 3D face auto-encoder network, which embeds point features from unordered point-set into a feature space. This point feature embedding describes the combinatorial interaction in the point-set, and it is invariant to the order of the input point-set. As a result of this, we are able to overcome the problem of point correspondence when aligning 3D face models. Moreover we introduce a method to provide local support to the neural network architecture of PointNet auto-encoder by mapping the point feature embedding into a higher dimensional feature space. With this higher dimensional feature space, the network can learn the local patterns on face models. Through this network setup, we are able to identify the local structure of discriminate features of occlusion on a face model. On this basis, the local structure of the occluded parts, can be used to effectively recover the face model. Our contributions are briefly summarized below.

1. Extend PointNet auto-encoder architecture to a 3D face auto-encoder network that can directly process unordered point-set data of 3D face to overcome the problem non-linearity in feature representation.

2. Introduce a method that transforms the low dimensional point feature embedding from the 3D face auto-encoder network into a high dimensional embedding to learn local patterns of occlusion on a face model.

3. Effective reconstruction of an occluded face without modifying the structure of unrelated parts is possible.

4. Experimentally evaluate our model with real-world 3D face data with different forms of face occlusion to show that our method has the potential to solve the problem of face occlusion in ideal scenarios.

1.3 Outline

In this chapter, we introduce the concept, challenges and our contributions to the recovery of occluded face model for 3D face recognition. In chapter 2, we investigate some related work to our proposed approaches. We also discuss the specifics of our dataset that makes it a good choice for the proposed approaches. In chapter 3, we discuss our method of synthesizing 3D face data from unordered point-set, with the PointNet autoencoder [11] network setup. This approach, allows us to learn intuitive features from unordered point-set data of 3D faces and provides an easy manipulation of the point data. In chapter 4, we discuss our method of learning local patterns on face models from the low-dimensional feature representation of PointNet autoencoder [11]. Here, we introduce an operation that serves as a local support to the network. In chapter 5, we present the different experiment to evaluate our methods quantitatively and qualitatively. In chapter 6, we review this thesis work and proposed some solutions to improve the performance of our models. We also, introduce the next phase of this project, to reconstruct occluded face models with the basis of the analysis of the local pattern on the faces.

Chapter 2

Background

This chapter aims to provide an overview of the concepts underlined in this thesis. The first sections introduce the topic by reviewing relevant research work while the following sections introduce other important information regarding 3D face data. A more detailed description of our method on 3D face recognition is discussed in the next chapter.

2.1 Related work

Research work that has significantly contributed to the synthesis and geometric computation of 3D models and face recognition neural network architecture is introduced in this section.

2.1.1 3D face occlusion recovery

The earlier attempts for the recovery of the occluded facial region aim at using visual patterns that are either masked manually or with additional detection algorithms to segment the occluded region on a face. Recently, the generative models have been widely used for shape completion [14], [31] which in contrast, does not need any preprocessing on occluded region. This has yielded more promising results for the recovery of face occlusion. With the development of 3D Morphable Model (3DMM), some research work [29], [3] proposes an approach to incorporate a

3DMM as appearance prior in a Random Sample Consensus (RANSAC)-like algorithm for an occlusion-aware face model. All these approaches are based on the 2D image representations of the face model which are limited to facial texture and illumination on the face.

2.1.2 Synthesis of 3D objects

The earlier work on shape synthesis like [17] employs Principal Component Analysis (PCA) to synthesize mesh data to extract feature components. The feature components of PCA is global in its representation, which is not intuitive for 3D shape reconstruction. Sparse regularization is effective in localizing deformation features [9]. However, standard sparse PCA [34] does not take spatial constraints into account, therefore, the extracted deformation components do not aggregate in local spatial domains. By incorporating spatial constraints, a sparsity term is employed to extract localized deformation components [18], [2] which performs better than region-based PCA (clustered PCA) [26] in terms of extracting meaningful localized deformation components. The above methods are linear representations for local deformation features and cannot represent 3D shapes with non-linear deformations.

2.1.3 Neural networks

3D models have multiple representations. The work in [6] treats 3D shapes as voxels and extends 2D Convolutional Neural Networks (CNN) to 3D-CNNs for object recognition. For 3D shape synthesis, [28] used deep belief networks to generate voxelized 3D shapes. [24] proposed to combine ResNet and geometry images to synthesize 3D models. Both [16] and [4] proposed to use neural networks for encoding and synthesizing 3D shapes based on pre-segmented data. All the methods of synthesizing 3D models are restricted by their representations or the adopted primitives and do not capture full information of the raw data. Recently, we see a surge of interest in designing deep-learning architectures suited for point cloud data. These methods have demonstrated remarkable performance with the synthesis of 3D object for classification and segmentation. The challenge with these methods, is to determining a sufficient operation to

transform unordered point-sets to be invariant of their input order. The work of PointNet [21] use deep learning network to process unordered point cloud data. This network uses a symmetric function to aggregate point-wise features that are invariant to the input order. PointNet++ [5] hierarchical applied PointNet recursively to obtain both global and local features on point cloud.

2.1.4 3D face recognition

The most popular 3D method depends on computing local and global feature descriptors. For instance [23] matched the 3D Euclidean and geodesic distances between pairs of 25 anthropometric fiducial landmarks to perform 3D face recognition. [25] represented a 3D face with multiple mesh-DOG key-points and local geometric histogram descriptors while [12] represented the facial surface by radial curves emanating from the nose-tip. Alternatively, some other works found reasonable mappings from the 3D space to canonical 2D domains. [7] proposed frontalized 3D scan to generate 2.5D depth map and extract the depth map's features by using a VGG16 network [33] to represent the 3D face. [10] proposed a method that projects face point cloud into depth, azimuth and elevation maps to generate a three channel image. These methods reduces a 3D face representation to 2D images, to extract facial feature using 2D CNNs.

2.2 3D face data

In this section, we discuss some methods of data acquisition and available researches on 3D face recognition databases, with their peculiarity to the different approaches of face recognition. Following, we highlight the importance of our choice for 3D face dataset.

2.2.1 3D face acquisition

The acquisition of 3D face samples could be categorized, as active acquisition systems and passive acquisition systems [32]. Passive acquisition systems consists of several cameras placed apart from each other, where points observed from the different camera, are matched to extract

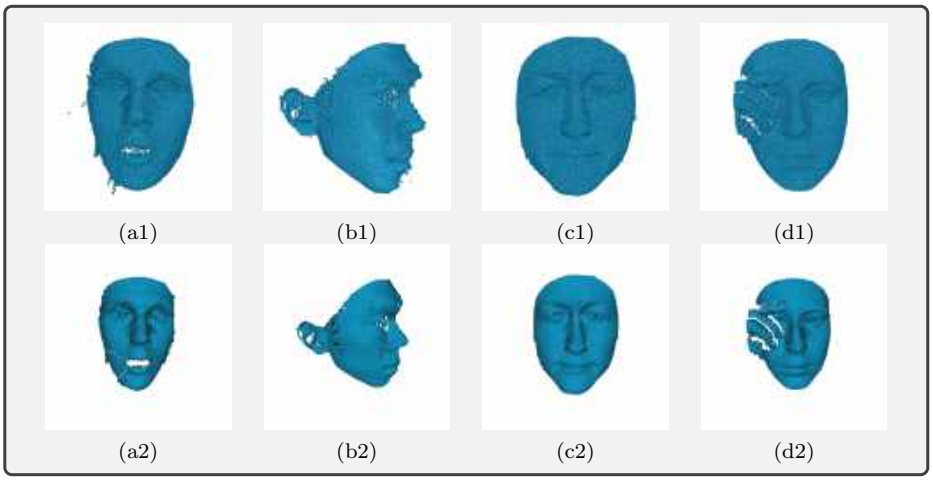


Figure 2.1: Samples from the Bosphorus 3D face dataset:
The first row shows point cloud data with a sample of (a1) expression-based variation, (b1) head-pose based variation, (c1) neutral face and (d1) occluded face variation while the second row shows a re-meshed face from the respective samples in the first row.

their 3D location. This method of acquisition is also called stereo-imaging. On the other hand, the active acquisition system uses non-visible light to illuminate the face and measure the reflection to determine the shape features of the target face.

2.2.2 3D face database

One of the main factors that have contributed to the level of success of 2D face recognition approaches is the availability of a huge amount of training data. For instance, FaceNet [8] was trained on a private database containing 200M labeled images of 8M identities while VGG-Face [19] used 2.6M faces of 2622 identities. Unlike the 2D face recognition databases, there is a limited number of publicly available sources of 3D face data. Also the types of sensors used as well as the techniques adopted in acquiring a 3D face data usually impact the 3D face recognition system [33]. Some examples of widely known 3D face recognition database are given in tab.2.1.

From the list of the available researches on 3D face recognition database known to us, we chose the Bosphorus 3D faces dataset for the development of the proposed approach. This

Database	Data format	Texture	Number of subject	Number of samples	Expression Specific	Pose specific	Scanner
ZJU-3DFED	Mesh	Yes	40	360	Yes	-	-
FSU	Mesh	No	37	222	Yes	-	Minolta Vivid 700
GavabDB	Mesh	No	61	540	Yes	Yes	Minolta Vi-700
FRAV3D	Mesh	Yes	105	-	Yes	Yes	Minolta Vivid 700
BU-3DFE	Mesh	Yes	100	2500	Yes	-	Stereo photography, 3DMD digitizer
Beckman	Mesh	Yes	475	-			CyberWare scanner
UoY	Mesh	Yes	350	5000	Yes	Yes	Stereo vision 3D camera
FRGC v2.0	Range image	Yes	466	4007	Yes	-	Minolta Vivid 3D scanner
UND	Range image	Yes	277	953	-	-	Minolta Vivid 900 range scanner
ND2006	Range image	Yes	888	13450	Yes	-	Minolta Vivid 910 range scanner
SHREC08	Range image	No	130	780	-	-	-
3D-TEC	Range image	Yes	214	428	Yes	-	Minolta scanner
SHREC11	Range image	No	130	780	Yes	-	Escan laser scanner
UMB-DB	Range image	Yes	143	1473	Yes	-	Minolta Vivid 900 laser scanner
Texas 3DFRD	Range image	Yes	118	1140	Yes	-	MU-2 stereo imaging system
Bosphorus	Pointcloud	Yes	105	4666	Yes	Yes	The Inspeck Mega Capturor II 3D scanner
BJUT-3D	Mesh	Yes	500	-		-	CyberWare 3030RGB/PS laser scanner

Table 2.1: 3D face recognition database
The table shows the different database that has been published specifically for 3D face recognition which is either adapted for the pose specific, occlusion specific or expression-based face recognition algorithms. The tables show the comparison between the number of object samples, the type of 3D data format, and type of scanning device used to capture the object sample of the referenced database.

database consists of 3D face scans captured with the Inspect Mega Capturor II 3D scanner. It includes a rich set of expressions, systematic variation of poses and different forms of face occlusions Figure 2.1. This database is suitable for our application in three ways:

1. Face data are stored in point cloud format (raw data of 3D face scans)

2. Different forms of face occlusions are included in this database

3. A rich set of head pose variations are available;

2.3 Discussion

Based on our reviews from the research work in Section 2.1, our idea of implementing a method for recovering occluded face is expressed. Also, we discussed available sources of research on 3D face recognition database and the reason for our choice of dataset. In the next chapter, we will discuss in more detail our proposed approach.

Chapter 3

3D face data reconstruction

The generation of point features from 3D face models is an important aspect of our methodologies. To achieve this, first, we preprocessed the 3D face dataset that we have obtained, see Section 2.2.2, with a suitable sampling algorithm, discussed in the first section. The preprocessed data, is then passed into the PointNet autoencoder network to learn models for point features of 3D faces, discussed in the second section. Through the training of these models, two different methods of loss function was adopted to analyze the method that gives a better representation of 3D faces. These methods of loss function are discussed in the last section of this chapter. With these models, we learn point feature in a low-dimensional feature representation and reconstruct a 3D face from this feature representation. This feature representation is important for our consequent computation which includes learning of local patterns on the face model.

3.1 Data preprocessing

To recover face occlusion from 3D face model, firstly, we need to compute an algorithm to identify the local occluded region on the face model. To achieve this, we must be able to compute an alignment between face models such that for every point on a face model, we

must be able to determine its corresponding point on a different face model. The conventional approach for synthesizing mesh data does not provide a straightforward solution for aligning mesh data. As a result, we consider an approach, to intuitively learn point features that encode order invariance from unorder point-set. Then, we are able to overcome the problem of point-to-point correspondence when aligning face models.

We investigate the PointNet auto-encoder architecture to develop a 3D face auto-encoder network. This 3D face auto-encoder network takes unordered point-set of 3D faces as input and embeds the point features which are invariant to the order of the input points.

3.1.1 Unordered point-set

In many cases, 3D face data are acquired from laser scan sensors in which its data are stored as point cloud data represented as an x, y, and z geometric coordinates on an underlying surface of the face. These point-sets are said to be unordered because they do not assume a particular order or combinatorial connectivity patterns unlike mesh models. This makes it easy to manipulate or transform the individual point data without affecting the geometry of neighbouring points in the set.

We obtained 3D face samples in point cloud data format from the Bosphorus 3D face database, see Section 2.2.2. This dataset consists of 3D face scans acquired from a laser sensor. Each sample of the dataset has a varying resolution which depends on the face shape and size.

Since most of standard neural network architectures require a fixed resolution of 3D point cloud data as input, we must compute an efficient algorithm that can unify the varying resolution of the face samples in the dataset. Our algorithm must be able to sample the data up to a fixed resolution while preserving relevant information represented on a 3D face model. Also, down-sampling the data would help to reduce the processing power required for face data of high resolution. Our proposed method of sampling point cloud data of face models is discussed in the next subsection.

3.1.2 Point sampling for 3D face

GT RS FPS

Figure 3.1: point-set Sampling:
A sampled 3D face data to a resolution of 10,000 points. GT: Ground truth, RS:Random Sampling, FPS:Farthest Point Sampling

A 3D face scan has only one surface where all the points lie on. The 3D geometric features of a face are well described by the contours of the nose, mouth and eyes. Hence, reliable operation for point sampling should well respect the curvature of the subset points of these contour regions on a face model. While the conventional sampling methods for point clouds data are dependent on sampling ratio or the ring size of points to be sampled, the Random Point Sampling (RPS) and the Farthest Point Sampling (FPS) method provides a straight forward approach to sample point cloud data to a fixed number of point-set. As a result, these sampling methods are more suitable for pre-procecing the 3D face dataset. In practice, FPS performs better than random sampling see Figure 3.1.

The FPS algorithm measures the Euclidean distance relationship between centroids of subset points. This method of distance measurement is however, not preferable for face model because the curvature of facial features are not well represented. A proposed solution by [33] is to parameterize the Euclidean distance estimation of FPS with a curvature sensitive parameter.

Given an input point-set $N = \{x_1, x_2, \ldots, x_n\}$, the corresponding sampled point-set $M =$

$\{\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_m\}$ is given by,

$$\operatorname{argmax} \sum_{k=1}^{i-1} d\left(\hat{x}_i, \hat{x}_k\right), \tag{3.1}$$

where $d\left(\hat{x}_i, \hat{x}_k\right)$ is a Euclidean distance between $\hat{x}_i$ the current point and $\hat{x}_k$ the remaining point to be searched in the subset of N. Hence, the curvature sensitive sampling is given by,

$$d_c = d\left(\hat{x}_i, \hat{x}_k\right) \cdot C_{x_i}^{\lambda}, \tag{3.2}$$

where C_{x_i} represents the curvature of the point x_i and λ is used to parameterize the curvature sensitivity. The log form of the curvature sensitive sampling can also be expressed as,

$$\log\left(d_c\right) = \log\left(d\left(\hat{x}_i, \hat{x}_k\right)\right) + \lambda C_{x_i}. \tag{3.3}$$

An increase in the value of λ will increase the chances of some points to be selected, as they tend to gain more distance compared to the ordinary FPS algorithm.

3.2 3D face auto-encoder

In order to process unordered point-sets, we must be able to learn certain feature represented in the categories of these point-sets. For a given point-set of a face model, each point is represented as a metric distance in space. However, these point data in the point-set are not isolated from each other. Therefore, we will need additional information about the interaction between neighbouring points in order to describe it as a face model. As a result, we must learn the interaction between points in a point-set to process them.

For our method of recovering face occlusion, we require a point feature generation for 3D face models that can capture the local structure of points interaction within a point-set, which must be invariant to the order of the point-set, and also invariant to the geometric transformation of individual point in a set.

To develop a neural network that can take unordered point-set as input and encode a feature

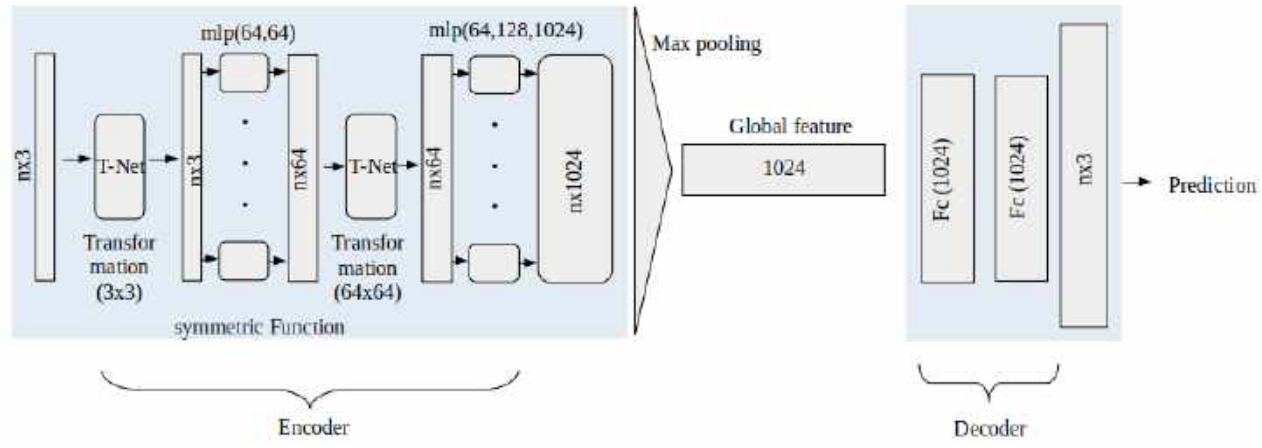


Figure 3.2: PointNet Auto-encoder:
encodes point features from unordered point-set using a symmetric function and aggregate the point feature into feature space embedding with a max pooling function. The Decoder part of the network then predicts the point-set from the feature space with a set of fully-connected layers.

representation that will meet the criterion mentioned above, we carefully study the PointNet [21] network architecture which gives a reasonable solution for our approach for the synthesis of 3D shape from the unordered point-set.

The PointNet [21] network which was optimally implemented to learn features of unordered point-set for 3D classification and segmentation was adapted to create a 3D face auto-encoder network Figure 3.2 to learn point features from the unordered point-set of 3D faces. This network computes a symmetric function that consists of feature transformation layers approximated by multi-layer perceptrons, and a max-pool layer that aggregates the information from all points into a feature space embedding, given as,

$$f\left(\{x_1, \ldots, x_n\}\right) \approx g\left(h\left(x_1\right), \ldots, h\left(x_n\right)\right), \tag{3.4}$$

where $f : 2^{\mathbb{R}^N} \to \mathbb{R}, h : \mathbb{R}^N \to \mathbb{R}^K$ and $g : \underbrace{\mathbb{R}^K \times \cdots \times \mathbb{R}^K}_{n} \to \mathbb{R}$ is the symmetric function. Here h is approximated with a multi-layer perceptron network and g by a composition of a single variable function and a max-pooling function. N is the dimension of the input point-set, n is the number of points and K is the dimension point features. Hence the point feature space is

given as $[f_1, \ldots, f_K]$.

After point feature embedding, the point-set of the input face model is reconstructed back using a set of fully-connected layers forming a decoder. The decoder transforms the point features into a point-set in 3D space with a number of points equal to that of the input point-set. The optimization of the whole network is performed through minimizing two loss functions that will be discussed in the next section.

3.3 Loss estimation

In order to optimize an auto-encoder neural network, it is common to include a loss function at the end of the decoder that compares the prediction of the network with its input. Given a set of input point-set $\{N\}$ along with their corresponding predicted reconstructions $\{\hat{N}\}$, the loss function is defined as,

$$L\left(\{\hat{N}\}, \{N\}\right) = \sum d\left(N, \hat{N}\right), \tag{3.5}$$

where $d\left(N, \hat{N}\right)$ is a distance to be estimated. Since we are dealing with unordered point-sets, a simple mean squared error could not be a reasonable choice for the distance d. To compute this distance, [11] introduced the Chamfer Distance (CD) and the Earth Mover's Distance (EMD). These methods which have different properties for capturing shape space [11] were used separately to train different models. While it is important to consider a function of distance estimation that is robust to unordered point-set, the CD and EMD provide a better solution to some of the problems associated with unordered point-set which includes the correspondences between the point-set.

Chamfer distance

To estimate the distance between predicted point-set and its ground truth, the CD algorithm implements the nearest neighbour search for each point in the other set. It is, however, viewed as a function of point locations in $\hat{N}$ and N. This distance is computed in both directions and

it is given as,

$$d_{CD}\left(\hat{N}, N\right) = \sum_{y \in \hat{N}} \min_{x \in N} \|y - x\|_2^2 + \sum_{x \in N} \min_{y \in \hat{N}} \|y - x\|_2^2. \tag{3.6}$$

The CD loss function does not require that the number of input point-sets has to be the same and it produces reasonable high-quality results in practice [11].

Earth Mover's distance

The Earth Mover's Distance which is a solution for a transportation problem is used to solve an optimization problem. It computes a bijection between the two point-sets given as,

$$d_{EMD}\left(\hat{N}, N\right) = \min_{\phi : \hat{N} \to N} \sum_{y \in \hat{N}} \|y - \phi(y)\|_2 \tag{3.7}$$

where ϕ is a bijection between $\hat{N}$ and N. Unlike the CD, the EMD always requires that the input point-sets to have the same number of points. However, one major drawback of EMD is that it is both memory and computationally intensive, hence is usually approximated [11].

3.4 Discussion

We implemented a network for the generation of point features for 3D face models from unordered point-sets. These features are invariant to the order and geometric transformations of its points. With our model of point feature generation, we can encode the input 3D point-sets of face models into low-dimensional representations and decode them to reconstruct back the input 3D point-sets without caring about the order of the points. While the encoded version of an input 3D point-set provides a compact and summarized version of it, it does not preserve the local patterns of the input. Since we are interested in localizing and recovering occlusions, we have introduced a new operation that allows the preserving of local patterns in the encoded versions of the point-sets. This operation involves the modification of our initial network setup by introducing a new layer called local support layer. The details of the implementation of the

local support layer will be discussed in the following chapter.

Chapter 4

3D face auto-encoder with local support

In this chapter, we introduce our approach of preserving local patters within the introduced PointNet auto-encoder in the previous chapter. This approach involves learning local patterns of faces occlusion from our initial model of point features generation for 3D faces. Firstly, we will discuss our new network setup, which consists of a new layer that transforms the feature space to a high dimension embedding that corresponds to point features of the input point-set. In the following section, we then introduce a method of regularizing the transformation function to learn the local patterns that will make the identification of the occluded part of the input face model easier and more robust.

4.1 Feature transformation

Using the auto-encoder introduced in the last chapter, we were able to encode an input 3D point-set into a low-dimensional representation using the encoder and reconstruct it afterwards using the decoder. In the context of our project, the main intuition behind using an auto-encoder is

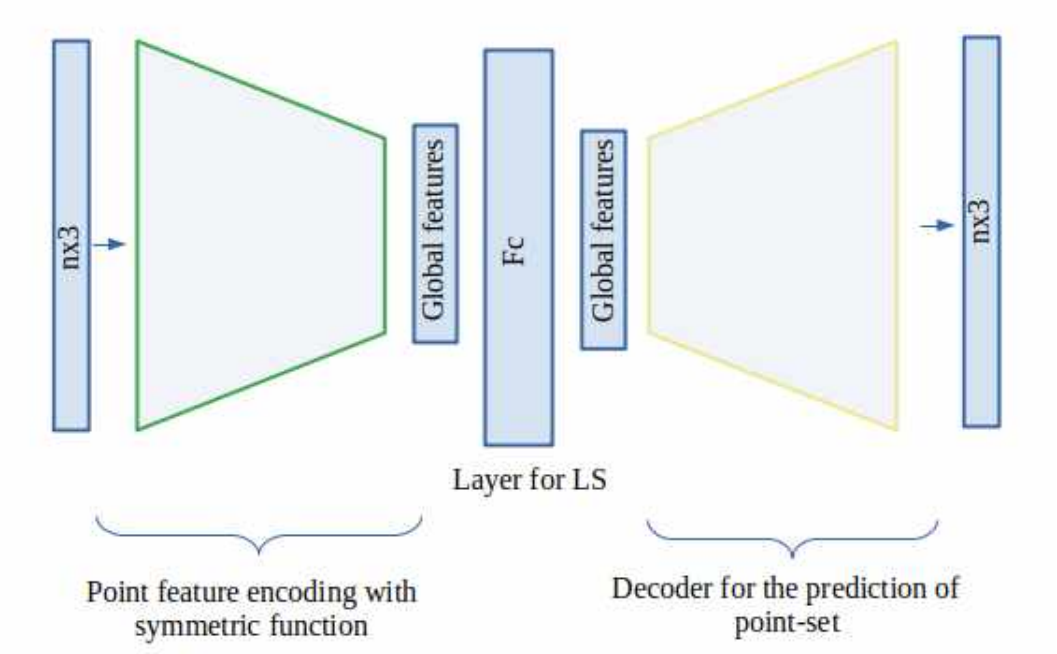


Figure 4.1: 3D face auto-encoder with local support
The network is developed from the previous point feature generation network. It includes the transformation of feature space into a high dimensional feature representation by a fully-connected layer.

to get a low-dimensional representation that allows easy manipulation of the point-set such as removing occlusions. However, the encoded representation obtained using PointNet describes the global shape of the input point-set and does not preserve its local patterns. To overcome this problem, we propose including a local support to the PointNet auto-encoder with the aim of making it able to preserve local patterns. Preserving local patterns is very important in our occlusion recovery task since occlusions usually occur in local parts of the face.

To learn local patterns, we first introduce a fully-connected layer without bias before the decoder part of our network of point feature generation for 3D faces, see Figure 4.1. This layer serves as a function to transform the feature encoding from the point feature generation network into a new high-dimensional feature space whose dimension is as the input point-set. In the new feature space, each component corresponds to a local pattern to be learned from the low dimensional encoding in the network.

In order to achieve local support with the introduced fully-connected layer, we regularize the transformation function with a parameter computed from the surface distance of a template face model Section 4.2. This operation helps us to learn the local patterns on face models. Hence, the difference in the values of the high dimensional feature space of the input point-set can be seen as the occluded part on a face model.

The transformation function of the point feature encoding is given as,

$$z = Cf, \tag{4.1}$$

where $C \in \mathbb{R}^{K \times \mu}$ is the function that transforms the low-dimensional feature representation $f \in \mathbb{R}^{\mu}$ to a high-dimensional feature representation $z \in \mathbb{R}^{K}$. μ and K are the dimensions of low-dimensional feature space and high-dimensional feature space respectively.

To decode the feature representation, we simply reverse the operation as,

$$\widehat{f} = C^{\mathrm{T}}. \tag{4.2}$$

4.2 Regularizing local support layer

In order to make the introduced fully-connected layer able to preserve local patterns, we introduced a new loss function that regularises the transformed feature representation of our local support layer. This is inspired by the research work on sparse localized deformation components [18]. Since the introduced fully-connected layer is parameterized by C, the regularization is applied on this matrix as follows,

$$\Omega(C) = \frac{1}{K} \sum_{k=1}^{K} \sum_{i=1}^{n} \Lambda_{ik} \left\| C_k^i \right\|_2, \tag{4.3}$$

where n is the number of points in the input point-set N and C_k^i is the μ-dimensional vector associated with component k of point i in given point-set N, Λ_k^i. This parameter is computed

from the normalized geodesic distance of a template face model which has a neutral expression and no occlusion. It is, therefore given as:

$$\Lambda_{ik} = \begin{cases} 0 & d_{ik} < d_{\min} \\ 1 & d_{ik} > d_{\max} \\ \frac{d_{ik} - d_{\min}}{d_{\max} - d_{\min}} & \text{otherwise,} \end{cases} \tag{4.4}$$

where d_{ik} is the geodesic distance from point i to the other points k in the point-set of the template face model. This method of surface distance estimation has been proposed by [15]. The regularization parameter simply maps this distance values linearly to the transformation function C. Thus, changing the regularization strength of each component locally. The layer for local support in the network is iteratively updated during optimization by recomputing the distances at every step for each component.

4.3 Discussion

We implemented a method that takes advantage of the network of point features generated from unordered point-sets of 3D faces by introducing a fully-connected layer before the decoder part of the network. This new layer serves as local support to compute an operation to locally manipulate the component feature representing each point of the input model. This is done by applying a regularization term on the new layer using the geodesic distance of a template face model and adding it to the loss function. By doing so, we learn discriminate features that preserve the local structure of the input occluded faces, allowing an easy identification of occlusion on face models.

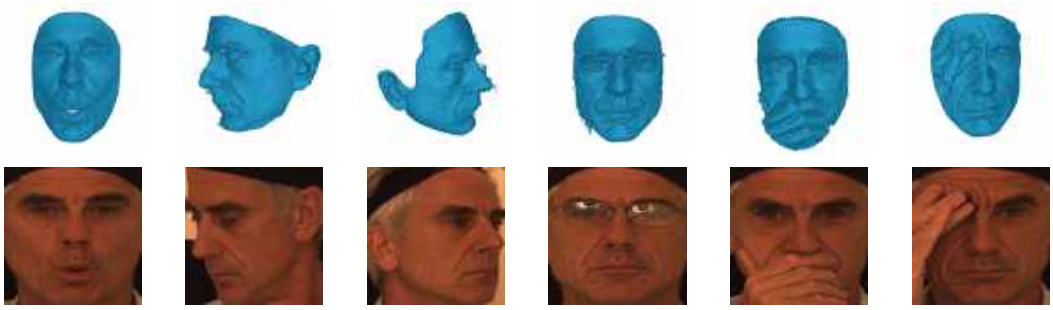
Chapter 5

Experimental results

In this chapter, we aim to report the outcomes of our methods. First, we describe the real-world dataset used for our experiments, then we describe the result from the proposed sampling methods. In the following section, we illustrate the results from our of implementations for point feature generation for 3D faces with and without local support. Also, we describe our proposed training strategies to improve the performance of our models. Lastly, we report both qualitative and quantitative results for all models.

5.1 Implementation

Our proposed method for synthesising 3D face model from unordered point-sets in order to identify occlusion on a face model was implemented using an autoencoder network setup. This can be divided into two main aspects, namely, the point features generation for 3D face model from unordered point-set and learning of local patterns from the encoded point features in the network to identify occlusion on a face model. The autoencoder network was implemented with the Tensorflow GPU framework and the models were trained with the Bosphorus 3D face dataset [22] and the ShapeNet dataset [30].



(a) (b) (c) (d) (e) (f)

Figure 5.1: Face samples from Bosphorus dataset:
the first row shows the point cloud data, while the second row shows the respective texture file for each samples. 'a' shows an expression variation, 'b' and 'c' shows a pose variation and 'd','e',and 'f' shows occlusion variation samples in the dataset.

5.1.1 Bosphorus 3D face dataset

The Bosphorus dataset [22] consists of 4666 real 3D face scan from 150 individuals. Each sample is stored in a raw point cloud data format, its respective texture file in a '.png' file format and handcrafted facial landmarks. We divided the 3D face samples available in the dataset into 80% training-set, 20% validation-set and 20% testing-set. All divisions of the 3D face samples in the dataset contains the different variations of faces represented in the dataset, which includes, pose, occlusion and expression variation. Each face sample in the dataset differs in resolution ranging from 22,500 to 93,292 points. We considered to pre-processing each sample to reduce its resolution to a number of point less than the minimum resolution available in the dataset. This helps to easy the computation power and also to meet the requirement of a 3D network that can only process a fixed resolution of 3D dataset.

5.1.2 Point cloud data sampling

The goal for sampling the 3D face data comes with the ability of the autoencoder network to take only a fixed resolution of point-set as input during training. To fix a resolution for the dataset, we must consider that important information on the faces are well preserved. We experimented the Farthest Point Sampling (FPS) and the Random Point Sampling (RPS) methods on the

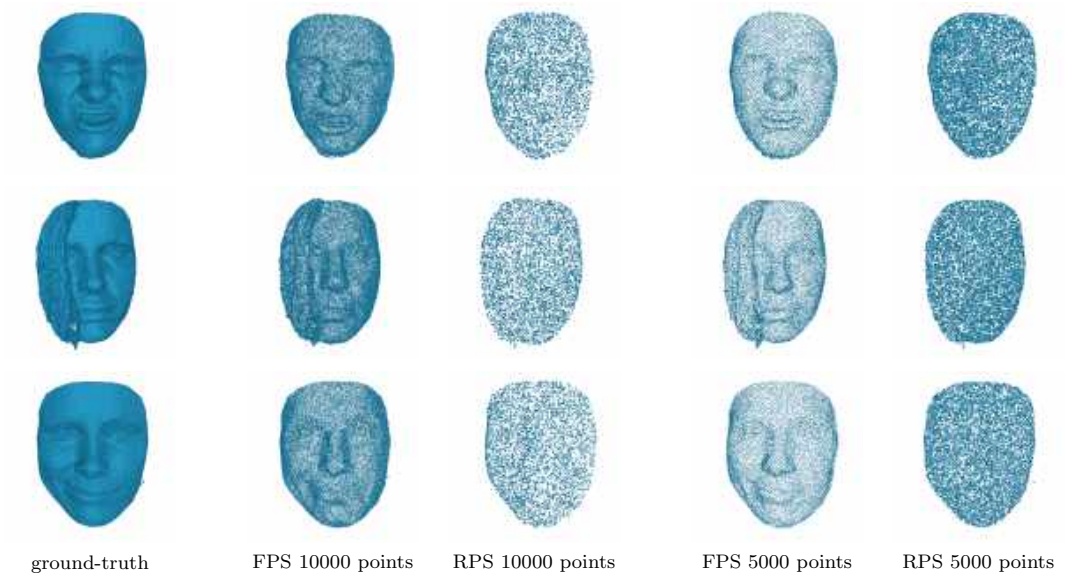


Figure 5.2: Sampling methods:
each row represent a sample of 3D face point-set. on the first column is the ground truth, the second and third column is 10000 point-set and on the last two columns are 5000 point-set for FPS and RPS respectively

3D face dataset to see how well these sampling methods represents the information on the face models. We evaluated the method with a resolution of 5,000 and 10,000 point-set see Figure 5.2, keeping in mind that the minimum resolution in the dataset is 22,500 point-set.

As expected, the performance of the sampling algorithm with 10,000 point-set was remarkably considerable, as relevant information on the face model is preserved with the FPS. In contrast, with 5,000 points-set, fine feature on expressive face fades out, however, it does not seem to be extremely undesirable especially when considering a trade-off with the computation power for processing large point-set. However, with RPS, the results are less satisfactory in both cases. Both resolutions of point-set for FPS were kept for further experiments on point feature generation for 3D faces.

5.1.3 ShapeNet dataset

This dataset consists of handcrafted point cloud data of objects of different categories such as chairs, tables, aeroplanes lamps etc. The dataset contains sixteen thousand models from about

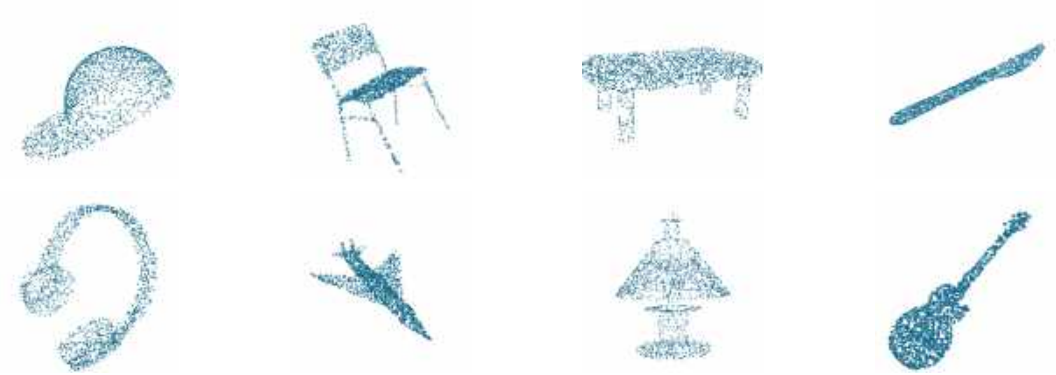


Figure 5.3: Objects from ShapeNet dataset:
samples of point-set of objects from different categories in this dataset

sixteen different categories with resolutions ranging from about 1,653 to 3,000 point-set. We used this dataset to experiment with the PointNet autoencoder network, and then we fine-tuned the learned network on the Bosphorus dataset for 3D face point features generation.

5.1.4 Training strategies

The initial training for the model of point features generation for 3D faces was systematical, through a few numbers of epochs to observe the behavior of the loss functions as well as the learning rate. The model was trained with 3,266 training samples and 700 evaluation samples. Each set of samples contains all the variations of 3D face models available in the Bosphorus dataset. Through the training and evaluation phase, we estimated the losses with **Chamfer distance** method of loss function Figure 5.5 in a model, and **Earth Mover's distance** method of loss function Figure 5.6 for a separate model. In each case, we trained a separate model for 5,000 points and 10,000 points to see the effect of the resolutions on the models. All training was done for 400 epochs, a decay step of 200,000 and the initial learning rate was set at 0.001, with a decay of 0.7 Figure 5.4.

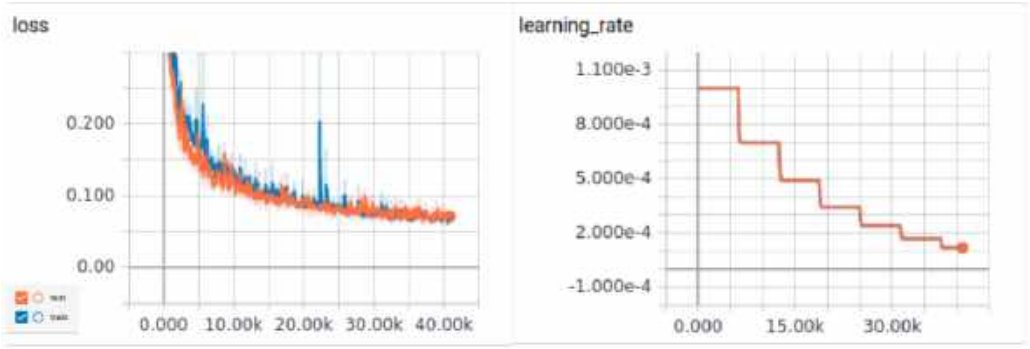


Figure 5.4: Model Scalar:
graph for losses and learning rate decay from the model point features generation for 3D faces.

Chamfer distance

The results of the prediction from the model with the Chamfer distance method of loss function Figure 5.5 shows more promising results with 10,000 point resolution. However, some important information about the expression on the faces are not well represented.

Earth Mover's distance

The performance from the model with the EMD method of loss function did not show a very good result with fine features on the faces, especially around the mouth and eyes. Also, on 'sample 3' of Figure 5.6, the traces of the fingers on the mouth of the subject could barely be noticed as compared to Figure 5.5. Despite these setbacks, it is good that in both scenarios of CD and EMD, the global shape of the faces, and the major facial features are preserved. Most importantly, the unique features of the face sample are also preserved.

Fine-Tuning Model of Point Feature Generation for Face Model:

The intuition here is that the ShapeNet dataset contains much more examples which help the model to better generalize on testing samples. This is called transfer learning. However, the

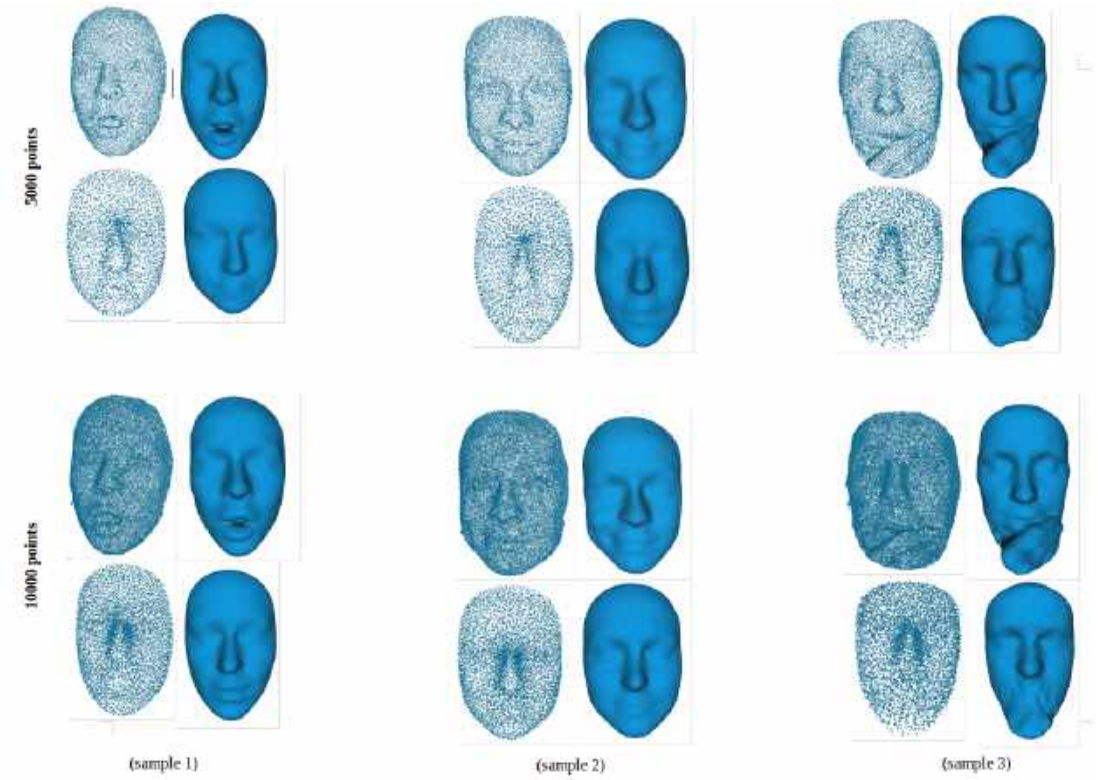


Figure 5.5: Predictions with CD loss:
for each subset grid, on the top-left is the input point set , its re-meshed data on the top-right, predicted point set on the bottom-left and its re-meshed data on the bottom-right

resolution of objects in the ShapeNet dataset is relatively small as compared with the Bosporus 3D face dataset. While important features of objects in the ShapeNet dataset can still be preserved with a resolution of about 2,500 points, it is challenging to adapt the model from this dataset to generate a model of 3D faces with 5,000 to 10,000 points.

To overcome this problem, we progressively increase the dimension of the last layer of the model from ShapeNet dataset to fit the resolution of the model of the 3D face. To achieve this, firstly, we restore all weights except the weight from the last fully-connected layer of the network which maps the embedding point features into the dimension of the input point-set. We then replace this layer with a new fully-connected layer of the same size as the resolution of the face data. The newly introduced layer was trained over a few numbers of epochs while the other

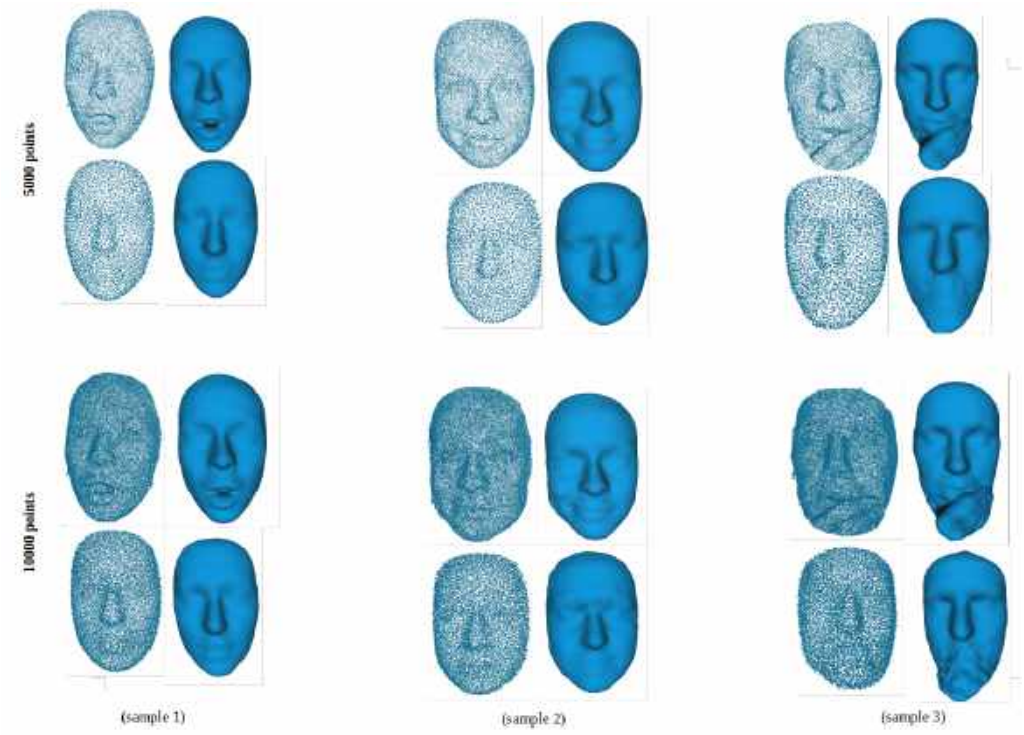


Figure 5.6: Predictions with EMD loss:
for each subset grid, on the top-left is the input point set , its re-meshed data on the top-right, predicted point set on the bottom-left and its re-meshed data on the bottom-right

layers in the network were frozen. Then, we continued the training with the remaining layers unfrozen, except the first two convolution layers. The fine-tuning network was systematically trained up to 500 epochs with the CD loss function, minimized with Adam's optimizer. The point resolution was fixed to 5,000 points and 10,000 points for different model.The initial learning rate was set at 0.001 with decay rate of 0.7 see Figure 5.7.

From the results of the fine-tuned model Figure 5.8, the global face shape, and the unique features of the of the subjects, are preserved on the predicted face model. However, it is to determine the expressions and other fine information around the mouth especially with 'sample 1' Figure 5.8.

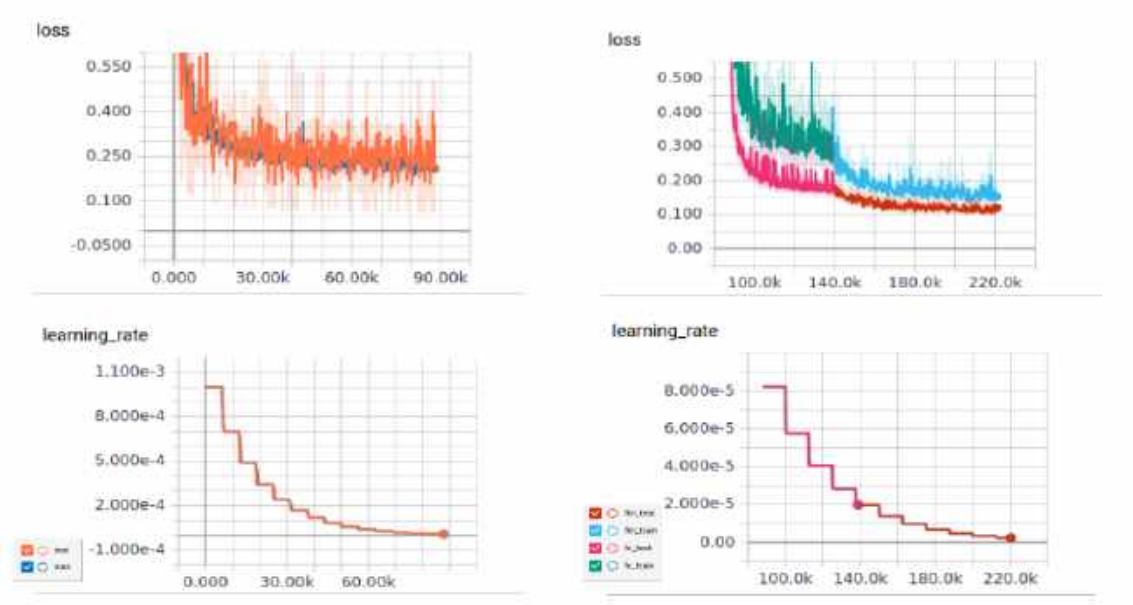


Figure 5.7: scalar from fine-tuned model:
on the first column shows scalar of pre-trained model with the ShapeNet dataset on the second column shows scalar for fine-tuned model for 3D face data, fc is the re-initialized last fully-connected layer, and ftn is the fine-tuned model.

5.1.5 Local support

The idea of introducing a local support is to identify the occluded part of a face model. We developed an autoencoder network for 3D faces with local support which basically is an addition of a fully-connected layer without bias to our network of point feature generation for 3D face model. This layer transforms the point feature encoding from a low dimensional space into a new latent encoding where each of its component is a representation of local patterns on a face model. With this representation, we learn the local patterns with respect to a template face model.

The new network was trained with the model that uses CD loss function, that is, the model with the best performance from our previous experiments see Table5.1. We restored the weights from this model to train the layer of the local support in order to generate a new model. During training process, the weights from the pre-trained model were kept unchanged while only the

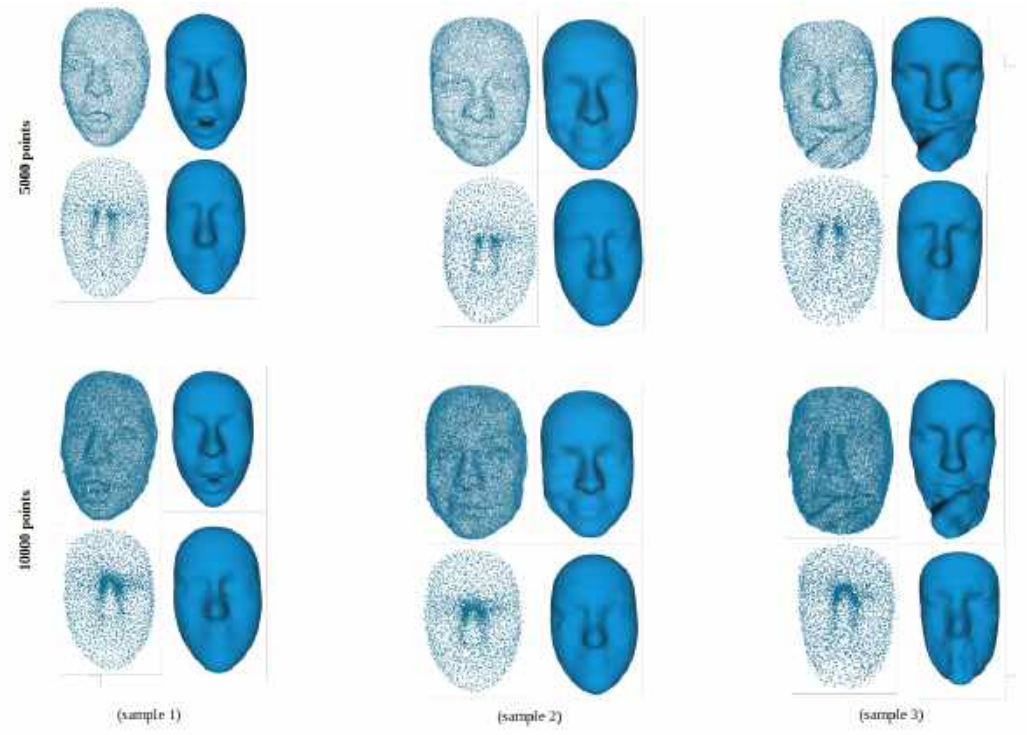


Figure 5.8: Prediction from the model:
for each subset grid, on the top-left is the input point set , its re-meshed on the top-right, predicted point set on the bottom-left and its re-meshed on the bottom-right

local support layer was trained progressively to about 400 epochs. With this model, we can determine discriminate feature on an occluded face with the changes in the values represented in the local support layer.

In order to observe the effect of the local support on the point-set prediction, we estimated the distance between a neutral face sample and an occluded face sample of the same identity, using the Chamfer distance algorithm see Section 3.3. In Figure 5.9, we report a visual representation of the distances on the predicted face. This distances were computed for the model with local support and without local support. From the results, we observed that the occluded part of the face was enhanced with the local support, in contrast to the model without local support. This effect makes the region of occlusion more visible for effective reconstruction of the occluded part.

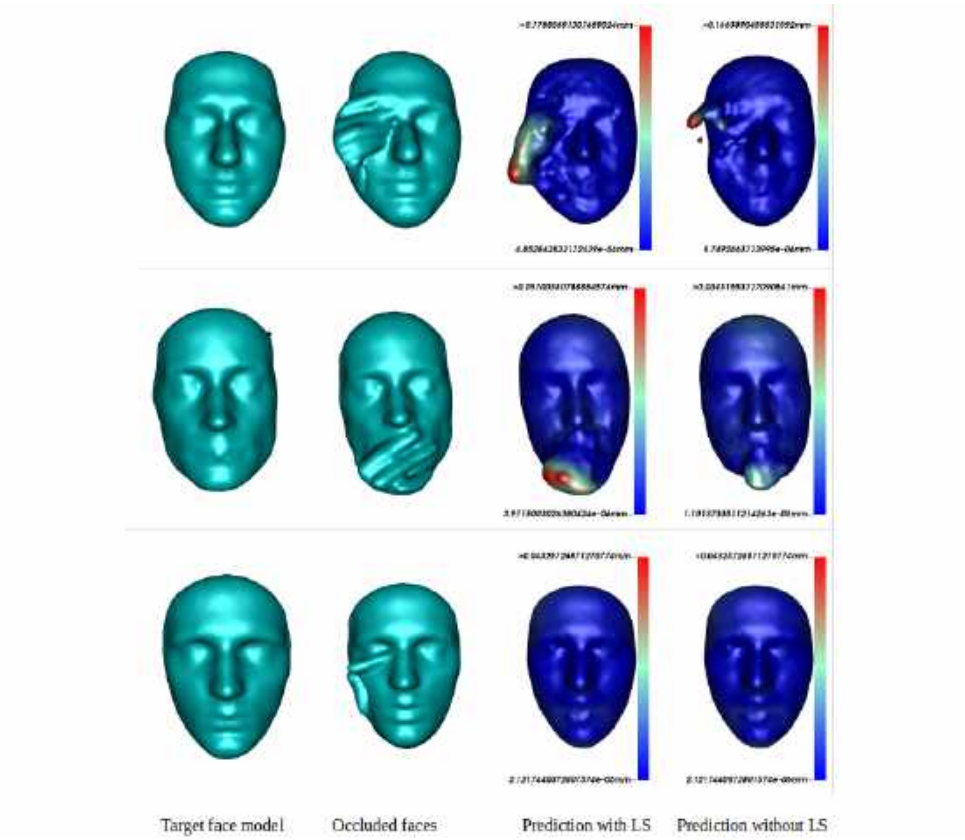


Figure 5.9: Prediction with local support
Each row is a sample of an occluded face model. On the first column is the target face model, the second is the input face with occlusion, the third is the prediction point set prediction with the local support and the Last column is the point set prediction from the model without local support.

Regularization parameters for local support

In our autoencoder network, the feature transformation function was constrained with a regularization parameter Λ to learn the local pattern of face models see Section 4.2. This parameter value has been estimated from the surface distance of a non-occluded template face model. In our case, we considered using the mean of the neutral faces in the dataset. However, it is however important to respect the curvature on the surface of a face model in order to estimate

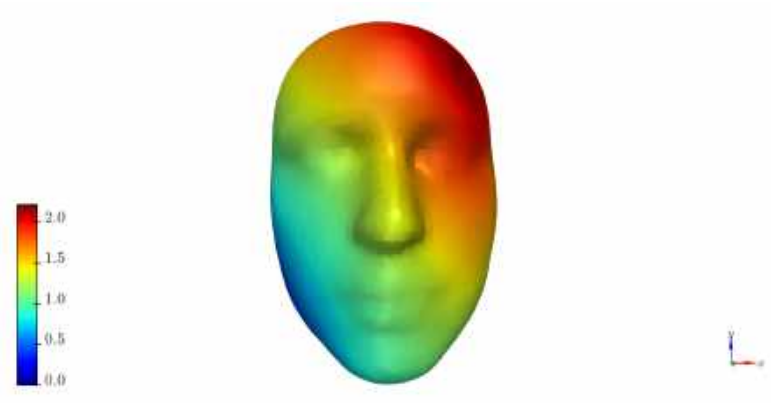


Figure 5.10: Geodesic distance computation
The color map shows the distance distribution from the center vertex to all vertices in the face model

this parameter. Since our approaches are entirely dependent on point cloud data, we therefore considered to adopt the method of surface distance estimation of a 3D object using heat flow [15] to capture as much information on the subset region on our template face model.

5.2 Evaluations

We perform both quantitative and qualitative analysis for our proposed method of point feature generation for 3D faces from unordered point-sets. We show the result from each model and we also report the mean error from all the models tab.5.1 over the testing samples to evaluate the approaches. From the visualization of each model see Figure 5.11, the prediction from our model with local support showed promising results as global representations of the face model are well preserved and also it conveniently preserve the information about the unique features of the samples. From the quantitative analysis point of view, it is obvious from tab.5.1 that the model with the Chamfer distance method of loss estimation gives a better performance. However, our model with local support could provide more useful information about local patterns on the face in contrast to the ordinary point feature generation network.

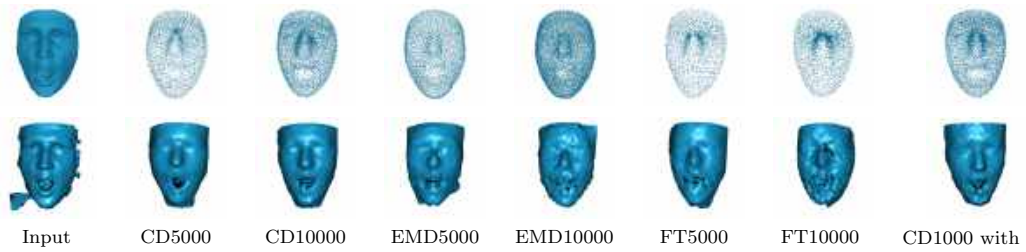


Figure 5.11: Reconstruction of 3D faces from unordered point-set:
the images shows the predicted point set from the different approaches of point generation for 3D face model without post-processing. The first column is the input point set and other columns are the predicted point-set from each model on the top, and its surface representation at the bottom.

	CD (PointNet)	EMD (PointNet)	FT (Ours)	CD with LS (Ours)
5,000pts	1.455	1.770	2.501	-
10,000pts	0.942	1.476	1.620	1.119

Table 5.1: Quantitative evaluation of models:
Mean error estimation from the model of Chamfers Distance, Earth Mover's Distance, Fine tuning from pre-trained model with ShapeNet Data set and CD with Local Support model

5.2.1 Limitation

Through this research work, we explored the possibilities of generating point features for 3D face model from raw data of unordered point sets with the implementation of an Autoencoder network. This implementation has allowed us to compute our method for to identify a local region of occlusion on a face model. From our various experiments we notice some shortfalls on the attempts which could have contributed to the performance of our model.

An important aspect of the limitation with our approach comes with our method of data pre-processing. By considering a sampling method which uses an euclidean distance estimation, does not well represent the information on a face model. The limitation to this is that the bordering effect on the faces are not well represented. As a result, we loose some important information on a face model during sampling.

Also, a good number of real-world 3D face data could significantly improve the performance of our model of point features generation for 3D faces.

Discussion

Through our various experiments, we have reached our target for the first component of the project on recovering face occluded face. We implemented a method to learn local patterns on a face model which will be a basis for restore an occluded face model. Through this approach, intuitive control for local pattern were also made possible which gives room a straight forward approach to compute an effective reconstruction for an occluded face model.

Chapter 6

Conclusion

In this chapter, we intend to give an overview of our contributions towards restoring occluded face models and discuss the prospects of this project.

6.1 Project overview

In this project work, we have explored a new direction of processing 3D face data from unordered point-sets. We developed a 3D face autoencoder which encodes point features in high dimensional space that allows easy manipulation of local patterns on a face model. With the high dimensional encoding of point feature in this network, we introduced a regularization parameter to learn local patterns on the face model. As a result, we can identify local region of occlusions on a face model with the changes observed in values of the high dimensional feature space of an occluded face model.

On the basis of the identified region of occlusion, we can now compute an effective reconstruction of an occluded face model such that the structure of the visible parts of the face are unaffected.

6.2 Future work

Throughout the implementation of point feature generation for 3D faces, we have observed that our approach could attain better performance of reconstruction of a face model. We attempted a training strategy through transfer-learning from the ShapeNet dataset. Contrary to our intuitions, the fine-tuned model could not give a better performance for our predictions. A more realistic solution to this problem could be to acquire more real-wold 3D face dataset to train our model.

With our model that identifies occluded parts on a face model, we propose an approach that translates the weights in the local support layer of our network. This approach will require a new branch of the network where the weight from an occluded face can be translated to the weight of the non-occluded face of the same face sample. With the difference of the translated weight, a reconstruction of the occluded face can be achieved.

Bibliography

[1] Soubhik Sanyal Anurag Ranjan, Timo Bolkart and Michael J. Black. Generating 3d faces using convolutional meshautoencoders. arXiv, 2018.

[2] Hertel F Goncalves J Bernard F, Gemmar P and Thunberg J. Linear shape deformation models with local support using graph-based structured matrix factorisation. CVPR, 2016.

[3] Andreas Morel-Forster ClemensBlumer Bernhard Egger, Sandro Sch onborn Andreas Schnei-der Adam Kortylewski and Thomas Vetter. Occlusion-aware 3d morphable models and an illumination prior for face image analysis. International Journal of Computer Vision, 2018.

[4] Nash C and Williams K. The shape variational autoencoder: A deep generative model of part-segmented 3d objects. Comp. Graph. Forum., 2017.

[5] Hao Su Charles Ruizhongtai Qi, Li Yi and Leonidas JGuibas. Pointnet++: Deep hierarchical feature learning on pointset in a metric space. NeurPIS, 2017.

[6] Maturana D. and Scherer S. Voxnet: a 3d convolutional neural network for real-time object recognition. IEEE Conference on Intelligent Robots and Systems, 2015.

[7] Jongmoo Choi andGerard G Medioni. Donghyun Kim, Matthias Hernandez. Deep 3d face identification. International Journal of Central Banking, page pages 133–142, 2017.

[8] Dmitry Kalenichenko Florian Schroff and James Philbin. Facenet : A unified embedding for face recognition and clustering. CVPR, 2015.

[9] Zhang G Gao L and Lai Y. Lp shape deformation. Science China Information Sciences, 2012.

[10] Syed Zulqarnain Gilani and Ajmal Mian. Learning from mil-lions of 3d scans for large-scale 3d face recognition. CVPR, 2018.

[11] Leonidas Guibas. Haoqiang Fan, Hao Su. A point set generation network for3d object reconstruction from a single image. arXiv, 2016.

[12] Anuj Srivastava Mo-hamed Daoudi Hassen Drira, Boulbaba Ben Amor and Rim Slama. 3d face recognition under expressions, occlusions, and pose variations. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013.

[13] Zhong Z. Liu Y. Huang Z., Yao J. and Guo X. localized decomposition of deformation gradients. Comp. Graph. Forum, 2014.

[14] Shiguang Shan Jiancheng Cai, Han Hu and Xilin Chen. Fcsr-gan: End-to-end learning for joint face completion and super-resolution. IEEE, 2019.

[15] Clarisse Weischedel Keenan Crane and Max Wardetzky. Geodesics in heat: A new approach to computing distance based on heat flow. ACM Trans, 2013.

[16] K Chaudhuri S Yumer E-Zhang H Li J, Xu and Guibas L. Grass: Generative recursive autoencoders for shape structures. ACM Trans, 2017.

[17] Alexa M and Muller W. Representing animations by principal components. Comp. Graph. Forum, 2000.

[18] Wenger S. Wacker M.-Magnor M. Neumann T., Varanasi K. and Theobalt C. Sparse localized deformation components. ACM Trans, 2015.

[19] Andrea Vedaldi Omkar M Parkhi and Andrew Zisserman. Deep face recognition. BMVC, 2015.

[20] Yu-Kun Lai Jie Yang Qingyang Tan, Lin Gao and Shihong Xia. Mesh-based autoencoders for localized deformation component analysis. AarXiv, 2017.

[21] Mo Kaichun R Qi Charles, Hao Su and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. CVPR, 2017.

[22] H. Dibeklioğlu O. Çeliktutan B. Gökberk B. Sankur Savran, N. Alyüz and L. Akarun. Bosphorus database for 3d face analysi. BIOID 2008, 2018.

[23] Mia K Markey Shalini Gupta and Alan C Bovik. Anthropometric 3d face recognition. International Journal of Computer Vision, 2010.

[24] Huang Q Sinha A, Unmesh A and Ramani K. Surfnet: Generating 3d shape surfaces using deep residual networks. CVPR, 2017.

[25] Alberto Del Bimbo andPietro Pala. Stefano Berretti, Naoufel Werghi. Matching 3d face scans using interest points and local histogram descriptors. Computers and Graphics, 2013.

[26] De la Torre F Tena J R and Matthews I. Interactive region-based linear 3d face models. ACM Trans., 2011.

[27] Zeng Z. Wang Y., Li G. and H. He. Articulated-motion-aware sparse localized decomposition. Comp. Graph. Forum, 2016.

[28] Yu F Zhang L Tang X Wu Z., Song S; Khosla A and Xiao J. 3d shapenets: A deep representation for volumetric shapes. CVPR, 2015.

[29] Kihyuk Sohn Xiaoming Liu Xi Yin, Xiang Yu and Man mohan Chandraker. Towards large-pose face frontalization in the wild. IEEE International Conference on Computer Vision, 2017.

[30] Li Yi, Vladimir G Kim, Duygu Ceylan, I Shen, Mengyan Yan, Hao Su, ARCewu Lu, Qixing Huang, Alla Sheffer, Leonidas Guibas, et al. A scalable active framework for region annotation in 3d shape collections. ACM Transactions on Graphics (TOG), 2016.

[31] Jimei Yang Yijun Li, Sifei Liu and Ming-Hsuan Yang. Generative face completion. CVPR, 2017.

[32] Song Zhou and Sheng Xiao. 3d face recognition: a survey. 2018.

[33] Yi Yu. Ziyu Zhang, Feipeng Da. Data-free point cloud network for 3d face recognition. arXiv, 2019.

[34] Hastie T Zou H and Tibshirani R. Sparse principal component analysis. J. Comp. Graph., 2004.