1-1-2018

3d Face Reconstruction And Emotion Analytics With Part-Based Morphable Models

Hai Jin
Wayne State University,

3D FACE RECONSTRUCTION AND EMOTION ANALYTICS WITH PART-BASED MORPHABLE MODELS

by

HAI JIN

DISSERTATION

Submitted to the Graduate School,

of Wayne State University,

Detroit, Michigan

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

2018

MAJOR: COMPUTER SCIENCE

Approved By:

_____

Advisor Date

_____

_____

_____

_____

DEDICATION

To my parents, advisor and friends.

ACKNOWLEDGMENTS

Foremost, I would like to thank my advisor Dr. Jing Hua for the continuous support of my Ph.D study and related research. His positive outlook and confidence in my research inspired me and I am very grateful for his patience, motivation, enthusiasm, and immense knowledge. More importantly, he constantly pushes me to go beyond my intellectual limits and accomplish tasks I deem impossible previously, a challenge I am becoming more and more confident to tackle. I think it was a great opportunity to complete my Ph.D study under his guidance and learn from his research expertise. I could not have imagined having a better advisor and mentor for my Ph.D study.

I also would like to thank my parents, who always behind me whenever I make life-changing decisions, and they encourage me to pursue what my heart tells me to do irrespective of what other people say. They are the examples I look up to for being a person of integrity, decency, and optimism.

I am also grateful to the group members over the years, including Jiaxi Hu, Darshan Pai, Nasim Hamidian and Xinyu Zhang, for the stimulating discussions. A special thanks goes to my fellow students, Yu Chen, Pengfei Ren, Shixing Chen, Yuehua Wang, Chuan Li and Xiaohui Liu. It was fantastic to have the opportunity to work with you. I also would like to thank Ehsan Kazemi, Yan Yan and Yang Yang for providing their handsome portraits to my research projects.

Finally, I must express my very profound gratitude to my friend Zhiguo Zhao and his wife Jinhee Lee, Dong Ruan and his wife Linlin Zhang, Jin Jin and his wife Lian Jin, for their tremendous support during the most difficult period of my life.

Last but not the least, I would like thank Dr. Hongwei Zhang, Dr. Abhilash Pandya, and

Dr. Zichun Zhong for serving on my committee and providing invaluable feedback.

Thanks for all your encouragement!

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1 INTRODUCTION

Understanding emotion status has always been an interesting yet challenging research topic in the past decades [62]. More recently, with the development of more human-centered services, such as targeted advertisement, trending analysis and self-emotion tracking, automatic emotion detection has become increasingly important. Various types of data sources, e.g., images of facial expressions, speech, electroencephalogram (EEG), electrocardiogram (ECG), can be utilized to analyze the emotions [31]. However, speech, EEG and ECG do not always present in daily routine occasions. Thus, using only images or videos for emotion detection is the most feasible and reliable way in many cases. Additionally, it has been proven that the most effective and natural means for classifying emotions are based upon the facial expressions [25, 3]. Therefore, there is growing interest in technologies of extracting the underlying information of facial images to analyze emotional states [73]. Information visualization of this type of visual data is also becoming more important lately.

Many research efforts have been made to explore the facial information through 2D facial images, including face recognition [70], age detection [26] and facial expression recognition [5]. Facial feature extraction from 2D images has been intensively studied and is proven effective. 2D-based methods such as Active Appearance Model (AAM) [51], Active Shape Model (ASM) [3] and Constrained Local Model (CLM) [21] are successfully applied to face tracking and face recognition applications. More recently, Convolutional Neural Network (CNN) became more popular in the field of computer vision, and was actively applied to object detection and recognition tasks [42]. However, 2D-based methods suf-

fer from their fundamental challenges: illumination and orientation variance may result in very different images for the same individual, which make the classification inaccurate and unstable.

To solve the problems existing in 2D-based methods, a potential solution is to extract features from 3D space directly. Therefore, reconstructing high quality 3D face model in a reliable and easy way is essential for solving this problem. With the advancement in visual sensing and acquisition technology, the ability to accurately capture 3D human faces has been significantly improved in recent years. The popular methods include laser scanning [4], structured light scanning [60], RGBD camera [37, 81] or multiview stereo [64]. Capture and reconstruction of 3D face models enable many applications such as modeling [74], animation [15], gaming [48], security [11] and 3D printing [13]. The current solutions often require expensive equipments and a significant level of expertise to achieve high-quality captures and reconstructions. They are far beyond the capability of general end users and therefore limit the potential applications of the technologies. Ichim *et al.* [30] presented a solution for creating 3D avatar using hand-held video input. However, the method mainly focuses on texture synthesis using the input video clip. The geometry of the reconstructed face mainly relies on a Structure-from-Motion (SFM) method to build a point set surface. It requires extensive smoothing and denoising with a morphable surface, which will generate a face model not very similar to the original human subject. Cao *et al.* [15] proposed a system to animate an avatar face using a single camera. Their work focused on tracking a user's facial expressions and then synthesizing the corresponding expression geometry in an avatar rather than reconstructing high-fidelity 3D face models.

The goal of our work is to provide a viable solution for allowing a general end user to

robustly and accurately model and reconstruct the user's 3D face using a single smartphone camera and support the analytics of emotion states. With a single smartphone camera, the user can capture his/her face by himself or herself. Using the captured images as an input, the solution needs to robustly reconstruct a high-quality 3D face. The straightforward idea based on this input data would be relying on Structure-from- Motion or multiview reconstruction methods [64]. Unfortunately, these methods fail upon this low-resolution, blurred, noisy and often incomplete data. Robust surface reconstruction of a high-quality face model from the blurred, noisy and incomplete data is a very challenging task. Also, for the end users to record an entire head scan video of him/herself is a time-consuming and uneasy work. To overcome these challenges, we develop a part-based 3D face representation, learned from a 3D face database using Non-negative Matrix Factorization (NMF) as prior knowledge, to facilitate robust global and adaptive detail data fitting alternatively to reconstruct an accurate and complete 3D face model. Only two selfie images from the front and side will be used as the input to a later iterative reconstruction process. Our iterative 3D face fitting method permits fully automatic alignment of the NMF part-based 3D face representation to the input facial images and the detailed 2D/3D features to reconstruct a high-quality 3D face model. Our method provides the users a simple and robust 3D face models with difference expressions.

We utilize our 3D face fitting method to construct a 3D face from an input image, which generates a dense correspondence to a reference 3D face. Using a fitted 3D face will not only provide a well registered 3D mesh surface, but also can decompose it into an uniform basis space to obtain normalized features. The generated 3D face can be represented by the weighted sum of the basis functions and the weight vectors can be used as one

of the features to classify the expressions, which can be further translated to emotional status. Our system allows the users to analyze emotions continuously by quantifying and visualizing the detected emotion in Valence-Arousal space.

The 3D information can also be provided to the 3D deep learning methods. Sinha *et al.* [65] presented a 3D model surface learning method based on CNN by creating geometry image from the input 3D shapes. Su *et al.* [66] rendered 3D models to several 2D images using multi-view method and used them to train a Multi-View CNN for shape learning. In a nutshell, these methods still transfer 3D shapes into 2D images as the input of the CNN framework. Instead of performing 2D operations in CNN, Wu *et al.* [76] used 3D voxel filters to process the voxelized depth data. Their model significantly outperformed existing approaches for shape recognition tasks.

To take advantage of the high performance CNN and richer information in 3D-based methods, we propose a 3D Mesh Convolutional Neural Network which performs general operations directly on the surface of the 3D mesh. In the facial expression recognition task case, these operations can be performed on the surface of the reconstructed 3D face model. To obtain a consistent sampling grid across the 3D faces, 3D face models are reconstructed by fitting a deformable face model to the scanned surfaces, in which a dense vertex correspondence can be roughly obtained. This property ensures that the processing operations including convolution and pooling are performed uniformly on the 3D surface.

More importantly, we propose a visual analytics approach to the learned features and networks, which is important for modification and optimization towards better performance of the network. Through an interactive visualization of the learned features and high activation feature areas, our system can demonstrate clustered nodes based on their

activation behaviors, which provides users an intuitive visual analytics on the trained networks, and allows them to interactively modify the networks. Based on the visualization result and through the interactions, the users can better understand expected and discover unexpected features, network node performance, etc., hence better fine tune the trained network and optimize its performance as well as reduce the over-fitting problems.

In order to solve those research problems, we propose the expression and emotion analysis approach by employing the morphable 3D face model. There are three major contributions presented in this work of 3D face reconstruction and emotion analysis.

- We present a deformable NMF part-based 3D face representation from a 3D face database which facilitates robust global and adaptive detail data fitting alternatively through the variations of weights. The bases have the property of local support. Our deformable part-based 3D face model serves as a better morphable model for data fitting and reconstruction under geometry and illumination constraints. We present a fully automated iterative 3D face reconstruction method which automatically registers the deformable part-based 3D face representation to the acquired face images and the detailed geometric features as well as illumination constraints to reconstruct a high-fidelity 3D face model. It provides general end users with a novel 3D face capture and reconstruction solution that robustly and accurately acquires 3D face models from a single smartphone camera.
- Our emotion analysis method is based on 3D morphable face model, which can extract more sensitive and reliable features from reconstructed 3D face to classify different emotions. It presents a fully automated 3D face reconstruction technique for 3D facial expression decomposition and feature extraction. We provide a robust VA

value computation and visualization method to measure the emotion changes continuously in VA space. Our system enables users to monitor and analyze emotions robustly and intuitively from a single camera.

- We present a 3D Mesh Convolutional Neural Network for learning facial expressions on 3D face models. Our method convolves 3D signatures directly over irregular sampled surface based on the geodesic distance. We also present a visual analytics framework for deeper understanding of the automatic feature selection and node activation procedure, and provide interactive node selection and removal operations for network modification and optimization. Our method is robust on the rotation of the face and environmental illumination variance since it focuses on the surface geometry descriptors as learning features instead of the intensities of the regular images.

The rest of the paper is organized as follows:

- Chapter 2: Introduces the essential background knowledge and techniques which are employed in this work.
- Chapter 3: Introduces 3D face reconstruction method based on Non-negative Matrix Factorization. This section presents the process of database decomposition and the process of the 3D reconstruction from 2D images.
- Chapter 4: Shows emotion analytics using the reconstructed 3D face models. By estimating the regression between the features of the reconstructed 3D faces and the emotion, the input faces are mapped to a Valence-Arousal (VA) space for emotion visualization. In the VA space, the emotions are intuitively visualized for emotion analysis.

- Chapter 5: Presents a 3D Mesh Convolutional Neural Network to learn the expressions from the 3D faces.
- Chapter 6: Gives a summary of the thesis.

CHAPTER 2 BACKGROUND

In this work, facial expressions and emotions are analyzed using 3D face model. Therefore, 3D face reconstruction is one essential step to achieve our goal. To generate the 3D face models with expressions, we use a method that reconstructs 3D face models from 2D images. In this method, a pre-scanned 3D face database is used as prior knowledge in the reconstruction process. The database is decomposed into a number of basis faces, which are used to generate an arbitrary 3D face. The following will introduce basic concepts for 3D face reconstruction, deep learning framework and emotion analytics. Typical geometry features are also explained, which are used for training 3D mesh neural networks.

2.1 Feature-based Methods

Generally, 3D descriptors includes principal curvatures, mean curvatures, Gaussian curvatures, conformal factors [29] and heat kernels [67], which can be used to describe the 3D shapes. As Hua *et al.* [29] indicated that a 3D shape can be uniquely defined if the mean curvatures and the conformal factors are given. These 3D descriptors are used for the feature-based 3D model analysis, including 3D face models.

2.1.1 Mean Curvature

In differential geometry, mean curvature is an extrinsic measure of the curvature at a given location of a surface S, and it is also equal to the average of the principal curvatures. On a discrete triangular mesh, we compute the mean curvature at point p by

$$H(p) = \frac{1}{4A} \| \sum_{q \in N(p)} (cot\alpha + cot\beta)\vec{pq} \|^2, \tag{2.1}$$

where α and β are two opposite angles of the shared edge $\vec{pq}$ in two triangles, and $N(p)$ is the set of 1-ring neighbor vertices of vertex p. A is the Voronoi area of the vertex p, which can be computed by

$$A(p) = \frac{1}{8} \sum_{q \in N(p)} (cot\alpha + cot\beta) \|\vec{pq}\|^2, \tag{2.2}$$

if the triangles in the 1-ring neighborhood are non-obtuse. Meyer *et al.* [52] presented the solution for the obtuse case. Fig 1 illustrates the angles and the area of the vertex p.

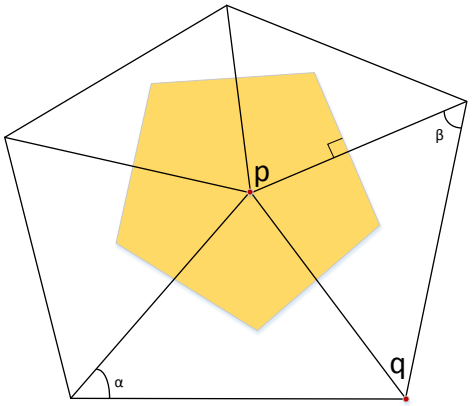


Figure 1: α and β are the opposite angles to the edge $\vec{pq}$. Yellow area shows the Voronoi area associated to the vertex p.

2.1.2 Conformal Factor

In the theorem of differential geometry, a diffeomorphism $f : M \rightarrow N$ is conformal if and only if, for any surface patch σ_m on M, the first fundamental forms of σ_m and $\sigma_n = f \circ \sigma_m$ are proportional. Mathematically, this means that $f \circ ds_m^2 = \lambda ds_n^2$, where λ is called the conformal factor, ds_m^2 andds_n^2 are the first fundamental form on M and N. Given a surface patch M, its conformal image I_c can be created using conformal mapping. Conformal maps preserve both angles and the shapes of infinitesimally small figures, but not necessarily their size or curvature. There is one-to-one correspondence between the

vertices in M and the vertices in I_c.

Conformal factor λ measures the vertex area change of the deformed shape. The discrete conformal factor at point p can be defined as

$$\lambda(p) = \frac{A_e(p)}{A_n(p)}, \tag{2.3}$$

where the $A_e(p)$ and $A_n(p)$ are the averaging areas of the vertex p on surface $\vec{e}$ and $\vec{n}$.

The conformal factor function $\lambda(u, v)$, and the mean curvature function $H(u, v)$, defined on D, satisfy the Gauss and Codazzi equation, as a conformal surface $S(u, v)$ is parameterized on a domain D. Therefore, if $\lambda(u, v)$ and $H(u, v)$ are given with boundary conditions, the surface $S(u, v)$ can be reconstructed uniquely. The mean curvature and the conformal factor are two important signatures which carry fundamental information of a surface.

2.1.3 Heat Kernel Signature

A heat kernel signature (HKS) is a feature descriptor of spectral property of a 3D shape and is widely used in deformable shape analysis [67]. HKS defines a local and global geometric properties of each vertex in the shape by a feature vector, which is used for segmentation, classification, structure discovery, shape matching and shape retrieval. The HKS k is a function of time t, which can be solved by the differential equation $\frac{\delta h}{\delta t} = -\Delta h$. The HKS at a point p is the amount of remaining heat after time T, which is

$$k(p) = \sum_{i=0}^{\infty} e^{-T\lambda_i} \Phi_i^2(p), \tag{2.4}$$

where λ_i and Φ_i are the i^{th} eigenvalue and eigenfunction of the Laplacian-Beltrami

operator. The HKS is an intrinsic property of a given mesh, thus it is stable to noises and articular transformations or even some topological changes. The HKS is also a multi-scale signature of the shape, which means changing the time parameter T can control the scale of the signature. For example, a larger T represents a more global feature and smaller T represents a more local feature of the shape.

2.2 Model-based Methods

Model-based methods try to compose a new face based on a number of basis faces. These basis faces are usually obtained by decomposing a set of scanned face database. Changing the weights of the basis faces can generate different 3D faces. Following methods are typical model-based methods.

2.2.1 PCA based Morphable Face Model

Blanz *et al.* [10] presented a 3D morphable face model in 1999. The morphable model includes two major steps: database decomposition step and reconstruction step. To build a morphable 3D face model, a dataset of 3D faces needs to be scanned and all the scanned 3D faces need to be fully registered. First, 3D faces are scanned using laser scanners or RGBD cameras. Then, to find a uniform dense correspondence among all the unregistered faces, an optical flow algorithm is employed. The algorithm computes a flow filed $(\delta h(h,\phi), \delta\phi(h,\phi))$ that minimizes the difference $\|I_1(h,\phi) - I_2(h,\phi)\|$ between two 3D scans. By finding the flow fields from a reference face model S_{ref}, T_{ref} to each 3D face samples in the database, the dense correspondences are established.

Once the database is fully registered, the database is decomposed using Principal Component Analysis (PCA). The 3D face models are represented with a shape-vector

$S = (X_1, Y_1, Z_1, X_2, ..., Y_n, Z_n)^T \in R^{3n}$, that contains the X, Y, Z coordinates of its n vertices. Similarly, the texture of the shape is represented with a texture-vector $T = (R_1, G_1, B_1, R_2, ..., G_n, B_n)^T R^{3n}$, that contains the R, G, B colors of the n corresponding vertices. A morphable face model is then build from a dataset of m exemplar faces, each represented by its shape-vector S_i and texture-vector T_i. Any new shape S_new and new texture T_new can be reconstructed by a linear combination of the shapes and textures of the m exemplar faces:

$$S_{new} = \sum_{i=1}^{m} \alpha_i S_i, \qquad T_{new} = \sum_{i=1}^{m} \beta_i T_i, \qquad \sum_{i=1}^{m} \alpha_i = \sum_{i=1}^{m} \beta_i = 1. \tag{2.5}$$

The morphable model is then defined as the set of faces $(S_mod(\vec{a}), T_mod(\vec{bb}))$, parameterized by the coefficients $\vec{\alpha} = (\alpha_1, \alpha_2, ...\alpha_m)^T$ and $\vec{\beta} = (\beta_1, \beta_2, ...\beta_m)^T$. Thus, new faces can be reconstructed by finding the optimal parameters α and β that control shape and texture. To make the reconstructed 3D model a plausible human face, the coefficients need to be restricted in a certain range. Therefore, based on the coefficients of the exemplar faces, the probability distributions of the coefficients α and β are computed. Using the distributions as prior, the likelihood of the coefficients can be regulated to prevent extreme cases.

The m exemplar face models are arranged in a $m \times n$ matrix form, where n is the length of each shape vector. Then using PCA, the matrix is decomposed to an orthogonal coordinate system, which is formed by the eigenvectors and covariance matrix:

$$S_{model} = \bar{S} + \sum_{i=1}^{m} a_i s_i, \qquad T_{model} = \vec{T} + \sum_{i=1}^{m} b_i t_i, \tag{2.6}$$

where s_i and t_i are the covariance matrices of geometry and texture datasets. The probability for coefficients $\vec{a}$ is given as follows:

$$p(\vec{a}) \sim exp[-\frac{1}{2}\sum_{i=1}^{m}(a_i/\sigma_i)^2], \tag{2.7}$$

where σ_i is the eigenvalues of the shape covariance matrix. The computation of the probability for coefficient $\vec{b}$ is similar.

Once the morphable face model is built, an arbitrary 3D face model can be reconstructed from a 2D input image by optimizing the shape coefficient α and texture coefficient β. The optimization is done by minimizing the energy function

$$E = \sum_{k=1}^{m} \|I_{input}(\vec{p}_{x,k}, \vec{p}_{y,k}) - I_{model,k}\|^2, \tag{2.8}$$

where k is the index of the vertices, $(\vec{p}_{x,k}, \vec{p}_{y,k})$ is the projected image locations of the model. The energy is minimized by taking the partial derivatives with respect to $\vec{\alpha}$ and $\vec{\beta}$.

2.2.2 Blendshapes

Blendshapes are widely used in the facial expression animation purpose. A set of facial expression meshes can be generated for each person. Using the facial rigging algorithm proposed by Li *et al.* [46], a user specific expression space is generated. By computing a linear combination of the expressions, we can interpolate the intermediate expressions. Following the Ekman's Facial Action Coding System [23], 46 base expressions $B = B_0, B_1, ...B_46$ are captured for each person, and any new expression H is then

computed by

$$H = B_0 + \sum_{i=1}^{46} \alpha_i (B_i - B_0), \tag{2.9}$$

where α is the weight coefficients for each expression. Employing the rigging algorithm, the blendshape system is optimized to a more general model $A = A_0, A_1, ...A_46$, which improves the optimization performance.

2.2.3 Bilinear Face Model

Extended from Morphable face model, the bilinear face model decomposes a dataset with 3 dimensions: vertices, identity, expression. The bilinear facial mesh dataset contains n subjects with the same 47 facial expressions (1 neutral and 46 others). These face meshes are obtained by deforming a reference face model to the scanned raw depth image, therefore, all these meshes have a one-to-one correspondence by nature. The dataset is assembled into a 3-mode tensor T, which represents vertices $\times$ identities $\times$ expressions. The data tensor is arranged in an obvious fashion, so that each slice with varying second factor and fixed third factor contains face vectors with the same expression (for different identities), and each slice with varying third factor and fixed second factor contains the same identity but with different expressions. The tensor dataset is decomposed by N-mode singular value decomposition (SVD). The vertex dimension is excluded in the decomposition, since the entire face is required in most of the facial reconstruction applications.

The N-mode SVD process is represented as

$$T \times_2 U_{id}^T \times_3 U_{exp}^T = C, \tag{2.10}$$

where T is the data tensor and C is the decomposed core tensor. U_id and U_exp are orthonormal transform matrices, which contain the left singular vectors of the 2nd mode (identity) space and 3rd mode (expression) space respectively. Similar to PCA, N-mode SVD sort the variance of C in decreasing order for each mode. This enables data compression by removing the insignificant components of C. Therefore, an approximation of the original tensor can be recovered by

$$T \simeq \hat{C} \times_2 \hat{U}_{id} \times_3 \hat{U}_{exp}, \tag{2.11}$$

where $\hat{C}$ is the simplified core tensor obtained by keeping the significant columns of the original core tensor in mode 2 and 3. $\hat{U}_{id}$ and $\hat{U}_{exp}$ are the simplified matrices from $\hat{U}_{id}$ and $\hat{U}_{exp}$.

With $\hat{C}$, any facial expression of any person can be approximated by the tensor contraction

$$S = \hat{C} \times_2 w_{id}^T \times_3 w_{exp}^T, \tag{2.12}$$

where w_{id} and w_{exp} weight vectors for identity and expression respectively.

2.3 Deep Learning-based Method

Recent achievements in deep learning enabled many applications in machine learning. Deep learning commonly rely on complicated neural networks and Convolutional Neural Network (CNN) is one particular class of neural network which focuses on image data. Recently, CNN is also modified to learn 3D datasets and achieved great performance in shape classification tasks [65, 66, 76]. We explains basic knowledge of CNN in following

section.

2.3.1 Convolutional Neural Networks

CNN is usually built with convolution layer (C), pooling layer (P) rectified linear units (LeRu) layer, fully connected layer (FC) and followed by loss layer. Different layers serve for different purposes and a proper arrangement of these layer components can maximize the performance of the learning.

Convolutional layer The convolutional layer is the core component in a CNN framework. Convolution preserves the spatial relationship between pixels by learning image features using small squares of input data. This layer consists of a number of filters, and each filter is a matrix with different weights in its elements. During the convolution procedure, the filters slide over the input image to compute element-wise weighted summation, producing a 2 dimensional feature map of the filter. As a result, the network learns filters that activate when it detects some specific type of feature at some spatial position in the input.

All the feature maps generated by different filters forms a output volume of the convolution layer. Therefore, every entry in the output volume can also be interpreted as an output of a neuron that looks at a small region in the input and shares parameters with neurons in the same activation map. Convolutional layer often followed by ReLU layer and pooling layer, which is discussed as follows.

Pooling layer Pooling is another important concept in CNN, which is a form of non-linear down-sampling. There are several pooling methods, such as max pooling, average pooling, median pooling and etc. In case of Max Pooling, the image is partitioned into a set of non-overlapping areas i.e. a 2×2 window, and in each area the largest element is

preserved. In practice, Max Pooling has been shown to work better.

Pooling makes the input representations (feature dimension) smaller and more manageable, therefore, it reduces the number of parameters and computations in the network and prevents overfitting. Pooling also makes the network invariant to small transformations, distortions and translations in the input image. It is common to periodically insert a pooling layer between successive convolutional layers in a CNN architecture.

ReLU layer Non-linear activation functions are often applied immediately after convolution layer to introduce nonlinearity to the convolved outputs. It increases the nonlinear properties of the decision function and of the overall network without affecting the receptive fields of the convolution layer.

In the past, functions such as the saturating hyperbolic tangent $f(x) = \tanh(x)$ and the sigmoid function $f(x) = (1 + e^{-x})^{-1}$ were often used. However, Rectified Linear Units (ReLU) function becomes more popular since it improves the computational performance several times faster without significantly losing the accuracy. This layer applies the nonlinear activation function $f(x) = \max(0, x)$ to all of the values in the input volume. In basic terms, this layer changes all the negative activations to 0.

Fully connected layer Fully connected layer usually follows several convolutional layers and pooling layers to learn the high-level knowledge. Just as a normal neural network, neurons in a fully connected layer connect to all activations in the previous layer. Therefore, their activations can be computed with a matrix multiplication followed by a bias offset.

Loss layer A loss function (error, cost, objective function) is an important concept in general machine learning framework, which drives the learning process toward the better result. The loss function measures how good the current parameters describes the training data. It converts the input multi-variant data to a scaler value and compares it with the true target value. Hence, the learning result is evaluated by the loss function and the goal of learning is to minimize the loss function.

In CNN framework, three types of loss function are often used. Softmax layer is used for predicting a single class from several mutually exclusive classes. Sigmoid cross-entropy loss is used for predicting the independent probability values in $[0, 1]$. Euclidean loss is used for regressing to real-valued labels $(-\infty, \infty)$.

CHAPTER 3 3D FACE RECONSTRUCTION

3.1 Introduction

In this section, we introduce a novel 3D face modeling and reconstruction solution that robustly and accurately acquire 3D face models from a couple of images captured by a single smartphone camera. Two selfie photos of a subject taken from the front and side are first used to guide our Non-Negative Matrix Factorization (NMF) induced part-based face model to iteratively reconstruct an initial 3D face of the subject. Then, an iterative detail updating method is applied to the initial generated 3D face to reconstruct facial details through optimizing lighting parameters and local depths. Our iterative 3D face reconstruction method permits fully automatic registration of a part-based face representation to the acquired face data and the detailed 2D/3D features to build a high-quality 3D face model. The NMF part-based face representation learned from a 3D face database facilitates effective global and adaptive local detail data fitting alternatively. Our system is flexible and it allows users to conduct the capture in any uncontrolled environment. We demonstrate the capability of our method by allowing users to capture and reconstruct their 3D faces by themselves.

3.2 Related Work

Capturing and reconstructing 3D surfaces of objects is one of major research topics in geometric modeling, computer graphics and computer vision. Human face reconstruction and modeling is one of the most active ones among general surface reconstructions. Various methods on face modeling [39] have been extensively studied.

Thanks to the rapid development of capturing devices, the modeling of faces has be-

come more accurate and automated. From scanning directly by professional laser scanners, to using multiple high-resolution cameras to capture 3D face based on multi-view geometry [53], researchers have achieved significant successes recently. Commercial laser scanners, such as $Cyberware$ and $NextEngine$, can now provide us high-quality face modeling. Also, stereo-based face modeling techniques [6, 7, 2], relying on multiple high-resolution cameras, can also achieve high-quality face modeling. For example, Beeler *et al.* [6] used five high resolution digital single-lens reflex (SLR) cameras to capture accurate 3D geometry of a face from a synchronized shot. However, the costs of these systems are still very high and they require a complicated calibration process before actual operation, which limits the uses of these systems in non-studio environments or by general end users.

On the contrary, to using expensive high performance scanners or stereo capturing systems, Blanz and Vetter [9] presented a morphable face model for reconstructing 3D face from a single image and Lei *et al.* [45] presented a face shape recovery method using a single reference 3D face model. In order to recover the missing depth information from 2D image, prior knowledge of face is needed. Learning through a face database is an effective approach for tackling this problem. Therefore, statistical face models based on Principal Component Analysis (PCA) are proposed and constructed [9], and then used as prior for estimating depth information. Similar face fitting methods, such as piecewise PCA sub-models [69], were proposed as well. Approaches based on 2D images achieved plausible 3D face reconstruction results [9], however, they need to carefully tune the parameters for pose and illumination, which requires a lot of empirical knowledge and makes it impractical for general end users. Also, the detailed geometry reconstruction is the main limitation of these methods due to the global property of PCA methods. As the extension

of their work, Blanz *et al.* applied the morphable 3D face reconstruction method to facial recognition problem [10].

More recently, modeling based on RGBD camera such as $Kinect$ [81, 56], has become another active research topic. Chen *et al.* [18] proposed a system that captures a high-quality a face model and its performances using a single $Kinect$ device. They provided a markerless motion capture approach that increases the subjects' flexibilities and improves the resolution of facial geometry. Newcombe *et al.* [56] presented a $Kinect$ based 3D reconstruction method for non-rigid objects such as human face. Without using a RGBD device, Cao *et al.* [15] proposed a system to animate an avatar face using a single camera. Other than capturing and reconstructing the entire face, they only detected a set of feature points for computing shape regression to animate the target avatar face. As the extension work, they proposed displaced dynamic expression regression method to further improve the performance of the system [14]. Their work focused on tracking and synthesizing facial expression geometry rather than reconstructing high-fidelity 3D face models.

Another direction of work is 3D face reconstruction from photometric stereo-based method. Suwajanakorn *et al.* [68] presented an impressive 3D face reconstruction technique from a collection of images or a clip of videos of the person. They first learned an average 3D face of the subject from the input images as the base shape, which was then used to fit the individual images with different expressions. A shape-from-shading method was used to optimize the fine details of the shape. However, since this method is reconstructing the shape by optimizing pixel depth of the image, the side of the 3D face such as cheeks and ears are not fully reconstructed. By this means, this method is a 2.5D reconstruction of the face.

SFM-based shape reconstruction has also been researched extensively [75, 32]. However, it is difficult to reconstruct a fine detailed 3D face model due to the high noise-to-signal ratio. Smoothing and denoising to the point cloud data will significantly reduce the high frequency details of the model. Ichim *et al.* [30] presented a dynamic 3D avatar creation approach from mobile devices. The approach uses a noisy point cloud built from SFM as the constraint to deform the template 3D head. As the purpose is for entertainment applications on mobile devices, the details are added by refining the albedo texture and normal map instead of actually adjusting the local geometry details. In order to create a realistic detailed 3D avatar head, many off-line edits are still needed, which is a difficult task for the general end users who have little or no 3D modeling knowledge.

Our work mainly focuses on capturing and reconstructing face geometry robustly and automatically by general end users. Therefore, in this paper, we assume the input two images for our 3D face reconstruction are self-acquired from a single camera of a mobile device. Extending from learning-based approach, we instead establish a deformable part-based 3D face representation based on non-negative matrix factorization of a 3D scanned face database prior to the reconstruction stage. Compared to the previous methods, our deformable NMF-based 3D face model serves as a better and more robust morphable model for data fitting and reconstruction as the NMF bases are corresponding to localized features that correlate better with the parts of 3D faces under geometry and illumination constraints. During the reconstruction stage, our method can produce a high-quality 3D face model from the input images and recover details of the subject's face through surface reconstruction from shading constraints. Figure 2 illustrates the entire process, which is fully automated without users' intervention.

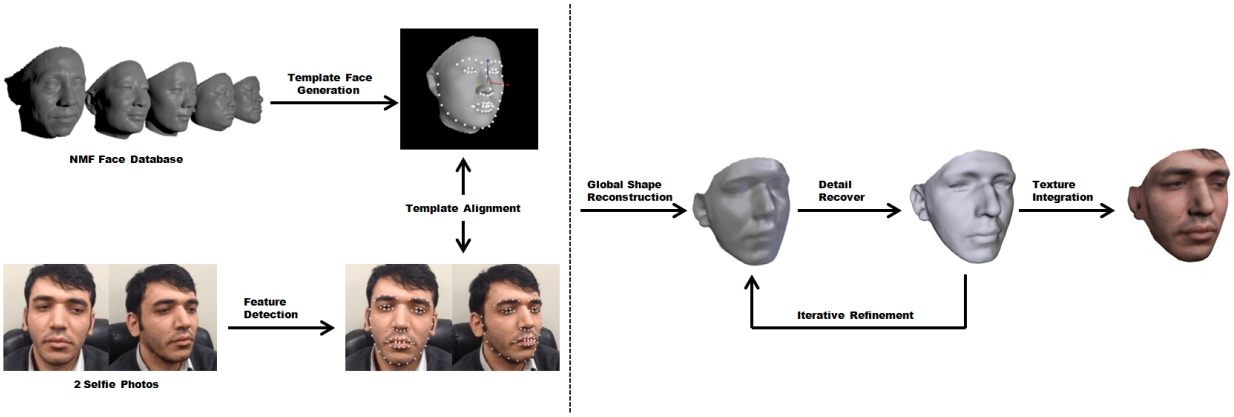


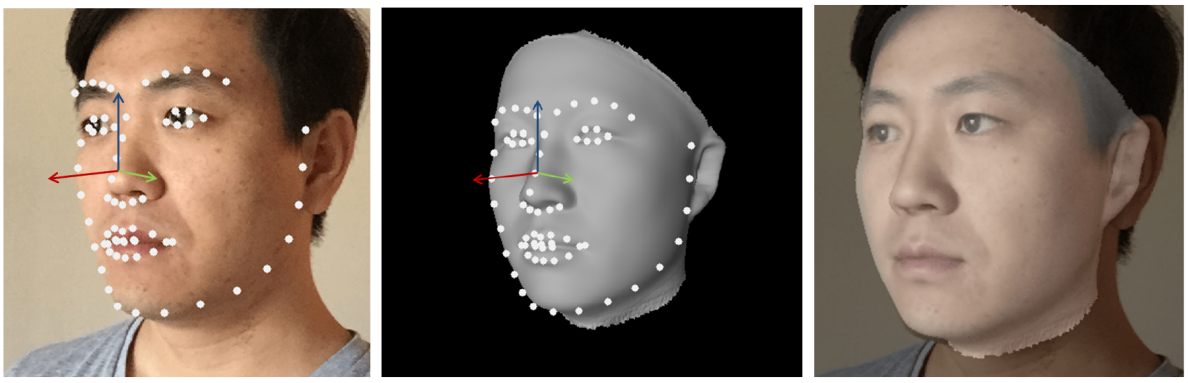
Figure 2: The pipeline of our iterative 3D face reconstruction based on deformable NMF part-based 3D face model.

3.3 Face Reconstruction through Part-based 3D Face Model

In this section, we present in detail the 3D face reconstruction technique using our deformable NMF part-based 3D face model. In Section 3.3.1, we first explain Constrained Local Models [21] for the initial feature point detection and pose estimation based on the input images. Then, in Section 3.3.2, we explain the process to build our deformable part-based 3D face representation based on non-negative matrix factorization of a 3D face database. In Section 3.3.3, we show how the part-based 3D face model can be used as a deformable model to reconstruct a high-quality face from a frontal and a side facial images. In Section 3.3.4, we present the detail fitting process based on the illumination constraints. Briefly, our approach reconstructs a final 3D face iteratively by alternating two steps: global fitting and detail fitting.

3.3.1 Initial Pose Estimation and Template Face Alignment

An accurate initial alignment of the template 3D face to the input images is important for the fitting process since a good initial state may significantly reduce the optimization



(a) (b) (c)

Figure 3: Feature point detection and initial alignment. (a) is detected features points and the estimated pose according to the input image; (b) is affine transformed template face based on the estimated parameters according to (a); (c) is aligned template face to the input face.

iterations and improve the reconstruction quality. We employ the Constrained Local Models (CLM) [21] to estimate the feature points P_n (n is the number of points), which are used by a feature-based head pose estimator to obtain the initial template translation (T), rotation (R) and scale (S) automatically. Figure 3 shows the detected feature points P_n on the input facial image and the posing direction of the face. P_n is a vector of 2D pixel coordinates of the feature points on the image.

Prior work [20, 55] has studied how to estimate head poses from monocular image. In this paper we propose a method similar to [20]. The ground-truth poses and their depth maps are acquired via a $Kinect$ device along with the images to build a training database. Then, we train a view-based appearance model to estimate the head poses. For each key frame i, we obtain a training set $F_i = \{P_{ni}, \Omega_i\}$, where P_{nk} are the feature points detected by the CLM and $\Omega_i = T, R, S$ are the affine transformation parameters. Note that, the translation T and scaling S can be estimated straightforwardly using the shape

of the feature points. Therefore, we focus on solving the rotation R with the training set P_{ni}, R. Particularly, the rotation term R consists of three angles $\theta_x, \theta_y, \theta_z$ around x, y, z axis, respectively. Therefore, the training set can be noted as $\{P_{ni}, \theta\}$, where θ represents θ_x, θ_y, or θ_z. The shape of the feature points can be represented as

$$P = \bar{P} + Q\vec{\beta}, \tag{3.1}$$

where $\bar{P}$ is the average shape of feature points among all the faces in the training database, Q is the variation matrix of the training data, which can be obtained using the PCA method, and $\vec{\beta}$ is the coefficient which controls the shape of feature points. The prediction model can be established via the following regression model:

$$\vec{\beta} = \vec{a} + \vec{b}cos(\theta) + \vec{c}sin(\theta), \tag{3.2}$$

where $\vec{a}, \vec{b}, \vec{c}$ are the parameters to be trained from the training set F. Eq. 3.2 can be solved by $(cos(\theta), sin(\theta))' = R^{-1}(\vec{\beta} - \vec{a})$, where R^{-1} is pseudo inverse of $(\vec{b}|\vec{c})$, i.e., $R^{-1}(\vec{b}|\vec{c}) = I_2$. Thus, given a detected feature point set, P, we first compute the representing coefficient $\vec{\beta}$ using Eq. 3.1, then, we can obtain θ based on the trained predictive model, i.e., Eq. 3.2.

Once the template face is transformed to the same pose as the input image, the same number of feature points are detected on the rendered template face (i.e., 2D projected image of the 3D template face), which are traced back to 3D space to obtain the nearest corresponding vertices V_n. V_n, corresponding with P_n, is a 3D vertex coordinate vector of the feature points on the 3D template face model. We denote this operation as $\mathbb{F}$ and the

details will be described in Section 3.3.3.

3.3.2 Deformable Part-based 3D Face Model for Data Fitting

PCA and vector quantization(VQ) are two common approaches to decompose a data and PCA is also widely used in 3D face data. PCA learns the face data globally and decompose it to 'eigenfaces', which are basis vectors in the face space. Different from PCA and VQ, NMF decomposes a shape in localized features [43]. In this paper, we construct a deformable 3D face representation by parts based on non-negative matrix factorization, which will significantly improve the reconstruction of details of the 3D face. Any face can be represented as a linear combination of basis parts. The part-based 3D face model permits better local control, therefore leading to more accurate and robust morphable fitting to the target.

To generate better performing NMF bases, the scanned 3D face data samples need to be carefully aligned and registered first. There exist many methods facilitating this task [47]. Given a 3D face database with M examples, we first employ multi-scale expectation-maximization iterative closest point method to accurately register all the 3D face examples and then resample all the examples into the same number of vertices to establish a dense mapping and indexing [27]. The 3D face database can then constructed as a $N \times M$ matrix S, where N is the number of vertices in a 3D face and M is the number of face examples in the database. Each column represents the geometry of the face with a data vector of 3D coordinates, $s_i = \{x_1, y_1, z_1, x_2, y_2, z_2, \ldots, x_n, y_n, z_n\}^T \in \mathbf{R}^{3n}$. Note that, the quality of the dense correspondence will significantly affect the result of factorization.

Next, non-negative factorization of matrix S is constructed as $S \approx BW$, where B is the

basis matrix and W is the weight, or it can be represented as

$$S_{ij} \approx (BW)_{ij} = \sum_{a=1}^{r} B_{ia} W_{aj}, \tag{3.3}$$

where r is the rank of factorized basis. Then each face in the database can be restored from

$$s_k = B\vec{\mathbf{w}}_k, \tag{3.4}$$

where $\vec{\mathbf{w}}_k = (w_1, w_2, ...w_a)^T$ is corresponding column vector in weight matrix W. New faces can be generated by manipulating the weight vector $\vec{\mathbf{w}}_k$ and compute the linear combination of the bases.

To find a factorization, we need to solve the following optimization problem,

$$(B, W) = \underset{B \geq 0, W \geq 0}{\operatorname{argmin}} \|S - BW\|^2. \tag{3.5}$$

According to the theorem in [44], the Euclidean distance $\|S - BW\|$ does not increase under the following update rules:

$$W_{aj} \leftarrow W_{aj} \frac{(B^T S)_{aj}}{(B^T BW)_{ia}}, \qquad B_{ia} \leftarrow B_{ia} \frac{(SW^T)_{ia}}{(BWW^T)_{ia}}. \tag{3.6}$$

In practice, B and W are initialized as random dense matrix [8] and a simple additive update rule for weight W is used as in [44],

$$W_{aj} \leftarrow W_{aj} + \eta_{aj}[(B^T S)_{aj} - (B^T BW)_{aj}], \tag{3.7}$$

where

$$\eta_{aj} = \frac{W_{aj}}{(B^T BW)_{aj}}. \tag{3.8}$$

Once NMF basis matrix B is computed, arbitrary new face can be decomposed based on the bases and represented by the corresponding weight vector. In other words, varying the weights over the NMF basis matrix B constitutes a deformable part-based 3D face model that can be used to fit in a nonrigid means to any given 2D/3D face data input. Therefore, we name $s = B\vec{w}$ as a deformable part-based 3D face representation which carries the prior knowledge of faces for nonrigid fitting.

We used 120 scanned face data for training deformable part-based 3D face model. In this paper, the data was normalized and registered based on the multi-scale expectation-maximization iterative closest point method [27]. Every face data has 60000 vertices, which are represented as vector $\vec{x}_i = \{r_1, h_1, \theta_1, r_2, h_2, \theta_2, \ldots, r_n, h_n, \theta_n\} \in \mathbf{R}^{3n}$. Since NMF requires non-negative elements in the matrix, face data samples were transformed into cylindrical coordinate system so that all the values in the vector are positive. Data vectors of 120 subjects formed a $3n \times 120$ matrix for NMF decomposition. We chose 120 columns for basis matrix as the factorization basis, therefore, the weight vector $\vec{w}$ has 120 elements. Figure 4 shows some samples of our 3D face database used in this paper and Figure 5 shows the local support of the part bases on an average face model. Note that, all the faces used in training the deformable part-based 3D face model are not employed for testing the performance of our system.

Figure 4: The sample 3D faces in our database.

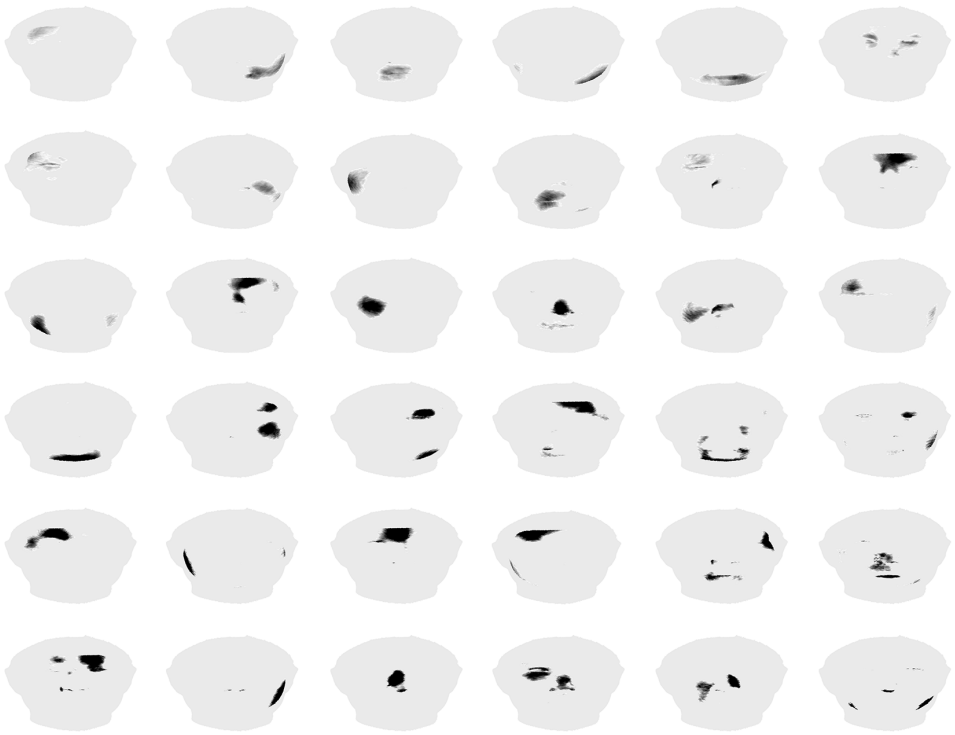


Figure 5: The randomly selected 36 computed NMF basis projected onto an average face to display the local support of the basis.

3.3.3 Iterative Global Reconstruction with Part-based 3D Face Representation

The reconstruction process is divided into two major parts: global fitting and detail fitting. In this section we explain the global shape reconstruction via feature point fitting by updating the deformable part-based 3D face model iteratively .

In a global fitting step, the deformable part-based 3D face model $B\vec{w}$ will optimize its weights, $\vec{w}$, based on the previously estimated rotation, scaling and translation factors (in Section 3.3.1) to fit the feature points P_n in input images I. In order to find the best fitted result, we minimize the Euclidean distance as follows,

$$\vec{w} = \underset{\vec{w}}{\operatorname{argmin}} \sum_{k=1}^{m} \|P_n^k - \mathbb{P}^k(\mathbb{F}(R^k B\vec{w} + T^k))\|^2, \tag{3.9}$$

where m is the number of images used for global fitting, B is the basis of the part-based 3D face representation, R is a rotation and scaling matrix, T is a translation matrix, $\mathbb{F}$ is feature point extraction operation and $\mathbb{P}$ is the projection operation. We followed the approach in Zhang *et al.* [80] to estimate the camera intrinsic parameters, which can be further used to compute the projection matrix. In practice, we use two images taken from the front and the side as the input, thus, the number of images $m = 2$. However, the number of input images can be unrestricted in our deformable fitting model. In general, using more images may produce a better quality reconstruction, but the computational cost will also increase and the user-experience can be adversely affected. To our experience, two images can already provide a rather high-quality reconstruction while keeping an excellent performance in terms of computational time. Figure 2 shows the feature points detected in both the frontal and side images.

We solve the Eq. 3.9 via gradient descent method. We compute the partial derivative of the energy term in Eq. 3.9 with respect to the weight vector $\vec{w}$ as follows,

$$\bigtriangledown E(\vec{w}) = \frac{\partial E}{\partial \vec{w}} = -2\sum_{k=1}^{m}(P_n^k - \mathbb{P}^k(\mathbb{F}(R^k B\vec{w} + T^k)))\frac{\partial \mathbb{P}^k(\mathbb{F}(R^k B\vec{w} + T^k))}{\partial \vec{w}}, \tag{3.10}$$

where

$$\frac{\partial \mathbb{P}^k(\mathbb{F}(R^k B\vec{w} + T^k))}{\partial \vec{w}} = \vec{c}_k \tag{3.11}$$

is a constant vector $\vec{c}_k$ for each image. Thus, the gradient in each iteration is only determined by the distance between the projected feature vertices in the current stage and the target feature points in the input image. In each iteration i, the weight vector $\vec{w}$ is updated by

$$\vec{w} \leftarrow \vec{w} - \nabla E(\vec{w}_i). \tag{3.12}$$

The Algorithm 1 iteratively updates the weight vector and obtains the optimized $\vec{w}$. This process continues until the weight vector converges, which usually takes around 5$\sim$7 iterations. This step recovers the global features in the data, such as face size, and approximates the shape of different parts of the face. Since we can obtain the correspondence between the reconstructed model and the input image sequence, we can project the registered images to the result shape to obtain a texture, reconstructed 3D face model, $(B\vec{w}, \rho)$, where ρ is the texture over $B\vec{w}$. This will be used for the next NMF based detail fitting process.

Algorithm 1 Iterative Global Reconstruction

procedure ITERATIVE 3D FACE RECONSTRUCTION
    $\vec{w}, T, R, \mathbb{P}, \mathbb{F}, threshold \leftarrow initialization$
    Compute the gradient $\nabla E(\vec{w}_i)$ using Eq. 3.10
    while $\nabla E(\vec{w}_i) > threshold$ **do**
        $\vec{w} \leftarrow \vec{w} - \nabla E(\vec{w}_i)$
        Re-compute $\nabla E(\vec{w}_i)$ using Eq. 3.10
    $S = B\vec{w}$

3.3.4 Shading Based Detail Reconstruction

Although our part-based fitting can reconstruct a quite plausible 3D face with well fitted global features, the details (such as the major wrinkles and folds around mouth and nose) are still akin to the template shape. Thus, we perform a detailed refinement process based on the result of global fitting.

In the NMF detail fitting step, we perform data fitting to the images in order to fine tune the model by minimizing the Euclidean distance from the rendered 3D face to input images. Since the global fitting recovered the rough shape of the face and the transformation matrix, the current face model can be automatically projected to the corresponding images using the information obtained from previous feature point alignment process. We denote the rendered image of the 3D face model as $\hat{I} = \mathbb{R}(B\vec{w}, \rho)$. That is to say, we project and render each vertex of the face model on the image plane to form $\hat{I}$, and the comparison between the rendered image and the original image is based on the projected locations of the 3D vertices. Therefore, the main goal of the optimization problem is to minimize the sum of distances as follows,

$$\vec{w} = \underset{\vec{w} \in W}{\operatorname{argmin}} \left(\sum_{i=1}^{m} \| I_i(\mathbb{P}^i(V)) - \hat{I}_i \|^2 + \eta \| \vec{w} - \vec{w}_g \|^2 \right), \tag{3.13}$$

where m is the number of images used for detail fitting, ρ is the albedo texture, $\mathbb{R}$ is the rendering operation, $I_i(\mathbb{P}^i(V))$ is the re-sample of the input image I based on the projected locations of the 3D vertices V on the 2D image domain, η is the regularization coefficient and $\vec{w}_g$ is the weight vector of the local parts derived from global fitting. The second term is the regularization term which constrains the final shape is close to the result of previous global fitting stage. In practice, we only use the frontal image for detail refinement, i.e., $m = 1$, since most of the details are captured in the frontal view of the image. Since the rendering operation $\mathbb{R}_i$ and the corresponding image I_i is known, the only variable that needs to be updated is weight vector $\vec{w}$. Based on the same idea to the global shape reconstruction step, we compute the derivative of the error function with respect to $\vec{w}$ to find the maximum decent direction, which is used for updating the weight vector.

In order to compute the partial derivative of the energy term with respect to $\vec{w}$, the shading model of the rendering operation needs to be defined. Inspired by Suwajanakorn *et al.* [68], we transform the optimization problem in Eq. 3.13 to an optimization problem for the photometric normals as follow:

$$\boldsymbol{N} = \underset{\boldsymbol{N}}{\operatorname{argmin}} \|I(\mathbb{P}(V)) - \hat{I}\|^2. \tag{3.14}$$

For shading computation, we use the Phong reflectance model in our method. In this paper, the rendered image $\hat{I}$ of the 3D face is computed by

$$\hat{I} = \mathbb{R}(B\vec{w}, \vec{\rho}) = (k_a + k_d(\boldsymbol{N}\vec{l}) + k_s(\boldsymbol{V} * \vec{\boldsymbol{r}})) \circ \vec{\rho}, \tag{3.15}$$

where k_a, k_d and k_s are constant weights of ambient light, diffuse light and specular light, respectively. $\vec{l}$ is the light directions. $\boldsymbol{N}$ are the normals at the vertices and $\vec{\rho}$ are the albedo vector containing the texture information at each vertex. $\vec{\boldsymbol{r}}$ are the reflection vectors and $\boldsymbol{V}$ are vectors from each vertex to the view point. The $*$ represents a row-wise inner product and the $\circ$ represents the element-wise product. Since we assume a weak specular reflection in the model, we ignore the specular term in Eq. 3.15 when optimizing vertex normals. We compute the final vertex normals by minimizing the following equation:

$$\{\boldsymbol{N}, \vec{l}, k_a, k_d\} = \underset{\boldsymbol{N},\vec{l},k_a,k_d}{\operatorname{argmin}} \|I(\mathbb{P}(V)) - (k_a + k_d(\boldsymbol{N}\vec{l})) \circ \vec{\rho}\|^2. \tag{3.16}$$

Lighting Optimization: To optimize the shading through changing the vertex normals by Eq. 3.16, we first estimate the lighting parameters $\vec{l}, k_a, k_d$ by solving the optimization problem,

$$\{\vec{l}, k_a, k_d\} = \underset{\vec{l},k_a,k_d}{\operatorname{argmin}} \|I(\mathbb{P}(V)) - (k_a + k_d\boldsymbol{N}\vec{l}) \circ \vec{\rho}\|^2, \tag{3.17}$$

where the vertex normals $\boldsymbol{N}$ and the albedo ρ are considered as constants at this stage. To simplify the optimization problem, we let the albedo is equal to the input image I, so the problem becomes

$$\{\vec{l}, k_a, k_d\} = \underset{\vec{l},k_a,k_d}{\operatorname{argmin}} \sum_{i=1}^{e} \|1 - (k_a + k_d\vec{l} \cdot \vec{n}_i)\|^2, \tag{3.18}$$

where e is the number of vertices and $\vec{n}_i$ is the normal vector of each vertex. In practice, we solve can the linear equation in a linear time by randomly selecting 40 vertices on the shape. Since it is a over determined linear system, it can be easily solved by QR factorization. Figure 6 shows the result of lighting optimization.

(a) (b) (c)

Figure 6: Lighting optimization result: (a) is the initial random lighting; (b) is optimized lighting; (c) is the original image.

Normal Optimization: After we estimate the optimal lighting parameters, we compute the partial derivative of normal vector $\vec{N}$ with respect to the weight vector $\vec{w}$, and then the Eq. 3.13 can be solved by chain rule. We define the normal of each vertex as follow:

$$\vec{n}_i = \frac{\vec{u} \times \vec{v}}{\|\vec{u} \times \vec{v}\|}, \tag{3.19}$$

where $\vec{u}$ is the vector from V_i to its adjacent vertex in positive x direction and $\vec{v}$ is the vector from V_i to its adjacent vertex in positive y direction. Since the vertices are in 3D domain, we pre-compute the vertex location adjacency in cylinder coordinate system. Here, we only update the depth value of each vertex to modify the vertex normal. Therefore, we compute the partial derivative on z element of the normal, which is $\frac{\partial \vec{n}_{iz}}{\vec{w}}$. Therefore, the Jacobian of the normal vector $\vec{N}_z$ is

$$J = \frac{\partial \vec{N}_z}{\partial \vec{w}} = [\frac{\vec{n}_{1z}}{\vec{w}}, \frac{\vec{n}_{2z}}{\vec{w}}, ..., \frac{\vec{n}_{kz}}{\vec{w}}] = [\vec{\delta}_1, \vec{\delta}_2, ..., \vec{\delta}_k]. \tag{3.20}$$

Algorithm 2 Iterative Detail Refinement

procedure ITERATIVE DETAIL REFINENMENT
  $\vec{w}, \alpha, I, threshold \leftarrow initialization$
  Compute normal vector $\vec{N}$ on mesh $S = B\vec{w}$
  $\hat{I} \leftarrow$ Render current mesh with Eq. 3.15
  $d = \|I - \hat{I}\|^2$
  while $d > threshold$ **do**
    for each visible vertex v_i on the rendered image $\hat{I}$ **do**
      Compute $\vec{\delta}_i = \partial \vec{n}_{iz}$ with respect to $\vec{w}$
    Computer $\nabla E(\vec{w})$ using Eq. 3.21
    Update $\vec{w} = \vec{w} - \alpha \nabla E(\vec{w})$;
    Update shape normals $\vec{N}$
    Update d

Thus, the gradient of Eq. 3.13 can be computed as

$$\nabla E(\vec{w}) = \frac{\partial E}{\partial \vec{w}} = 2(\sum_{j=1}^{M} dI_j(k_d \vec{l}_z \vec{\delta}_j \rho_j) + \eta \vec{w}), \tag{3.21}$$

where M is the number of vertices, dI_j is the pixel difference in gray scale and $\vec{l}_z$ is the z element of the lighting direction.

We optimize Eq. 3.13 with respect to the weight vector $\vec{w}$ iteratively using Algorithm 2.

3.4 Experiments

In our experiments, the testing users' faces are new to our system. Their face models are not used in prior training process. In a data acquisition and initialization stage, a user takes two selfie photos from front and side using an iPhone 5. In order to align the template face to the input images, 68 feature points [82] are first detected and the shape of them is decomposed to estimate the head pose using the method described in Section 3.3.1. The template 3D face is then transformed based on the detected head pose and aligned to the image. The same number of feature points are detected and back

projected to the 3D space to obtain the feature vertices on the 3D template face. Then iterative reconstruction as illustrated in Alorithm 1 is conducted to find the optimal weight vector $\vec{w}$ for reconstructing the global shape of the 3D face model. Once the global fitting process converged, the system automatically continues the lighting parameter estimation process before the detail fitting process. Based on the estimated lighting parameters, the detail refinement process is done iteratively by Algorithm 2. Figure 7 shows the number of iterations against the total fitting energy of 4 subjects. The energy is computed by normalizing the error between the rendered image and the input image in terms of the total number of pixels. Figure 8 visualizes the intermediate result of detail fitting process after 10, 20, 30 and 40 iterations for subject 1.

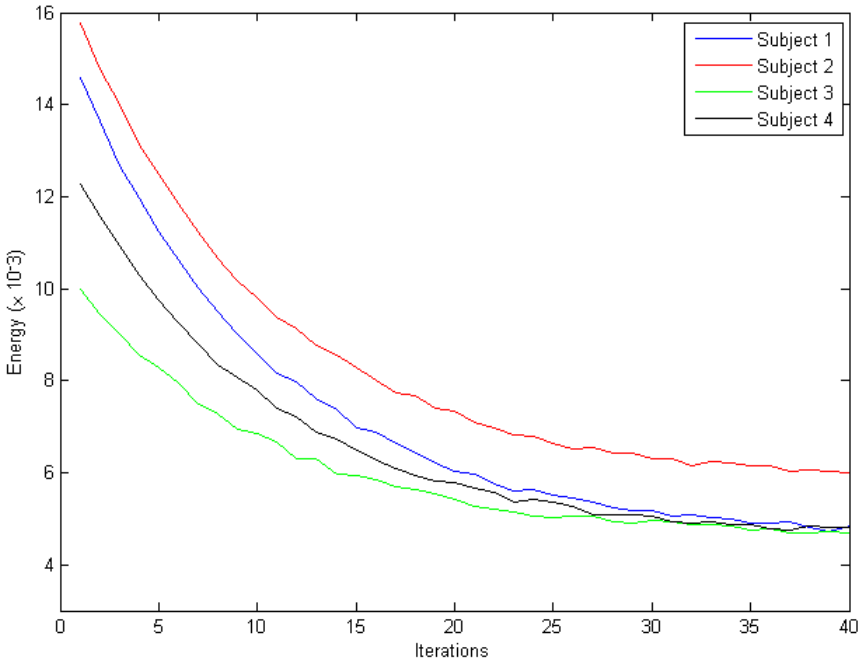


Figure 7: Energy decrease of detail fitting process.

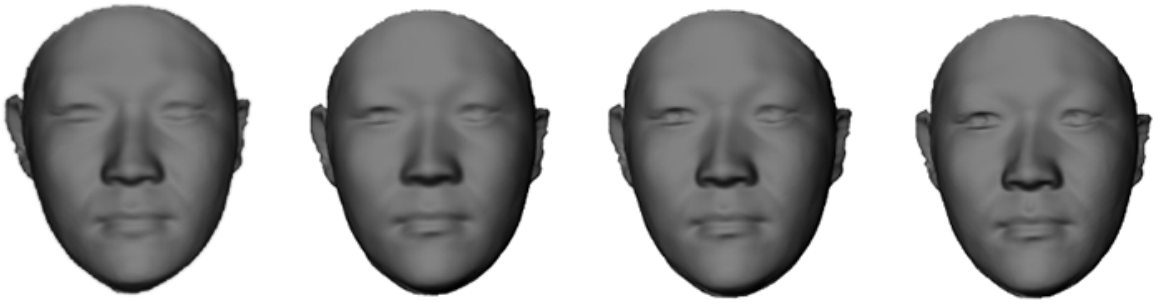


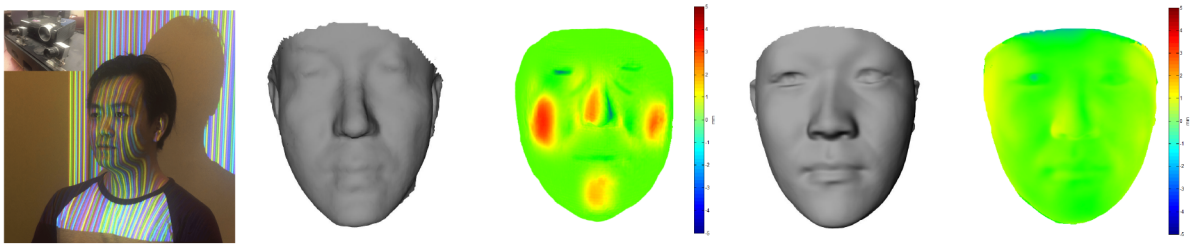
Figure 8: Detail fitting results after 10, 20, 30, 40 iterations respectively.

The textures is finally mapped to the 3D face after the details are refined. We perform our experiment on a regular PC with 3.0GHz Core2 CPU, 8 GB memory and GeForce 9800GT graphics card. The global fitting process takes around 5 iterations which takes about 3 seconds. Lighting parameters estimation takes approximately 300ms and the detailed refinement process takes about 10 seconds.

We compared our NMF based method with PCA based method, and the result is shown in Figure 9. Figure 9 shows the reconstructed result by PCA and NMF respectively without detail fitting process. The more bases used in a reconstruction process, the better reconstruction quality can be obtained for both methods. In practice, both methods used the most significant 105 bases to fit the input image. We found the improvement of the reconstruction is very limited beyond these bases as compared to the increase of computational cost. The result shows our NMF based method can effectively reconstruct major wrinkles on the face while PCA based method fails. NMF decomposes the database by parts whereas PCA decomposes globally, which means NMF is more effective on local detail reconstruction than PCA.

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

Figure 9: Comparison of the model reconstructed by PCA [9] and by our method. (a) shows the input frontal image; (b) is the reconstructed result by PCA bases (without detail reconstruction); (c) is the reconstructed result by NMF bases (without detail reconstruction); (d) shows local details of the result from PCA method (e) shows the local details of the result from our method.



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

Figure 10: Comparison of the fringe pattern scanned model with the results reconstructed using $Kinect$ and our method. (a) shows a groud-truth scan using a fringe pattern scanner; (b) and (c) are the $Kinect$ scanned surface generated by $DynamicFusion$ [56] and its Hausdorff distance map (error map) to the ground truth scan; (d) is the reconstructed result using our method and (e) is the color map of Hausdorff distance to the same ground truth scan.

In order to further show the quality of the reconstructed result, we used a high-resolution fringe pattern scanner to generate a faithful 3D model of the testing subject as shown in Figure 10(a). The comparison among $DynamicFusion$ [56] result and our result to the ground truth point clouds is shown in Figure 10. Our reconstruction result recovers more details than $DynamicFusion$ and reaches a higher accuracy in terms of 3D face modeling and reconstruction.

To further evaluate the effectiveness of our method, we reconstructed 3D face models from synthetic facial images using PCA based method [9] and our method. High resolution 3D models were rendered to generate the synthetic facial images (Figure 11 (a)), which were used as the input to the reconstruction algorithms. Figure 11 (b) and (c) shows the reconstructed results and the error maps using PCA based method and our method respectively. We computed the error maps for both results using the scanned 3D model as the ground truth, which show our method has smaller error compared to the PCA based method. From Figure 11 (d), we can confirm our method is more powerful in reconstructing local details. Table 1 shows the quantitative comparison between two methods. Our proposed method achieves smaller Mean Square Error and Standard Deviation than the PCA based method while keeping similar runtime performance.

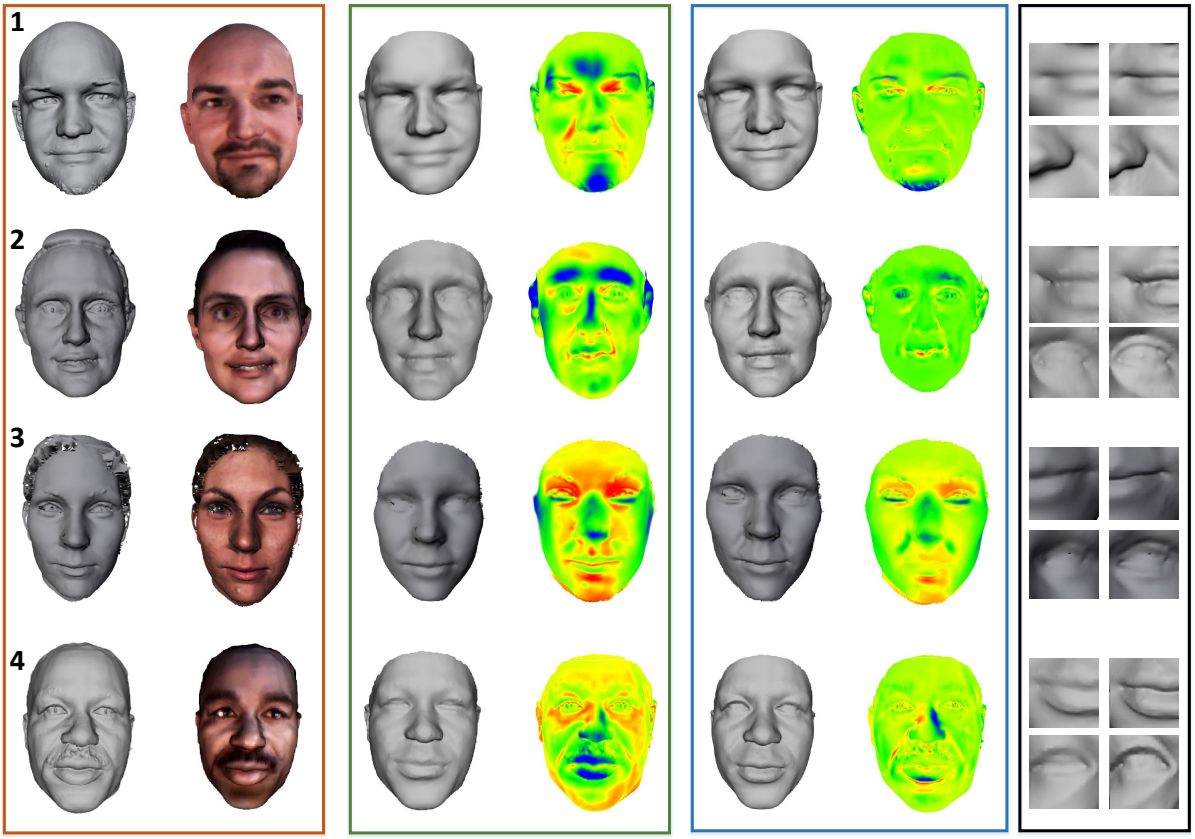


Figure 11: Comparison of the model reconstructed by PCA and by our method. (a) shows the scanned 3D face models and the rendered image of them; (b) is the reconstructed result by PCA bases (without detail fitting) and its error map; (c) is the reconstructed result by NMF bases (without detail fitting) and its error map; (d) left column shows local details of the result from PCA method and (d) right column shows the local details of the result from our method.

	PCA			NMF		
Subject	MSE ($\times 10^{-1}$)	σ_e	runtime	MSE ($\times 10^{-1}$)	σ_e	runtime
1	4.89	0.72	3.2s	2.15	0.41	3.3s
2	4.45	0.68	3.5s	1.68	0.38	3.4s
3	5.95	0.85	3.0s	3.14	0.51	2.9s
4	5.21	0.76	3.3s	1.89	0.45	3.5s

Table 1: Quantitative comparison between PCA based method and our method of the results in Figure 11. Table shows of the Mean Square Error (MSE), the Standard Deviation (σ_e) and runtime of the two methods.

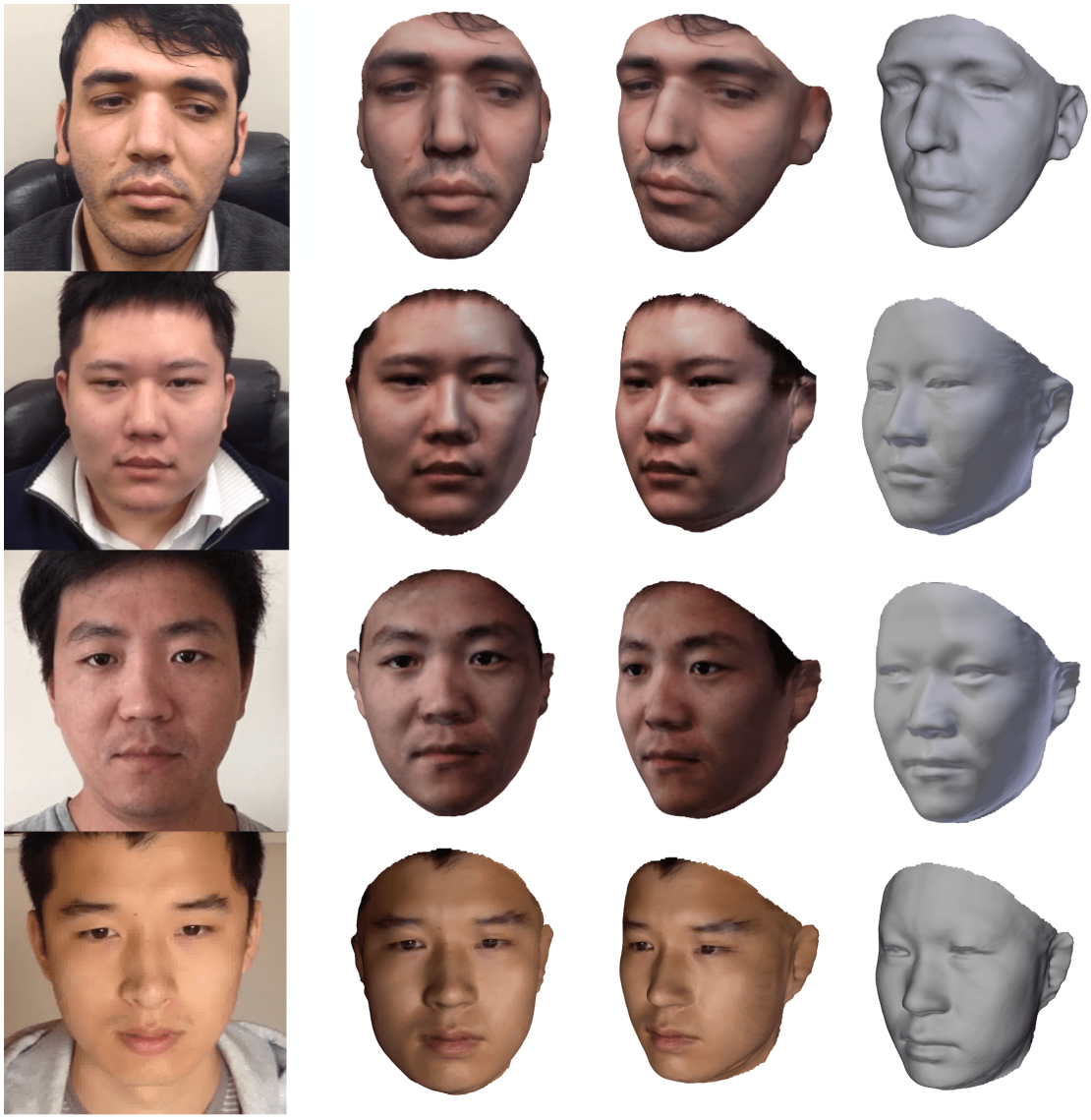


Figure 12: Some reconstructed 3D faces from input frontal and side photos using our method.

Some more results of our method are shown in Figure 12. Figure 13 shows another example, where the two input images are downloaded from Internet. The reconstructed results are shown with and without texture map respectively.

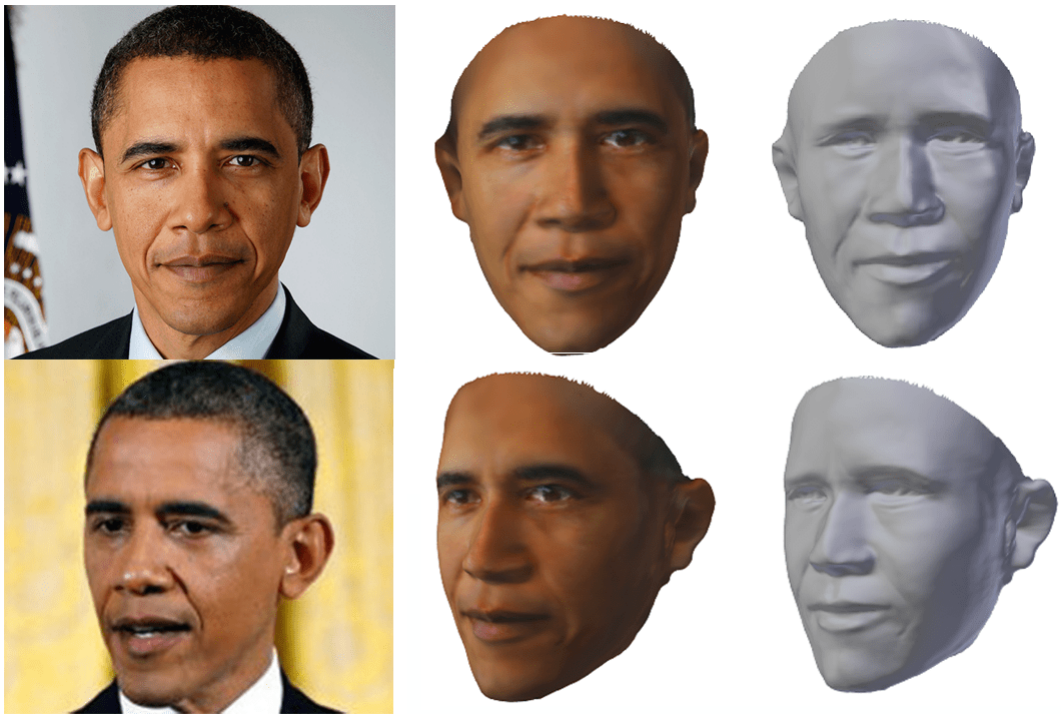


Figure 13: A reconstructed 3D face from two photos downloaded from Internet.

CHAPTER 4 EMOTION INFORMATION VISUALIZATION

4.1 Introduction

In this section, we present a novel approach to analyze the facial expressions from images through learning of a 3D morphable face model and a quantitative information visualization scheme for exploring this type of visual data. More specifically, a 3D face database with various facial expressions is employed to build a NMF part-based morphable 3D face model. From an input image, a 3D face with expression can be reconstructed iteratively by using the NMF morphable 3D face model as a priori knowledge, from which basis parameters and a displacement map are extracted as features for facial emotion analysis and visualization. Based upon the features, two Support Vector Regressions (SVRs) are trained to determine the fuzzy Valence-Arousal (VA) values to quantify the emotions. The continuously changing emotion status can be intuitively analyzed by visualizing the VA values in VA-space. Our emotion analysis and visualization system, based on 3D NMF morphable face model, detects expressions robustly from various head poses, face sizes and lighting conditions, and is fully automatic to compute the VA values from images or a sequence of video with various facial expressions. To evaluate our novel method, we test our system on publicly available databases and evaluate the emotion analysis and visualization results. We also apply our method to quantifying emotion changes during motivational interviews. These experiments and applications demonstrate the effectiveness and accuracy of our method. Fig.14 illustrates our interactive 3D emotion query and visualization system.

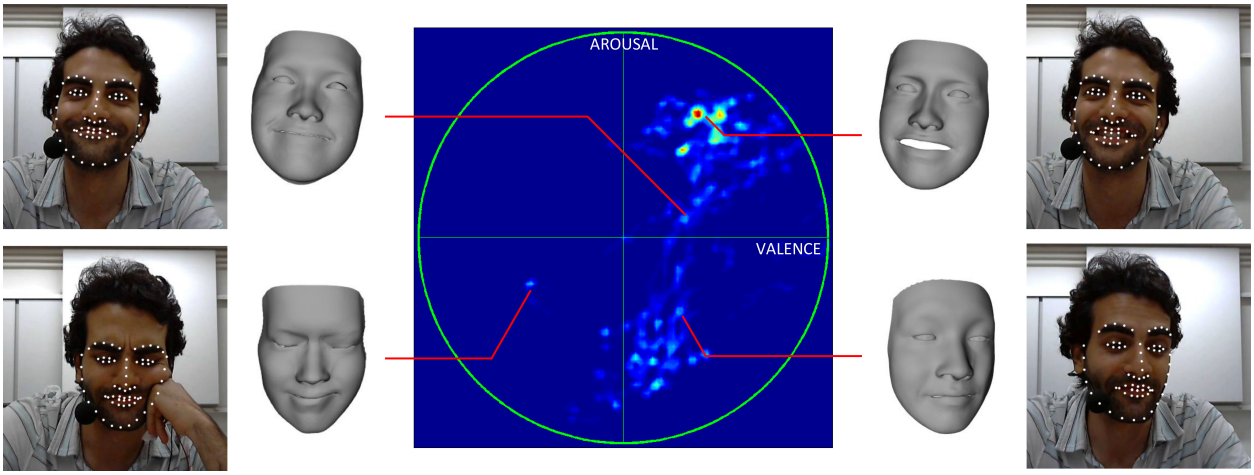


Figure 14: Interactive 3D emotion query and visualization in Valence-Arousal (VA) space. The images are the input frames from videos with extracted feature landmarks (as shown in white points). The corresponding 3D faces are reconstructed 3D models.

4.2 Related Work

Facial expression is the most direct reflection of emotion and shares common meanings across different races and cultures. According to Ekman and Friesen's (1975) study of human facial expressions, there are so called 'universal facial expressions' which represent those common emotions of people: happiness, anger, sadness, fear, surprise and disgust. This study justified the emotion recognition through facial expressions. Ekman and Friensen proposed a Facial Action Coding System (FACS) to describe facial expressions in a standard measurement, which is widely used in image-based emotion classification methods [22].

Extracting 2D features, such as displacement of feature points and intensity change from images, for emotions estimation is the most popular method. For example, Kwang-Eun Ko *et al.* [40] used Active Shape Model (ASM) to extract facial geometry features to classify emotions. Kobayashi *et al.* [41] and Valstar *et al.* [72] used the 20 feature points

on the face for emotion classification. Some work also used intensities around the feature points to enhance the features for predicting emotions. For example, Kapoor *et al.* [36] used pixel intensity difference to classify the emotions of human subjects. However, this method is heavily dependent on the image quality. There are also some hybrid methods that combine the geometry features and the pixel intensity features to estimate the emotions. Developed from ASM, Active Appearance Model (AAM) [19] fits the facial image with not only geometric feature points, but also the pixel intensities. Lucey *et al.* [51] showed the capability of AAM-based emotion detection method using Cohn-Kanade Dataset. Although 2D features are easy to extract from the images directly, they are unstable under the change of illumination or face pose as shown in Sandbach *et al.* [59]. Therefore, 3D features like curvature, volume and displacement are used in many 3D-based approaches. These 3D features are more stable and robust than 2D features since they are pose and illumination invariant in nature. Huang *et al.* [63] extracted the Bézier volume change as the features of the emotions, and Fanelli *et al.* [24] used the depth information of pixels to classify emotions.

More recently, 3D face modeling from images has made a significant progress, which provides new ideas for emotion analysis [9, 6, 7]. Lei *et al.* [45] presented a face shape recovery method using a single reference 3D face model. Prior knowledge of face is required for these methods to recover the missing depth information from a 2D image. Learning through a face database is another effective approach for tackling this problem. Along this direction, statistical face models based on Principal Component Analysis (PCA) were proposed and constructed [9]. Similar face fitting methods, such as piecewise PCA submodels [69], were proposed later. Since these approaches based on 2D images achieved

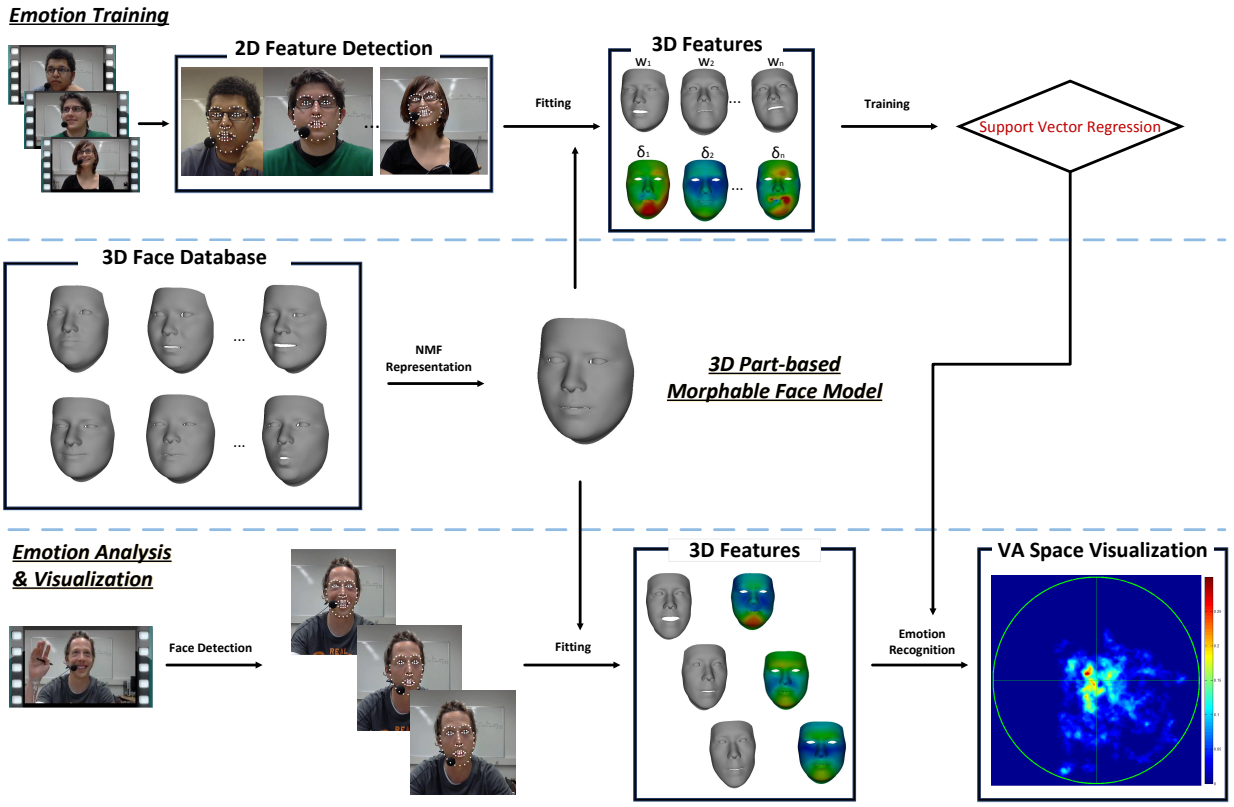


Figure 15: The pipeline of our emotion analysis and visualization framework with learning of 3D Morphable Face Model. Based on a 3D face database, a 3D part-based morphable face model is built. Using the 3D morphable face model as prior, the 3D face is reconstructed for each input video frame and the coefficient vectors and the displacement maps are obtained as features. The features are used to train a Support Vector Regression (SVR) in the emotion training phase. Based on the trained SVR and the 3D morphable model, the emotion VA values are estimated and visualized in the online emotion analysis phase.

plausible 3D face reconstruction results, Blanz *et al.* applied the 3D face reconstruction method to facial recognition problem [10]. However, these methods are still limited to neutral expression and produce poor reconstruction results on faces with expressions.

Therefore, many studies have been devoted to represent 3D faces with arbitrary expressions. Based on a collection of images or a clip of videos of a person, Suwajanakorn *et al.* [68] presented novel 3D face reconstruction technique. A base shape was learned using a template 3D face for the subject, which was then used to fit the images of the same

subject with various expressions. They used a shape-from-shading method to fine tune the details of the shape. The high computational cost is the main drawback, which makes this method not realistic for emotion tracking and analysis. Another popular approach for generating 3D face with an expression at high speed and low cost is Blendshape [34], which has been proposed for fast and robust 3D facial performance animation in industry-level applications. Cao *et al.* [15] proposed a system to animate an avatar face using a single camera and blendshape. Their work focused on tracking a user's facial expressions with a 2D camera and then synthesizing the corresponding expression geometry in an avatar [14, 17]. But reconstructing 3D models from images for emotion analysis is rarely studied and its capability in capturing discriminative features are under-explored.

On the other hand, emotion is rarely visualized in an intuitive way. Visualizing emotion information and status is another interesting topic in emotion analysis. The most general measurement of emotion is in Valence-Arousal space (VA space) [57]. Generally, valence value indicates the pleasantness in emotions from negative to positive, while arousal value evaluates the intensity of the emotions: from calm, peaceful to alert and exciting. The universal expressions can be translated into this two dimensional plotting system, which is clear and intuitive to users perception.

In this section, we present a novel robust approach that measures and visualizes the emotion status continuously in VA space. We use a 3D facial expression database to build a 3D part-based morphable face model which can be used to reconstruct 3D faces from input facial images. Then, we decompose the reconstructed 3D face to obtain its coefficient vector as well as displacement map for emotion quantification. Finally, we demonstrate the continuous emotion change by visualizing the emotion measurement in VA space. Fi-

gure 15 illustrates the entire process, which is fully automated without users' interventions.

This section is organized as follows: In Section 4.4, we present the training process of expression classification with Support Vector Regression (SVR), and show the details of testing new images for emotion analysis using our method. In Section 4.5, we explain our visualization method for the continuous emotion measurement. Finally, wo demonstrate our experimental results and an example application in Section 4.6.

4.3 Feature Extraction From 3D NMF Face Model

After we reconstruct the 3D face model from the input monocular image, we are ready to extract the 3D features from it. As our reconstructed 3D face model S_n is represented by $B\mathbf{w}_n$, where the corresponding weight vector $\mathbf{w}_n$ carries the essential information of the shape, it can be used as a part of the feature vector. Since we use part-based decomposition method to model the 3D face, the weight vector $\mathbf{w}_n$ contains localized feature coding information. Another advantage of using the weight vector is that the 3D model decomposition shares the same basis B, thus, all fitted models are naturally normalized, which makes the features robust for classification. In this paper, we take the first 200 dimension of the basis to represent the 3D face, which means the dimension of weight vector $\mathbf{w}_n$ is 200. Along with the weight vector, we also combine the displacement of vertices to the feature vector. Note that, as we use the NMF part-based model to keep reconstructing all the frames, the reconstructed model has point-to-point correspondence. For each subject we compute the spacial displacement as $\boldsymbol{\delta}_n = V_n - V_0$, where V_n is the reconstructed shape and V_0 is the neutral expression of the same subject which is manually selected as reference. We down sample $\boldsymbol{\delta}_n$ to 300 dimension to reduce the feature dimension in our

experiment. We combine the weight vector $\mathbf{w}_n$ and the displacement vector $\boldsymbol{\delta}_n$ as the final feature vector $\boldsymbol{f}_n$.

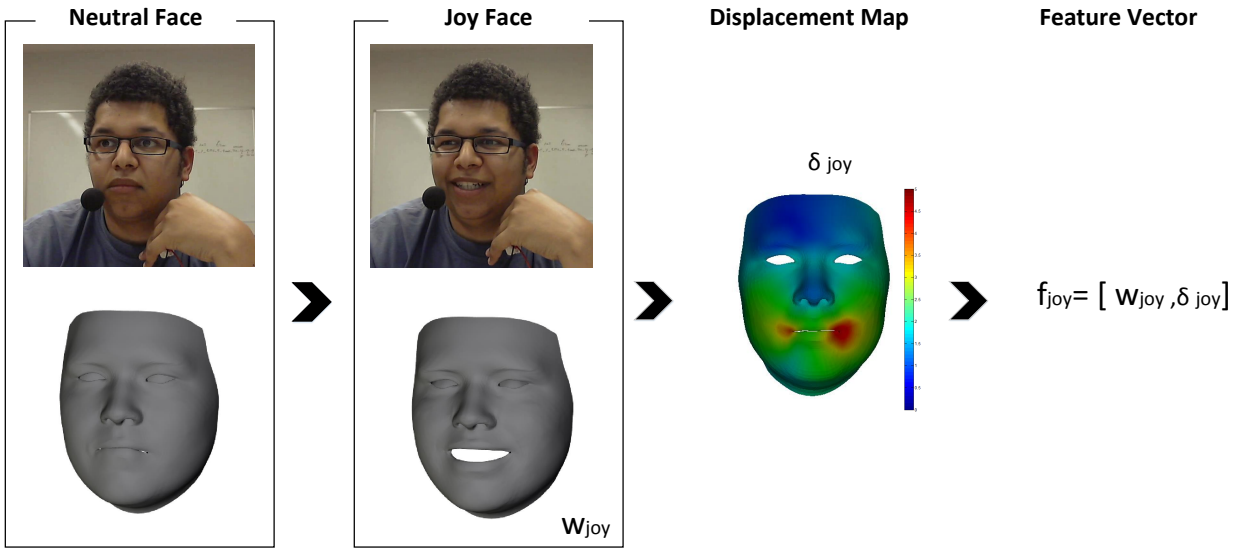


Figure 16: Illustration of feature vector construction for a joy face: the spacial displacement $\boldsymbol{\delta}_{joy}$ is computed between the joy face and neutral face, so the feature vector $\boldsymbol{f}_{joy}$ is composed by shape coefficient $\mathbf{w}_{joy}$ and displacement map $\boldsymbol{\delta}_{joy}$.

Figure 16 shows an example for constructing the feature vector $\boldsymbol{f}_{joy}$ for the joy expression of a subject, which is composed by the weight vector $\mathbf{w}_{joy}$ and the displacement map $\boldsymbol{\delta}_{joy}$. For the visualization purpose, we only show the absolute distance in the color map for the displacement map. Note that, in our algorithm, the displacement is a three dimensional vector containing x, y, z components and stored in a vector form. We use the constructed feature vectors of each video frame to train the support vector regression as described in the next section.

4.4 Emotion Analysis using SVR

We adopt a standard support vector regression as proposed by Valstar *et al.* [71] to establish our 3D NMF morphable model-based emotion analysis method. Our emotion analysis method includes a training step for Valence value regression and Arousal value regression, and then a runtime quantification step for estimating emotion VA values.

Training Data Preparation: Our training algorithm uses the feature vectors constructed from the generated 3D shapes, which is reconstructed from the video frames of the public database. For each video frame I, we obtain a feature vector $\boldsymbol{f}_i = [\mathbf{w}_i, \boldsymbol{\delta}_i]$ along the manually labeled valence/arousal values V_i and A_i to form a training tuple $(\boldsymbol{f}_i, V_i, A_i)$. We use these training tuples to train two $SVRs$ for valence and arousal respectively, namely SVR_V, SVR_A.

To improve the training robustness, we generate some randomness ($\pm 5\%$) to the 3D face reconstruction process to obtain more training datasets: (1) Add a random rotation ΔR to the initial face alignment. (2) Add a random translation ΔT to the initial face alignment. (3) Add a random scaling ΔS to the initial face alignment. Since the 3D reconstruction is sensitive to the initial alignment, by adding these random noise we can obtain more training datasets. In this work, we prepare approximately 100 frames for each dataset by sampling the video at 1fps. Then, we generate one random variation for each case at each video frame, so we have four reconstruction results for each frame and approximately 400 3D models for each dataset. We use 10 datasets with 4000 3D models in each one for SVR training. Feature vectors are extracted from the 3D models using our feature extraction method, as described in Section 4.3, and the training matrix

$M_{training} = \{\boldsymbol{f}_1, \boldsymbol{f}_2 ... \boldsymbol{f}_n\}^T$ is constructed.

Emotion Quantification: The online testing process takes the video frames as the input to our algorithm. We first employ the OpenCV implementation of face detection method with Local Binary Pattern (LBP) [1] to detect the Region of Interest (ROI) for the human face. Within the ROI, the 68 facial landmarks are detected by CLM and the head pose is estimated using the method presented in Section 3.3.1. Then, we reconstruct the 3D face using our NMF part-based morphable 3D face model, from which we extract the feature vector $\boldsymbol{f} = [\mathbf{w}, \boldsymbol{\delta}]$. Feeding the feature vector $\boldsymbol{f}$ to SVR_V and SVR_A, we can obtain the estimated VA values. Finally, the VA values are visualized in the VA space for user analysis.

4.5 Interactive Emotion Visualization

Many emotion detection methods quantify emotions by giving probability scores for the six fundamental expressions. Recent psychological studies shows that quadrants of emotion wheel [28] is more accurate and intuitive than using the six fundamental expressions. Figure 17 shows an example of the emotion wheel, which represents the VA space. All the common emotions including the fundamental ones, can be plotted in the VA space with certain translation rules [31]. As shown in Fig. 17, the first quadrant represents the positive emotions, such as joy and surprise while the third quadrant represents the negative emotions like sadness and disgust.

Our visualization is based on the VA space plotting to reveal the high-level information of the emotion status of the testing subjects from images or videos. We design two types of VA space plotting methods: Emotion Distribution Plotting (EDP) and Emotion Trajectory Plotting (ETP). EDP emphasizes the emotion distributions for the subject during the testing

session, while ETP emphasizes the individual VA value. Figure 19 shows the EDP and Figure 20 shows the ETP for 4 testing subjects. The density of each coordinate, d_{xy}, on the EDP is computed as follows,

$$d_{xy} = \frac{\Omega_r(p_{xy})}{N}, \tag{4.1}$$

where $\Omega_r(p_{xy})$ is the number of VA data points within the distance r from coordinate p_{xy}, and N is the total number of VA data points. In this paper, we sample the VA space from -0.5 to 0.5 with a step size of 0.01 and interpolate the intermediate coordinates. Instead of computing the density at every coordinate in the VA space, ETP only compute the density

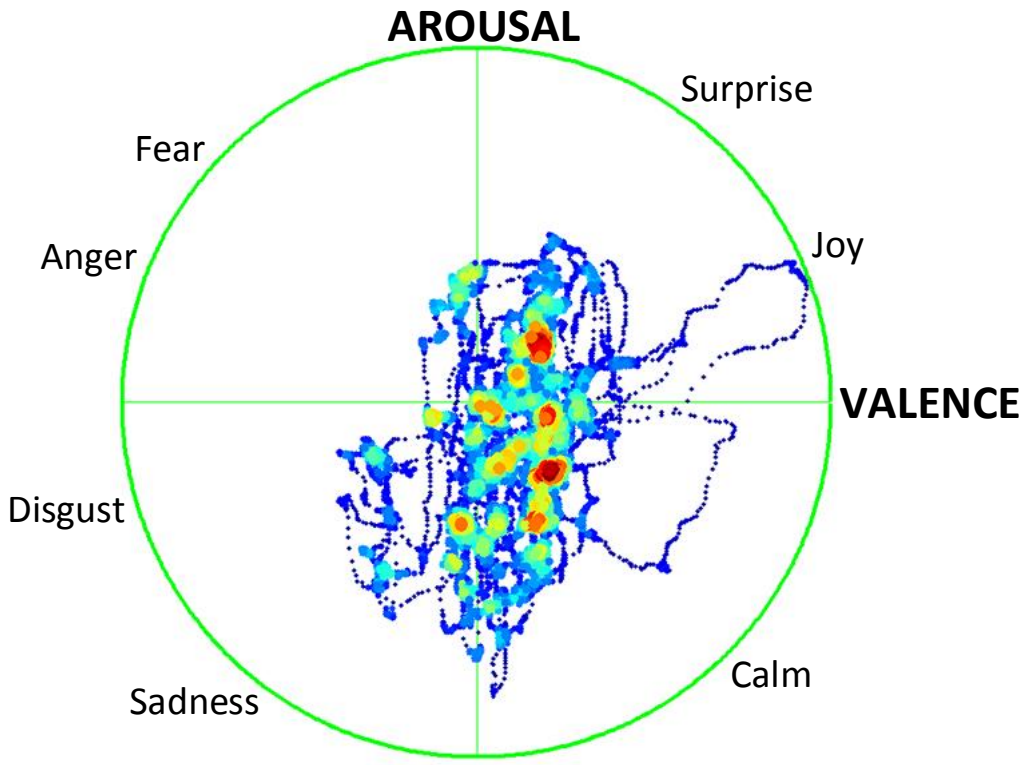


Figure 17: Illustration of the VA space

at each data point. We compute the density at data point p_i by

$$d_i = \frac{\Omega_r(p_i)}{\Omega_r(p)_{max}}, \tag{4.2}$$

where $\Omega_r(p_i)$ is the total number of data points within the vicinity of p_i with radius r and $\Omega_r(p)_{max}$ is the largest number of data points across all such p_i's vicinities with the same radius. The density is converted to color map for final visualization.

To facilitate interactive visual information visualization of emotion data, our EDP and ETP visualization platform supports interactive user query. Details related to a VA value, including the original video frames I, reconstructed 3D faces S, weights w and the corresponding displacement maps $\boldsymbol{\delta}$, can be provided to user by clicking the data point in the plotting. Our system also supports data point comparison which allows users to compare two data points from the plotting by selecting the source and target points. Our system can also provides frames, 3D faces, weights and the displacement map between the two shapes.

4.6 Experiments

We have conducted extensive experiments using public datasets to evaluate the accuracy and effectiveness of our method for emotion analysis and visualization. We also demonstrate some examplar applications using our system. Our experiments have been performed on a regular PC with 3.0 GHz 8 Core CPU, 8 GB memory and GeForce 980 graphics card. The computation time of our method is mainly spent on the 3D face reconstruction process. The reconstruction time is approximately 3 seconds for each frame. We have implemented SVR for regression training and used the trained coefficients in our

C++ implemented system. It takes about 5 seconds to generate the EDP and 3 for ETP with 5000 data points. The query for a data point costs about 3 seconds since we recompute the 3D face model to reduce the memory cost.

4.6.1 Evaluation of the Emotion Analysis and Visualization Method

The Audio/Visual Emotion Challenge Database (AVEC) [62] is a multi-modal dataset for continuous emotion detection using audio and video sequences. In this paper, we have only used the videos in the latest AVEC 2015 dataset for the experiments. Each set of data in the database includes a 5 minutes 1280×720 resolution video with the frame rate of 24 fps, and the Valence and Arousal values are manually annotated for each frame. There are 9 datasets for training, 9 datasets for test and 9 datasets for development use. We have used the AVEC 2015 data for training and evaluated our emotion recognition algorithm on the test data as well as on our in-house recorded data. We have randomly selected 10 datasets including 5 training dataset and 5 development dataset for training purpose and tested on 4 subjects.

To evaluate our emotion analysis method, we first compare it with the expression recognition method using 2D features only. For a fair comparison, both methods use the same SVR. Figure 18 shows the comparison of our 3D model-based method and 2D landmark-based method proposed in [62]. To compare with the ground-truth, we compute the Mean Square Error of the estimated VA values as follows,

$$MSE = \frac{1}{n}\sum_{i=1}^{n}((V_i - \hat{V}_i)^2 + (A_i - \hat{A}_i)^2), \tag{4.3}$$

where V_i, A_i are the estimated VA values and $\hat{V}_i$, $\hat{A}_i$ are the ground truth VA values. Our

3D face model based method achieves lower MSE than 2D landmark-based method [62].

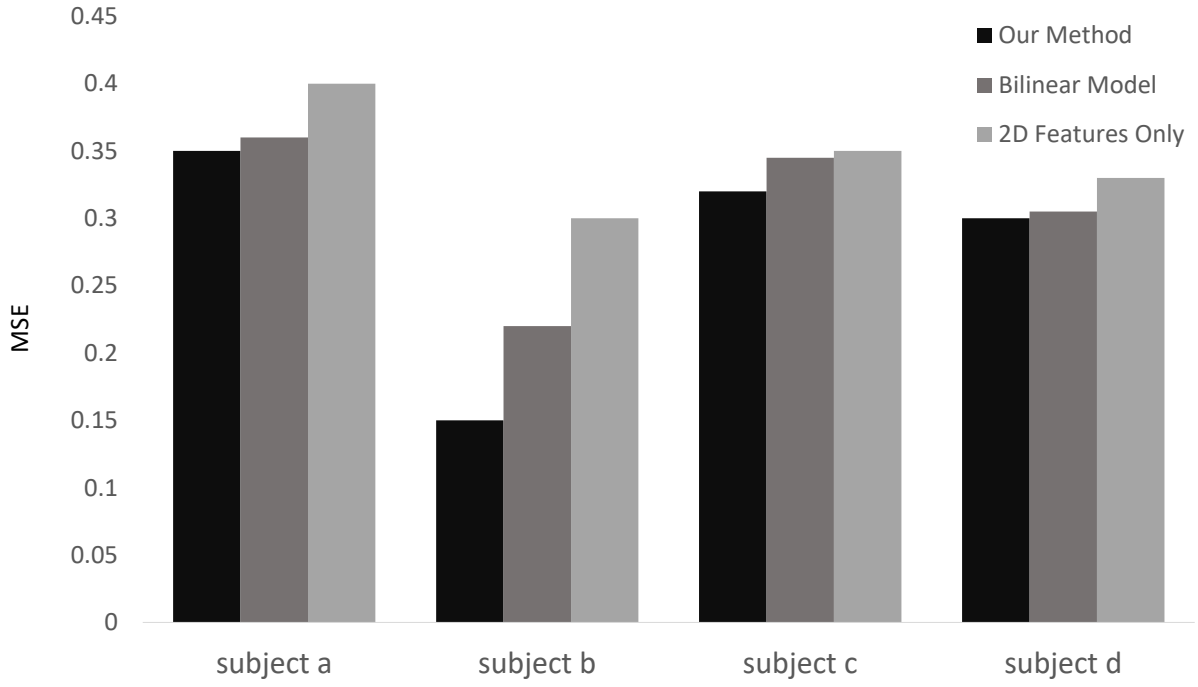


Figure 18: Mean square errors of detected VA values compared with the ground-truth: 1. our 3D feature-based method (NMF), 2. bilinear face model method and 3. the 2D landmark-based method [62].

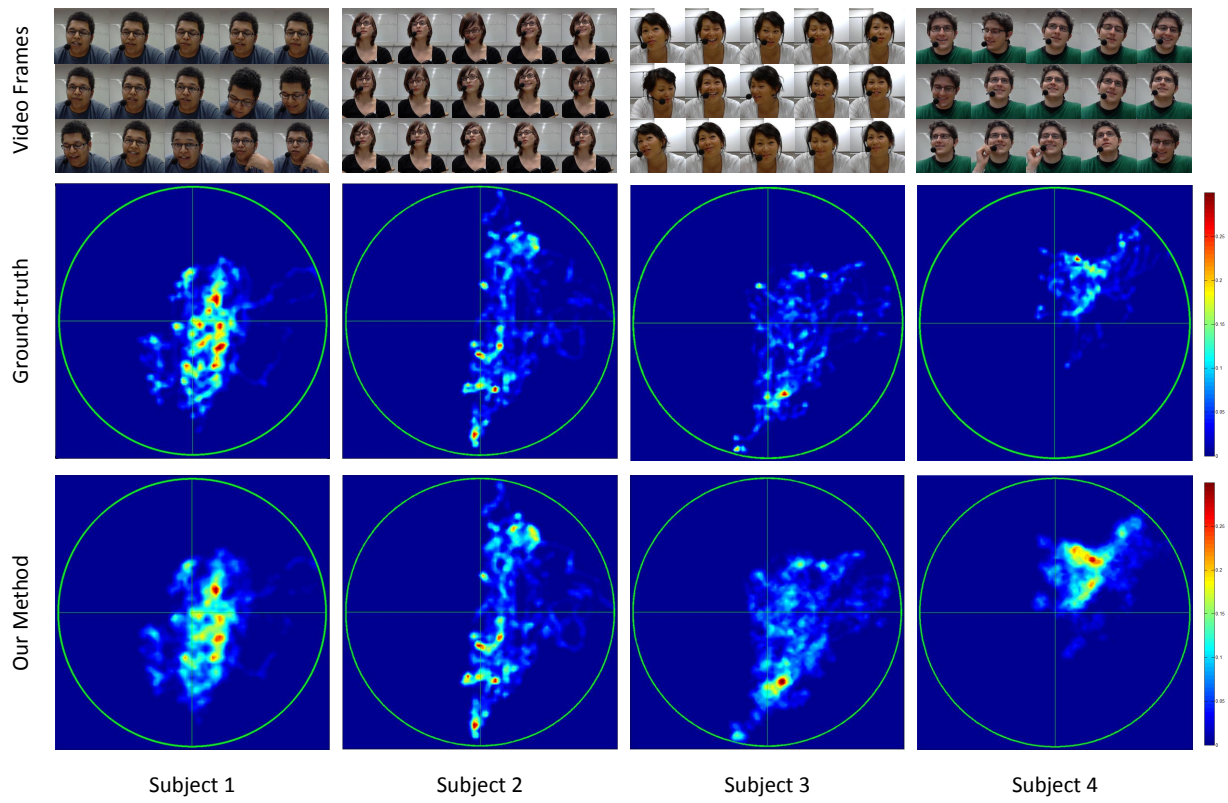


Figure 19: Emotion Distribution Plot (EDP) of VA values for 4 subjects. The top row shows the video frames for subjects 1 to 4 (from left to right). The middle row shows the EDP of the ground-truth VA values, and the bottom row shows the EDP of the estimated VA values using our method.

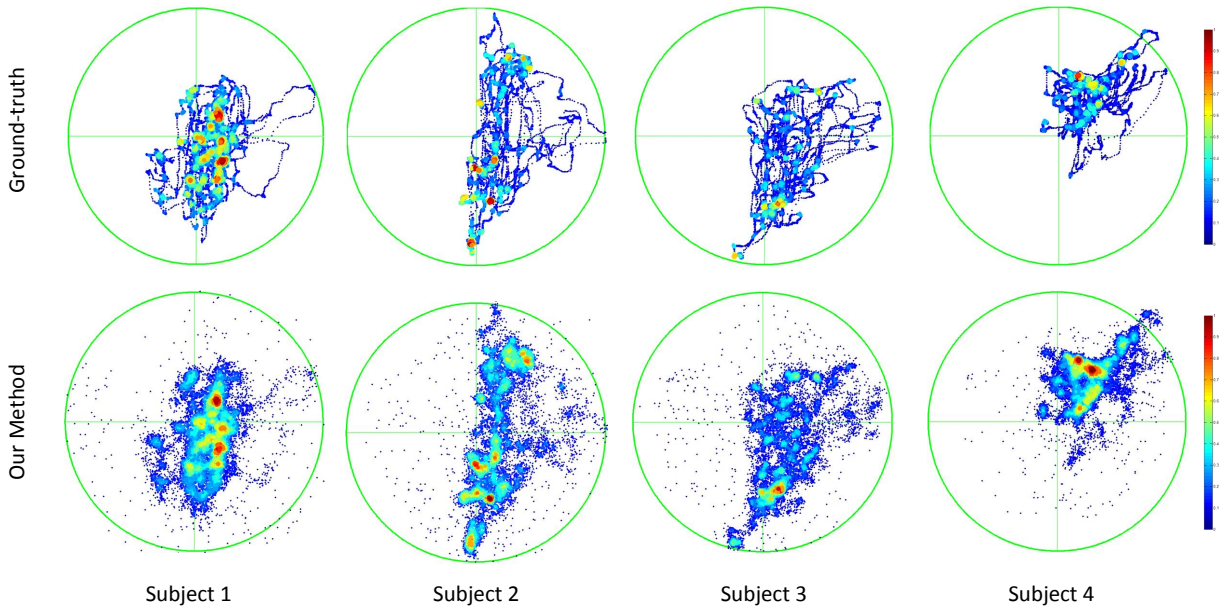


Figure 20: Emotion Trajectory Plot (ETP) of VA values for 4 subjects. The top row shows the ETP of the ground-truth VA values for subjects 1 to 4 (from left to right), and the bottom row shows the ETP of the estimated VA values using our method.

4.6.2 VA Space Analysis and Visualization

Based on the estimated VA values, we use our EDP and ETP for visualization. Figure 19 shows the EDP for 4 subjects from the development datasets in AVEC. The colormap shows the data density in certain location in VA space: red indicates high density while blue indicates low density. The upper row shows the ground-truth VA values plotted in EDP, and the lower row is the estimated VA values using our method. As shown in the result, our method can achieve very similar distribution heatmap to the ground-truth. From the EDP, users can understand the global trending of the subjects' emotions. For instance, the emotion data points of subject 1 are mostly located near the center and the region of 'calm', which means the user is neutral in most of the time during the session. For subject 4, the data is scattered in the positive region, which shows the subject is joyful throughout the session.

For the same datasets, we show the EDP in Figure 20, where the upper row is the ground-truth trajectory and the lower row is the emotion trajectory estimated using our method. The ETP focuses more on the trajectory of emotion changing, so that users can understand the details of the emotion change. For example, for subject 1, there is a clear emotion path from neutral to joy, and then back to neutral, whereas for subject 3, there is an emotion change from neutral to sadness. In order to understand these details in emotion status change, we utilize the emotion status query system which allows users to visualize the underlying information on the data points. By querying the specific data points in the trajectory, the users may understand when the change happens and the subject's condition. For subject 3, we execute two queries for illustration. We first execute

one query to check the status of an upper right data point as shown in Figure 21. The query returns the detected facial feature points, reconstructed 3D face, weight vector and the displacement map for a joy emotion. Since the data point is far from the neutral state, the displacement map shows high intensity around mouth and eyes. We are also provided with a frame ID, with which we could locate the video and see what is the actual situation causing this emotion. Query 2 (Figure 22) illustrates the emotion comparison of two data points by showing both frames side by side. We select two data points (in red and orange respectively) to see the emotion differences. The red data point shows a near neutral emotion while the orange data point shows a negative emotion. The orange frame has a very different head pose compared to the red frame, which is difficult to compare the two images directly. Using our 3D morphable face model, we reconstruct two 3D faces by deforming the template face, which gives us a dense correspondence among the 3D faces. Using the dense correspondence of the two 3D face, we can compute the shape difference between two frames. As the two frames are very close in VA space, the displacement map shows small values. Figure 23 shows a query on a large rotation of the head. Our method can handle different poses effectively since our method is invariant to the head poses. Also, based on the query information, it is possible to correct the misclassified data manually by relabeling or re-extracting the 3D features to improve the accuracy.

4.6.3 Application to Motivational Interview

We have also applied our method to quantifying the effectiveness of motivational interviews. We have used the videos of patients who were interviewed by professional counselors. The beginning phase of the interview is compared with the final phase in order to

quantitatively measure and visualize the effectiveness of this interview process. Figure 24 shows that the emotion of a subject mostly concentrates on calm and neutral status at the beginning phase (Figure 24(a)) while exhibiting joyful status at the final phase (Figure 24(b)), which indicates an effective therapy. The outcome can be quantitatively computed as the average VA value improvement. Figure 25 shows another case, where the emotion of the subject mostly stays in calm and neutral at the beginning (Figure 25(a)) and is improved a little bit at the end (Figure 25(b)). The outcome is not as good as the case in Figure 24. Our method provides a viable tool for the doctors to quantitatively evaluate the patients' emotion status.

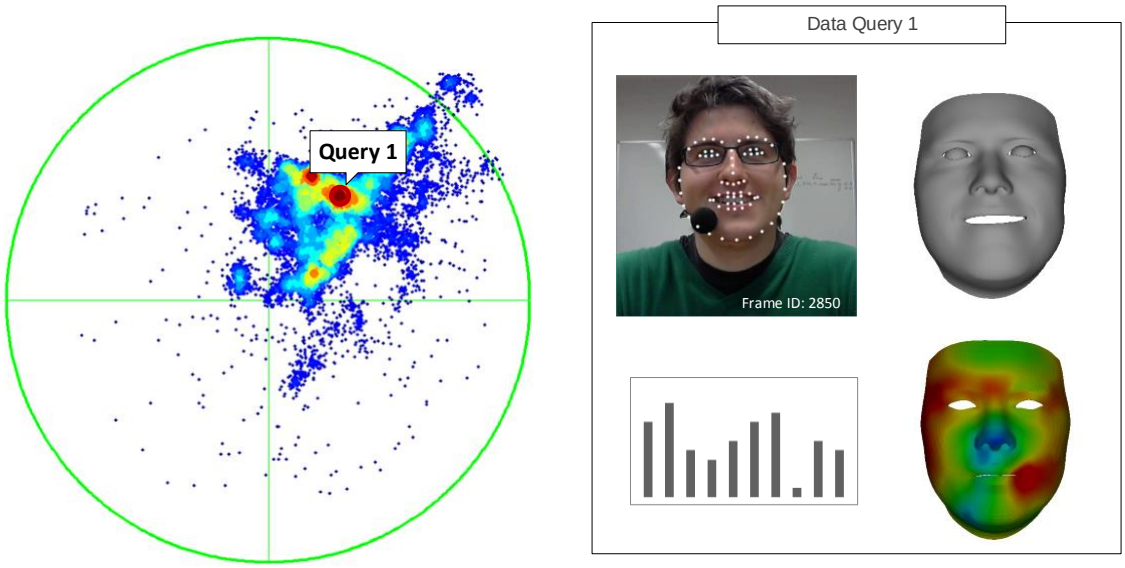


Figure 21: Querying and visualizing a data point in VA space. The query shows the original frame with detected face landmarks, reconstructed 3D shape, weight vector and the displacement map of the data point. The displacement map shows a high intensity for a smiling face.

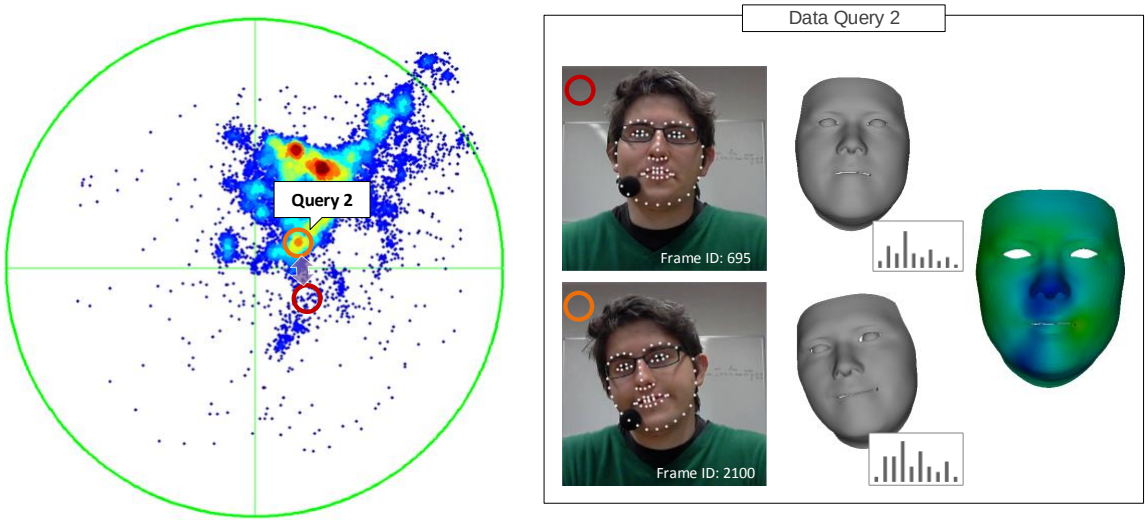


Figure 22: Selection of two data points. The red data point shows a near neutral emotion and the orange data point shows a positive emotion. The displacement map shows a small intensity as the two data points are close to each other despite of the rotation of the head in orange frame.

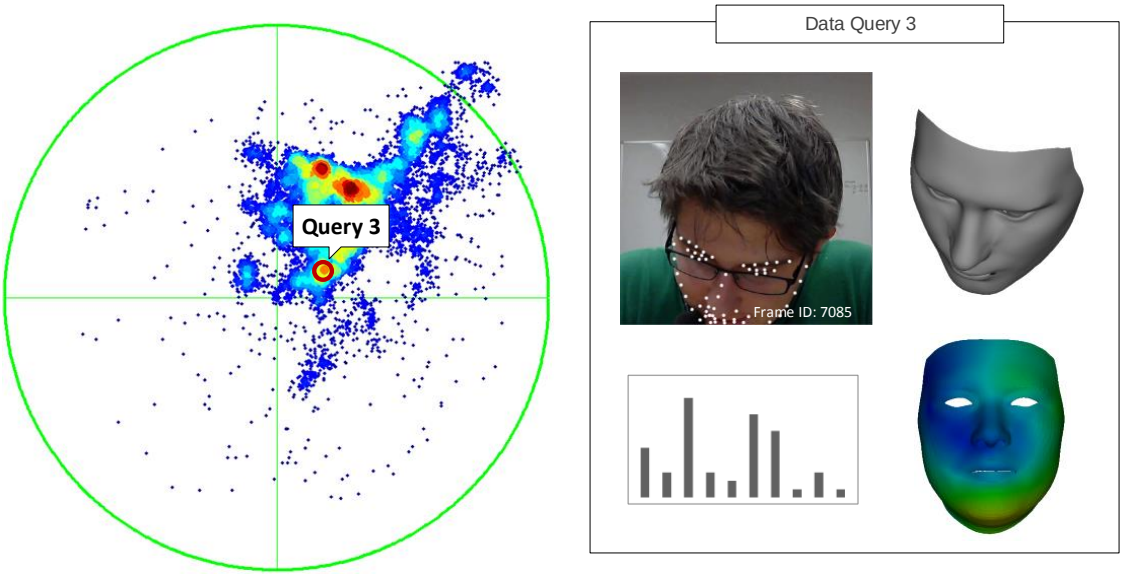


Figure 23: A query example on a large head rotation data point. Our method provides correct emotion estimation with the extreme head rotation.

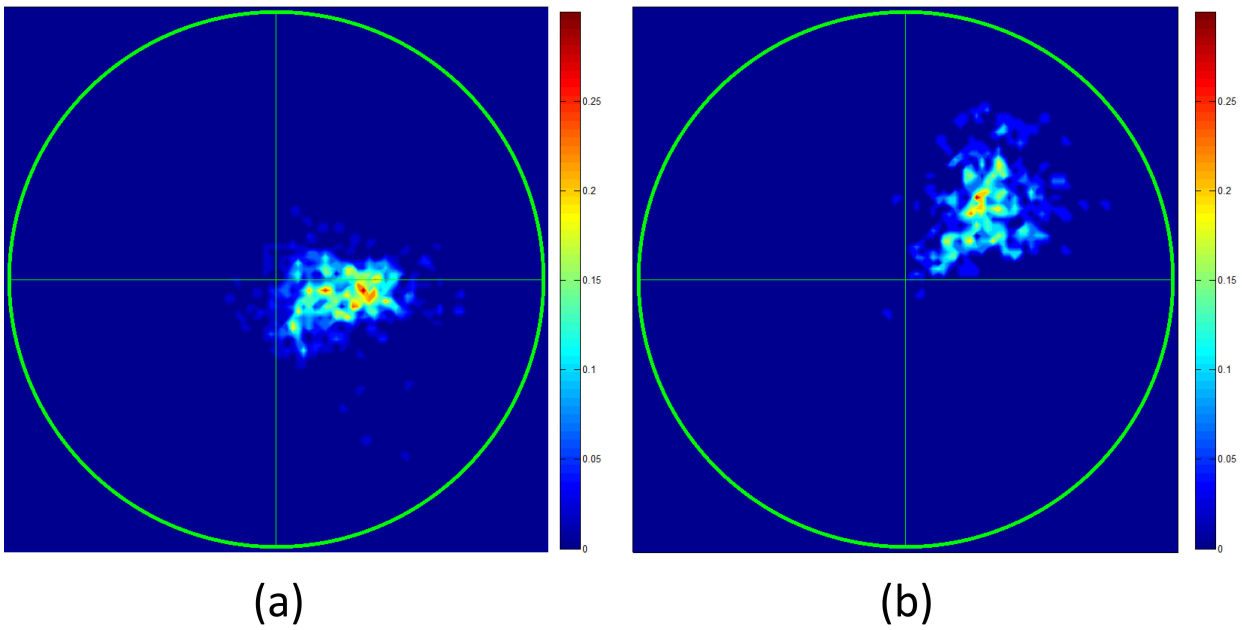


Figure 24: EDP visualization of a subject's emotions during a motivational interview. (a): beginning phase; (b): final phase.

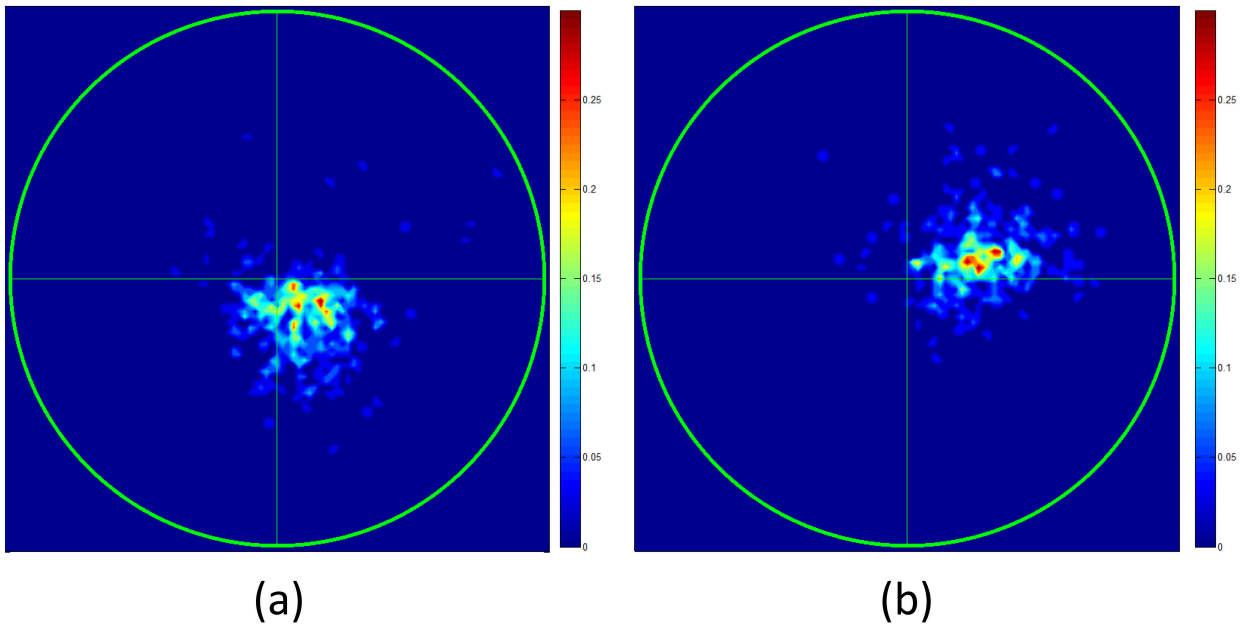


Figure 25: EDP visualization of another subject's emotions during a motivational interview. (a): beginning phase; (b): final phase.

CHAPTER 5 VISUAL ANALYTICS OF FACIAL EXPRESSIONS

5.1 Introduction

The previous sections provide a method to reconstruct 3D face models from images and a method to analyze the emotion status from the reconstructed 3D faces. In this section, we present a novel facial expression recognition approach with 3D Mesh Convolutional Neural Network (3DMCNN) and a visual analytics guided 3DMCNN design and optimization scheme. Our method has achieved a good result for facial expression classification tasks.

From a RGBD camera, we first reconstruct a 3D face model of a subject with facial expressions, and then compute the geometric properties of the surface. Instead of using regular Convolutional Neural Network (CNN) to learn intensities of the facial images, we convolve the geometric properties on the surface of the 3D model using 3DMCNN. We design a geodesic distance-based convolution method to overcome the difficulties raised from the irregular sampling of the face surface mesh. We further present an interactive visual analytics for the purpose of designing and modifying the networks to analyze the learned features and cluster similar nodes in 3DMCNN. By modifying low activity nodes in the network, the performance of the network is greatly improved. We compare our method with the regular CNN-based method by interactively visualizing each layer of the networks and analyze the effectiveness of our method by studying representative cases. Testing on public datasets, our method achieves a higher recognition accuracy than traditional image-based CNN and other 3D CNNs. The proposed framework, including 3DMCNN and interactive visual analytics of the CNN, can be extended to other applications.

5.2 Related Work

With the recent development of high performance deep learning techniques, such as Convolutional Neural Network (CNNs), image recognition accuracy has been greatly improved [42, 61]. Applying CNNs to facial expression recognition problem, Kim *et al.* [38] employed multiple CNNs to obtain a group of diverse models with various properties. Mollahosseini *et al.* [54] trained a single CNN based on multiple naturalistic datasets to obtain a high performance model across datasets. Zhang *et al.* [79] presented a method for inferring social relation from face images using CNNs. They presented a pairwise-face reasoning for relation prediction based on the subjects' age, gender, expression and head pose. Lopes *et al.* [50] proposed a method which combines CNNs and pre-processing techniques to reduce the data required for CNN training.

As we discussed in previous sections, the 3D features are pose and illumination invariant, therefore, they are more stable and consistent than the 2D features in different circumstances. To extend the high performance CNN framework to the 3D model classification task, Sinha *et al.* [65] proposed a 3D shape learning method using geometric images. Su *et al.* [66] used multi-view rendering method to render a 3D shape to a number of rendered image series from different angles and Wu *et al.* [76] worked on the volumetric shapes for deep learning. To simplify the deep learning framework, these methods transfers 3D shape to uniform square domain or cubic domain instead of computing directly on the shape surface. This is a straitforward solution for 3D shape classification with CNNs, especially learning various different shapes. However, the 3D shapes in the facial expression recognition task are all 3D facial models, which provide the possibility of normalizing

them to a standard facial area domain. Utilizing this property, we propose a 3D Mesh Convolutional Neural Network for facial expressions on the 3D facial surfaces.

To better understand the learned features of the network, Zeiler *et al.* [78] proposed a CNN visualization method for diagnostic purpose and Liu *et al.* [49] presented a directed acyclic graph-based CNN visualization method to obtain the overview of the CNNs. More recently, Pezzotti *et al.* [58] presented a progressive visual analytics approach for designing CNNs and successfully optimized the public large scale CNNs using the insights obtained from their system. Kahng *et al.* [35] presented a method to visually explore industry-scale Deep Neural Networks.

In this section, we present a robust approach for facial expression recognition based on 3DMCNN that learns 3D features of the reconstructed face models. We use a 3D facial expression database to fit the scanned depth image of the face to generate a high quality 3D face models with expressions. Then, we compute the geometry signatures, i.e., mean curvature, conformal, factor and heat kernel signature as the features for learning. We perform the learning processes, such as convolution, pooling, rectified linear unit (ReLU) and etc., directly on the 3D surface domain for training the 3DMCNN. Through visual analytics, we can modify and optimize the networks for better performance. Finally, we compare the performance of our method with conventional image-based CNN methods and analyze the advantages of our method by using interactive visualization techniques. Fig. 26 illustrates the entire process.

The rest of the chapter is organized as follows: Section 5.3 introduces the method for reconstructing the 3D face with expression using RGBD camera and deformable face model. In Section 5.4, we explain our 3DMCNN and its geodesic distance-based convolution

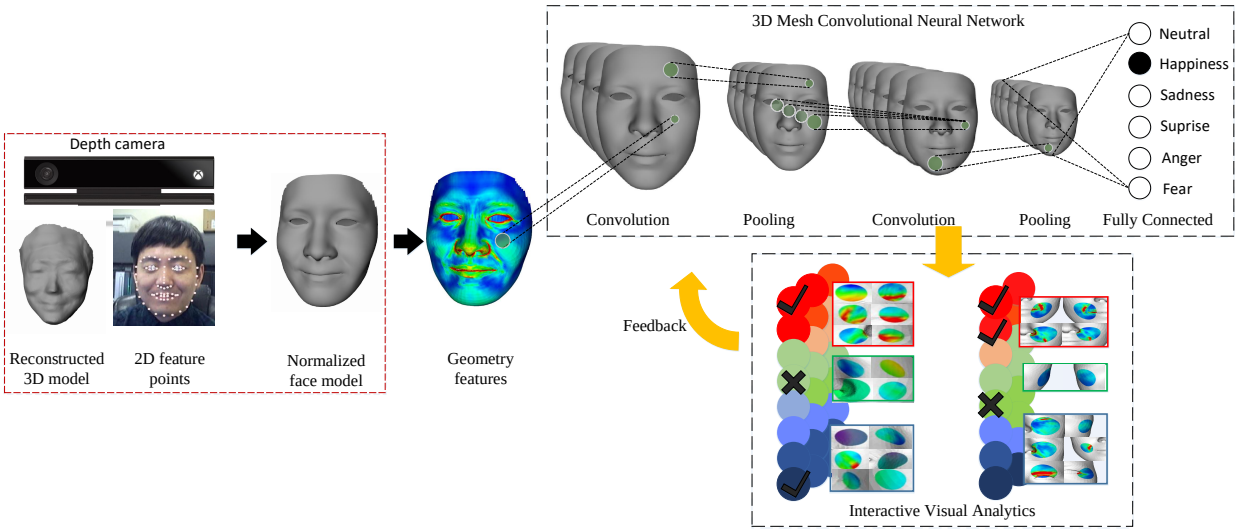


Figure 26: The pipeline of our 3D face model based facial expression recognition framework with 3D Mesh Convlutional Neural Network. Based on the captured depth image and facial landmarks on the color image, a 3D facial expression model is generated by fitting a morphable face model. The geometric signatures of the 3D facial model are computed and used for training a 3DMCNN for classify the facial expression.

and pooling framework. Then, the employed geometric descriptors of the 3D face is described. In Section 5.5 introduces the learned feature maps and the clusters of different nodes in our 3DMCNN, which facilitates further editing of the networks based on an interactive visual analytics approach. In Section 5.6, we first show the public database for the experiment and then explain the training and testing details. Finally, we show the results of the our method and compare it to stat-of-the-art 2D image-based CNN methods and other CNN methods for 3D shapes.

5.3 3D Face Reconstruction from RGBD Sensor

We employ a Kinect v2 system which captures 1920x1080 2D images and 512x424 depth maps at 30 frames per second. A raw depth image of the subject's face along with RGB color information is acquired. To improve the quality of the raw 3D facial data and

normalize it to a uniform face data space, we use a multi-dimensional 3D face database, Facewarehouse, as a refinement template. Cao *et al.* [16] presented a 3D facial expression database, which includes 150 subjects with 47 different expressions. The database is decomposed in a bilinear face model and each face model can be computed by

$$S = C \times_2 w_i^T \times_3 w_e^T, \tag{5.1}$$

where C is the decomposed core tensor, w_i is the identity weight vector and w_e is the expression weight vector, respectively. Then, the identity weight w_i and the expression weight w_e can be obtained by minimizing the surface distance between the scanned raw face $\hat{S}$ and the reconstructed bilinear model S as follows:

$$w_i, w_e = \operatorname{argmin} \|\hat{S} - (R(S) + T)\|^2, \tag{5.2}$$

where R is the rotation matrix and T is the translation matrix. In practice, we first estimate the rotation angle and translation using the method [33], then we optimize identity weight w_i, followed by optimizing expression weight w_e. To fine tune the expression of the generated 3D face model, we utilize the RGB color image that is captured simultaneously with the depth map. We use Constrained Local Model (CLM) [21] to find 74 landmarks on the color image. These points include the face contour, eye contour, eyebrow, nose and mouth contour, which are used for the following two purposes. One is that these landmark points can be used for locating the face position in the image and estimating the head rotations. Estimating head rotation is important to initialize the template face model to start

the fitting process. The second is that the points can be used fo fine tune the reconstructed expressions to improve the reconstruction accuracy of certain extreme expressions. Fig. 26 left block illustrates the reconstruction process.

There are two advantages of using the face decomposition approach to reconstruct the 3D faces. First, it provides an easy and low-cost solution to obtain higher resolution 3D face models for expression analysis. Traditional high resolution 3D scanners is expensive and the scanning usually takes a long time. Therefore, capturing a facial expression is a difficult task for these equipments. Meanwhile, although the commercial depth camera can provide a cheap and fast way to scan faces, the reconstructed 3D meshes often have low resolution for computing meaningful geometric features for learning facial expressions. The face decomposition approach can use a pre-scanned database as a prior-knowledge to generate high resolution facial expression 3D models from a low resolution depth scans. Second, it provides a consistent sampling domain across the generated 3D faces. Since the optimized 3D face model is obtained by computing the weight vectors, it is actually a linear combination of the decomposed basis faces. The generated 3D face always has one-to-one correspondence in terms of vertex to the average 3D face of the database $\bar{S}$. Therefore, the sampling points on the 3D faces across the subject and the expressions are consistent. We use the consistent sampling points as the grids to perform CNN computations, which will be explained in detail in Section 5.4.

5.4 3D Mesh Convolutional Neural Network

In this section, we present a 3D Mesh Convolutional Neural Network (3DMCNN) for facial expression recognition by learning facial geometric features. Our method conducts

operations including convolution and pooling by utilizing the mesh grid on a template face model as shown in Fig. 27 (a). The red points are the sampling grid for computation, equivalent to the pixels in 2D CNN. Instead of using regular uniform grid in 2D image CNN, we have denser grids around high activity and curvature regions including eyes, mouth and wrinkles near mouth for higher sampling rate.

5.4.1 Geodesic Distance-based Convolution and Pooling

Convolution: Traditional CNN uses uniform grid for convolution and pooling, which is efficient for processing images. Instead of transforming 3D surfaces to 2D planes for learning, our method directly learns the geometric features on the 3D surface. Since the reconstructed 3D face is a linear combination of the decomposed basis faces, there is one-to-one correspondence in terms of vertices across the face models by nature. These vertices serve as the consistent sampling points on the face domain for CNN computation, which is shown in Fig. 27 (a). The red points are sampling points on the face surface and we compute the convolution on each point.

Similar to the convolution on an image, we use a weighted filters to convolve the geometric signature values. Convolution on images is done by sliding a square filter over the images, however, defining a square filter on 3D surface is difficult. To solve this problem, we propose a geodesic distance-based convolution, where the weights are defined by the geodesic distances. In a continuous form of geodesic distance-based convolution, the weight values are continuous functions to the geodesic distance from the center point. To reduce the computational cost on the discrete meshes, we define limited directions for convolution. On each vertex, we search eight directions, which are East, North East, North,

North West, West, South West, South and South East as shown in Fig. 27 (b). Together with the center point, we denote these directions as D, which is illustrated in Fig. 27 (c). To search for the East direction, we start from the center point and searches over the surface until it reaches the defined geodesic distance. The normal vector of the center vertex is needed to compute the tangent plane and the initial searching directions are defined on the tangent plane. Similarly, we search for other seven directions, and compute the weighted sum around the center point as follows:

$$g_{n+1} = \sum_{d \in D} (w(d,l) g(d,l)_n), \tag{5.3}$$

where $w(d,l)$ is the weights defined by the distance l and direction d, $g(d,l)_n$ is the geometric signature value at the destination location and n indicates the layer of the network. As the geometric signature values are only defined at the sampling points, it is not guaranteed that we can find a value at the destination location. By using the barycentric interpolation of the nearby triangle, we can compute the value $g(g,l)_n$ at any location of the surface.

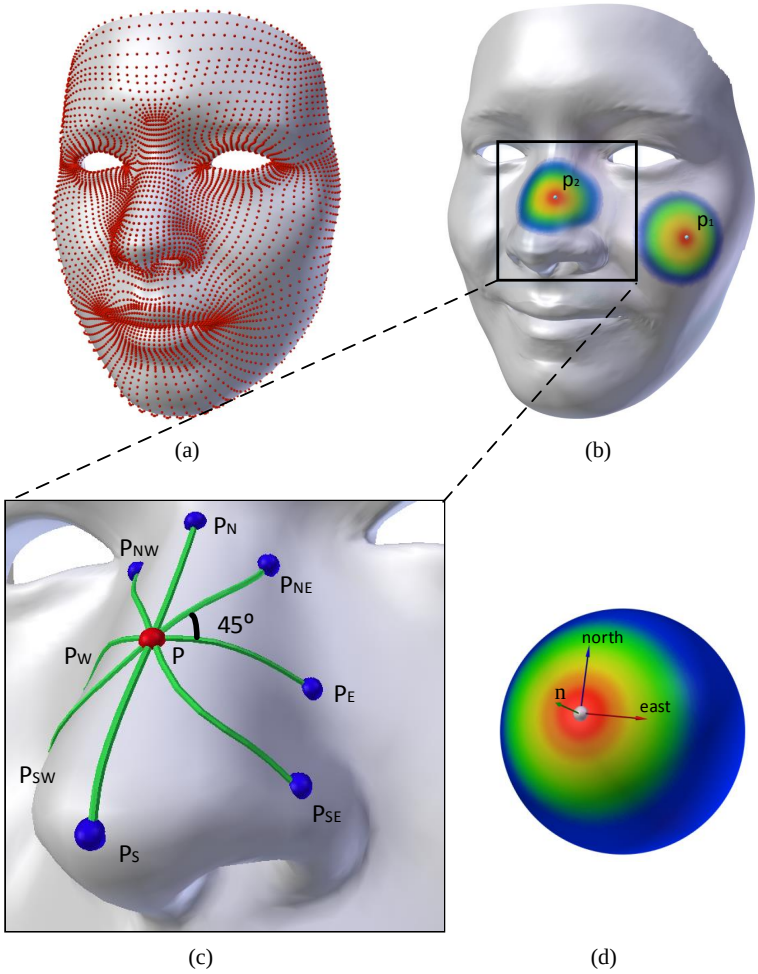


Figure 27: (a) illustrates the sampling points on the face surface. (b) illustrates the continuous geodesic distance rings around the center points. (c) is the discrete 8 directions for geodesic distance-based convolution. (d) shows the normal vector of the sample point for tangent plane computation and the defined directions.

The geodesic distance-based convolution has the advantages of preserving and identifying true features as well as preventing dislocated false features in the convolution space when taking the actual geodesic distance as a-priori information. Fig. 28 shows an illustrative one-dimensional example, where the curve is the shape and the line segment indicates the pixels of the 1D "image" of the curve shape. The sampling vertices V_1, V_2, V_3 and V_4 on the curve are rendered to the pixels P_1, P_3 and P_4 on the 1D "image". As shown in figure, although V_2 has a high curvature value on the curve, it is not rendered on the "image" due to the sampling interval. When the convolution is done on the image plane without considering the geodesic distance, a low curvature value at V_3 will be used for convolution instead of V_2's curvature value. Therefore, the important geometric feature of vertex P_2 will be lost in the image convolution framework, which is not desired. On the other hand, image convolution framework will result in an uneven convolution on the 3D surface domain. For example, the surface patches which are perpendicular to the image plane will be coarsely sampled and the parallel surface will be densely sampled. So using geodesic distance for convolution avoids these problems caused by simply applying image convolution framework to the 3D surfaces and provides more uniform convolution results. This is also evidenced by our later experiments in Section. 5.6.

Pooling: Similar to the convolution, we also perform pooling operation based on geodesic distance. Based on the selected geodesic distance, we compute the mean (or max) value of the feature values over a region within a certain geodesic distance around the sampling point. Since the sampling points become sparse after each pooling operation, we re-triangulate the remaining sampling points to reconstruct new mesh surface and double the geodesic unit for next layer computation. Once the 3DMCNN architecture is

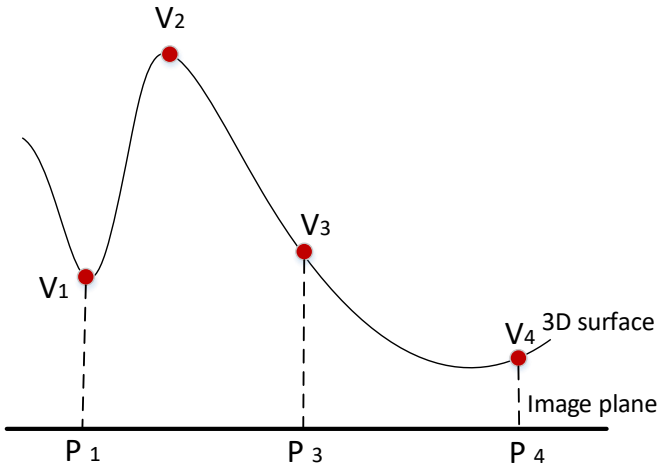


Figure 28: Illustration of the difference of pixel based convolution and the geodesic distance-based convolution. Color shows two rendered pixels P_1 and P_2 from mesh patch around vertex V_1 and V_3. V_2 is not rendered individually due to the resolution.

established, the resampling the 3D faces can be pre-computed before training process. We generate a cascaded face series with the reference face and map to each individual models. The geodesic unit is determined by the average geodesic distance between two sampling points on the shape surface. As we mentioned above, we increase the geodesic unit based on the stride of the pooling operation.

5.4.2 Architecture of 3D Mesh Convolutional Neural Network

The layers of the 3DMCNN is determined by the complexity of the 3D models and the size of the database. Our 3D face model approximately consists of 5000 vertices, and has about 60 $\times$ 80 sampling points on the surface. For each expression, there are 800 models with different identities. Therefore, we only need a relatively shallow architecture for the networks. Our CNN model has three convolution (C1, C2, C3) and pooling layers (P1, P2, P3) and two fully connected layers (FC1, FC2). Specifically, C1 is a convolutional layer

with feature maps connected to a neighboring area within the 2 geodesic units in the input.

The values in C1 are initialized by a uniform distribution with the range depending on the incoming nodes. P1 is a pooling layer with feature maps connected to corresponding feature maps in C1. In our case, we use the max pooling to amplify the most responsive vertex in the feature maps. C3 uses partial connection scheme for keeping the number of connections within proper bounds and breaking symmetry in the network. In this way, we can expect the kernels to update diversely to generate different feature maps.

5.4.3 3D Descriptors for Deep Learning

In this section we discuss the 3D descriptors that are used for the Mesh CNN training. Generally, 3D properties including principal curvatures, mean curvatures, Gaussian curvatures, conformal factors [29] and heat kernels [67], can be used to describe the 3D shapes. As Hua *et al.* [29] indicated that a 3D shape can be uniquely defined if the mean curvatures and the conformal factors are given, in this paper, we use mean curvature, conformal factor and heat kernel as the geometric descriptors of the 3D face model.

Mean Curvature Mean curvature is an extrinsic measure of the curvature at a given location of a surface S. Fig 29 shows the mean curvature on 3D meshes of three examples: an Armadillo, a human brain and a scanned human face. The mean curvature values are normalized to the range from 0 to 1 and mapped to RGB colors for visualization. Regions such as finger tips, brain sulci, eye contour and mouth show red, which have high curvatures. Mean curvature provides the face mesh deformation information for learning the expressions. Mean curvature is a local property which can be computed at each vertex in discrete mesh forms, therefore, the computation can be easily parallelized using GPU.

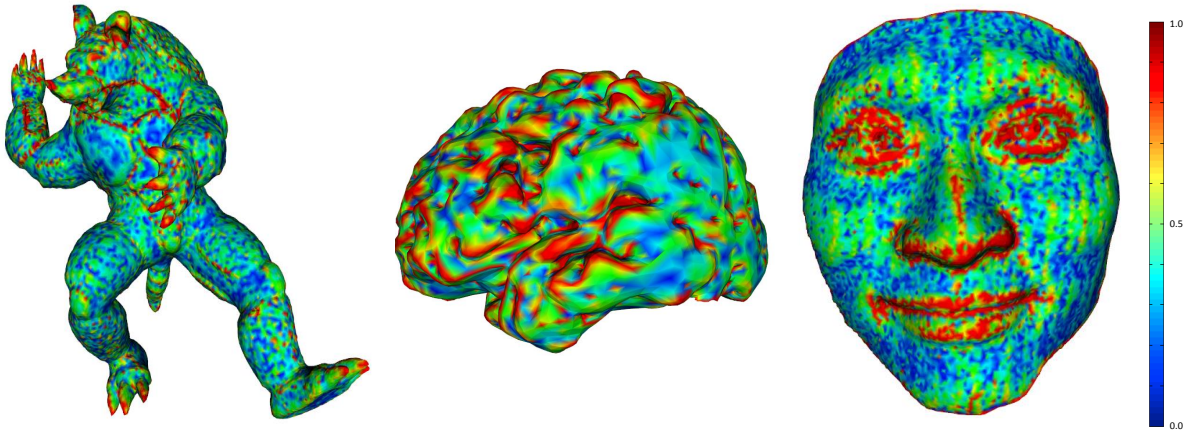
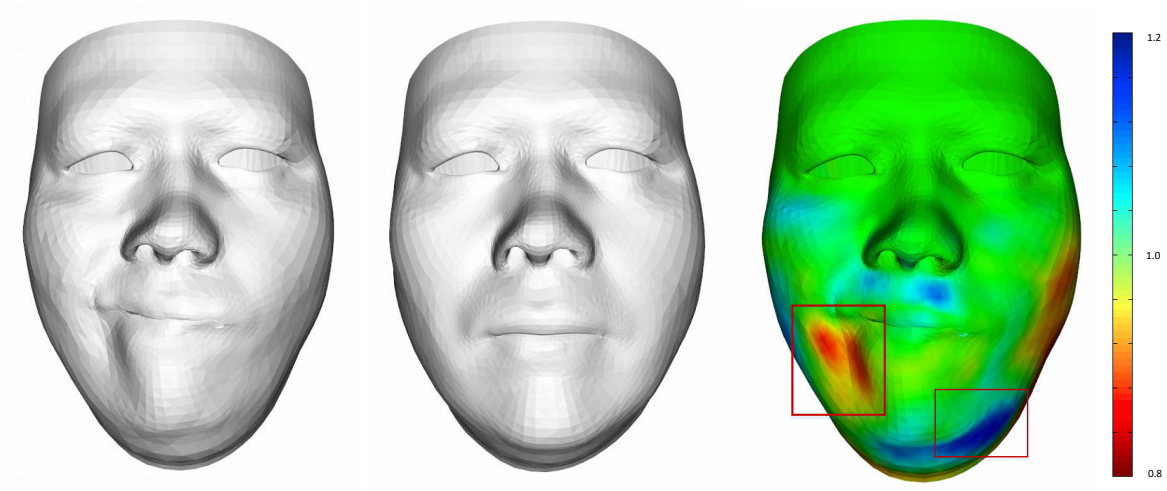


Figure 29: Illustration of mean curvature on three different shapes. The mean curvature values are normalized to [0, 1], and the color map is shown on the right.

Conformal Factor Conformal factor measures the vertex area change of the deformed shape. Fig. 30 shows an example of the conformal factors on the 3D facial model. Fig. 30 (a) is a non-neutral expression. Fig. 30 (b) is the neutral expression face, which is reconstructed once the identity vector W_i is estimated. Based on the computed Voronoi area using Eq. 2.2, the conformal factor $\lambda(p)$ is computed based on least square conformal mapping. The conformal factors are normalized and visualized with color map as shown in Fig. 30 (c). The highlighted areas in Fig. 30 (c) show the significant changes around the mouth, where the main deformations occurred.

Heat Kernel Signature A heat kernel signature (HKS) is a feature descriptor of spectral property of a 3D shape. Fig. 31 illustrates the heat kernel signatures of two expressions. The time variable T increases from left to right, showing the continues change of HKS on the face surfaces. Since HKS is a global feature of the shapes, it is widely used for shape analysis and shape retrieval tasks due to its significant differences between different shapes [12]. Although the HKS is relatively stable on the human face models, we found it



(a) (b) (c)

Figure 30: Illustration of the conformal factor on a facial expression model. (a) is a non-neutral facial expression model. (b) is the estimated neutral expression model after the identity vector W_i is obtained. (c) is the conformal factor map. The two highlighted areas show the area change of the surface.

also changes with the different expressions, especially with the smaller time parameters. To use more significant local differences as the feature while keeping a certain level of the global feature, we experimentally select a small $T = 10$ in our method. Therefore, HKS performs as a supportive feature to the more sensitive mean curvature and conformal factor in the 3DMCNN frame work.

5.5 Visual Analytics of Networks for Modification and Optimization

In this section, we present an interactive visual analytics method of the 3DMCNN for the purpose of network performance. First, to better understand the 3DMCNN, visualizing the learned features of neurons is an effective approach. Liu *et al.* [49] presented a directed acyclic graph-based CNN visualization method, which explores features and network structures by clustering similar neurons and visualizing them in a properly ordered layout. Their method focuses on visualization of the deep CNNs without interactive

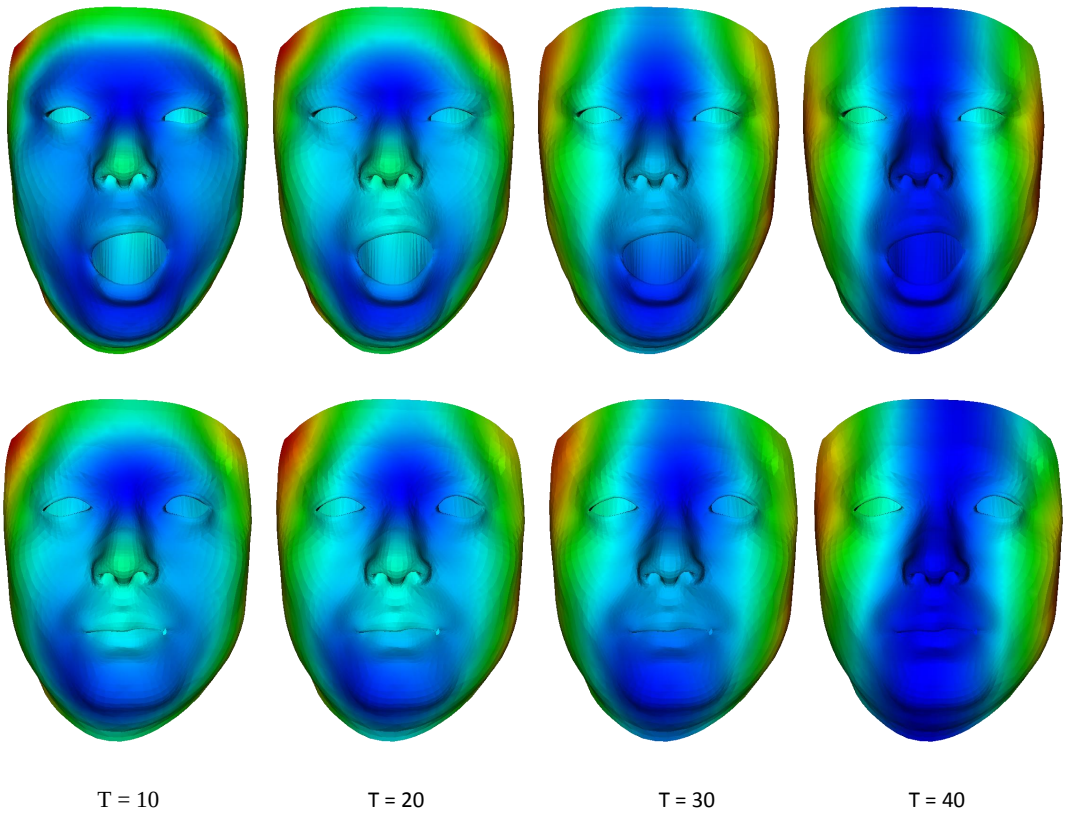


Figure 31: Heat Kernel Signature on two different facial expression models. Left to right shows T = 10, 20, 30 ,40. Upper row shows a sample face with surprise expression and the lower row shows a sample face with neutral expression.

modification of the network.

Inspired by their method, we present an interactive neuron visualization and modification framework, which provides an intuitive and detailed information for understanding and optimizing the network. Our visualization framework enables three abilities: first, find salient features of geometric signatures that affect the expression recognition result; second, find expression-specific features for each expression; third, modify the networks by removing unnecessary neurons with low activation values for network simplification. As we cannot accurately know how much neurons and layers are needed for training the

3DMCNN at the beginning, we usually provide sufficient number of neurons and layers as an initial setup. Then, the neurons with high activations for a certain expressions are clustered and the layouts of the network are rearranged accordingly. By analyzing these different clusters of neurons, we can understand important features and the neurons which are sensitive to them. Selecting the significant features and high activation neurons as the initial state of the retraining, the network structure can be simplified and the performance can be optimized.

Fig. 32 shows an example for visual analytics interface of the trained 3DMCNN. The network shows the first two layers, and each layer is composed by a convolution layer, a Rectified Linear Unit (ReLU) layer and a pooling layer. A three channel input data is passed to the 3DMCNN, and we visualize the first two group of layers. One of the neurons in the second layer is selected and the corresponding feature map is shown as well as the connected neurons in the first layer are highlighted. For the connections between neurons, the associated filters are displayed as well. The interactive inspection approach enables us to find and analyze the important features.

Clustering similar neurons into several functioning groups provides a clear overview of the learned network. In our visual analytics framework, the neurons are clustered in three types of groups: (1) High activation positive nodes, whose associated weights are large positive values. These nodes contribute positive features to the output nodes in the next layer. (2) High activation negative nodes, whose weights are large negative values. (3) Low activation nodes, whose associated weights are approximately zero. These nodes provides almost no features to the output layers, so they have no potentials and can be removed to reduce the network complexity and improve the performance of the network.

As illustrated in Fig. 33, a set of smiling 3D face data is passed to the trained 3DMCNN. The nodes in each layers are clustered and visualized, and the associated spectrum colors represent the different activation values. High activation positive nodes are labeled in red, low activation nodes are labeled in green and high negative activation nodes are labeled in blue. We cluster a node as a low activation node if the total contribution to the final output is less than 5%. The connectivities are also grouped for simplified visualization of the network. In Fig. 33, we trace back the classification result, for example “happiness”, to explore the learned feature patches. We consecutively select the high activation positive nodes to see the lower level neurons connected to them. Shallow layers of the CNN detect detailed features such as contours and color patches, while deeper layers detect more global features such as detecting the parts of the objects. As we learned by visualizing the feature maps in each layer, our 3DMCNN also learns the expressions in a similar manner: detects feature patches , and then detects larger areas of the face and their combinations. Since we start training the network with a sufficient number of neurons, there exists many redundant neurons. Visualizing and interactively removing these redundant neurons improve the network efficiency, simplify the network and reduce the over-fitting problems.

5.6 Experiment

We have applied our algorithm on public 3D face expression databases and Kinect scanned face models. The surface geometric features are experimented on first, followed by the training setups for the 3D MCNN. We compare our method with 2D CNN based method and geometry image based method for facial expression recognition. Then, we analyze cases in which our method performs better than other methods. Our experiments

have been performed on a Linux PC with 3.0 GHz 8 Core CPU, 8 GB memory and GeForce 980 graphics card with the NVIDIA CUDA Framework 6.5. The computation time of our method is mainly spent on the 3D Mesh CNN training process. The reconstruction and feature computation time for 3D face is approximately 3 seconds for each image.

5.6.1 Datasets

To evaluate our method, we employ two public 3D expression databases for training and testing. FaceWarehouse is used for training and 3D face generation in testing phase. BU-3DFE is used for testing and comparison among different learning methods.

FaceWarehouse [16]: FaceWarehouse is a database of 3D facial expressions for visual computing applications. It includes 3D face scans of 150 individuals aged between 7 to 80, from various ethnic backgrounds. There are 20 expressions including neutral expression for each person. For each expression, both 3D model and the 2D image with landmarks are provided. The landmarks include important facial feature points, which can be used for fine tune the reconstructed 3D faces. The 3D face models with expressions are obtained by deforming a template facial mesh to match the scanned depth image, These meshes with consistent topology are assembled as a rank-3 tensor to build a bilinear face model with identity and expression. In our experiment, the expressions are manually labeled for 6 prototypic expressions (happiness, disgust, fear, angry, surprise and sadness).

BU-3DFE (Binghamton University 3D Facial Expression) Database [77]: BU-3DFE database includes 100 subjects (56% female, 44% male), aged from 18 to 70, with a variety of ethnic backgrounds, including White, Black, East-Asian, Middle-east Asian, Indian, and Hispanic Latino. There are 7 expressions for each subject including neutral expression and the six prototypic expressions. For each non-neutral expression, there are four 3D shapes

with four levels of intensity. Therefore, there are 25 instances of 3D expression models for each subject, resulting in a total of 2,500 3D facial expression models in the database. A color facial image is associated with each expression shape model.

5.6.2 Visual Analytics Guided CNN Design and Optimization

We train the 3DMCNN with the FaceWarehouse dataset and test with the BU-3DFE dataset. Since there are symmetric and similar expressions, the data is manually labeled in 5 non-neutral classes and 1 neutral class. To improve the training robustness, we generate some randomness ($\pm 5\%$) to the 3D face reconstruction process to obtain more training datasets: (1) Add a random variance to the identity coefficient w_i to generate a new identity. (2) Add a random variance to the expression coefficient w_e. Including the original 150 faces of the subjects and 450 synthetic data, we prepare 600 faces for each expression. We compute the 3 types of signatures on the 3D faces. Fig. 35 illustrates the computed geometry signatures, (a) mean curvature, (b) conformal factor and (c) heat kernel. The geometry features show similarity within the same expression across different subjects while vary between different expressions.

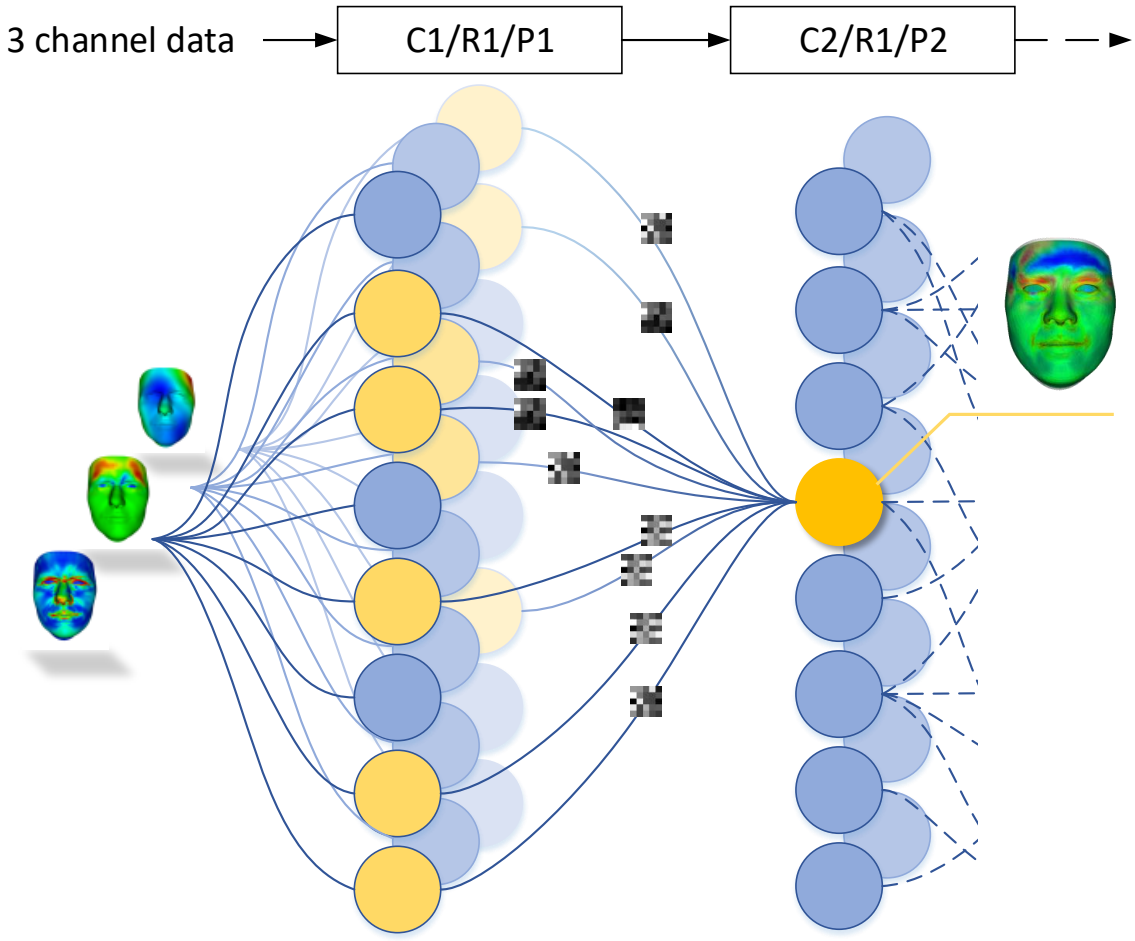


Figure 32: Illustration of the network visualization approach. The selected neuron in layer 2 is highlighted in yellow and its feature map is shown. The connection with the neurons in layer 1 is shown and the associated filters are displayed accordingly.

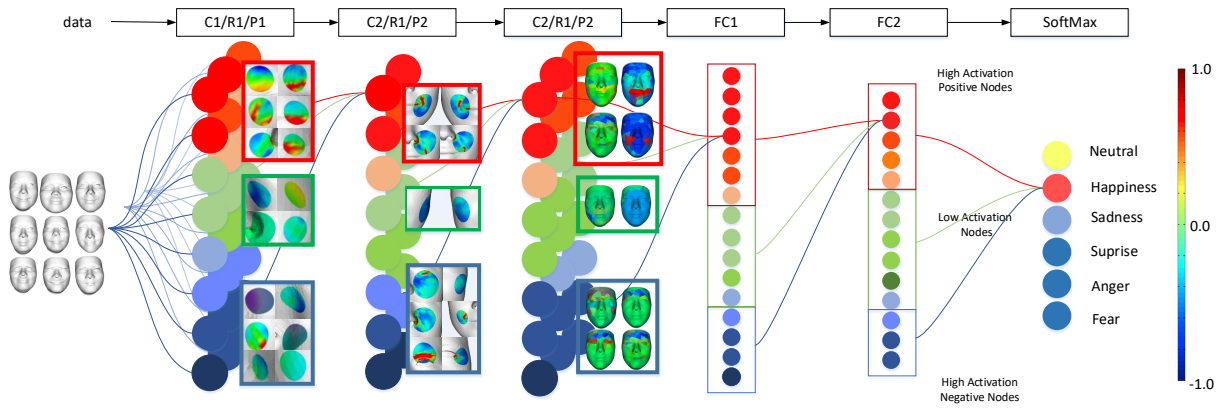


Figure 33: Illustration of the feature visualization process. Red cluster contains high activation positive nodes, green cluster contains low activation nodes and blue cluster contains high activation negative nodes. Gray nodes are unconnected node to the selected node. Sample feature maps are displayed beside each cluster.

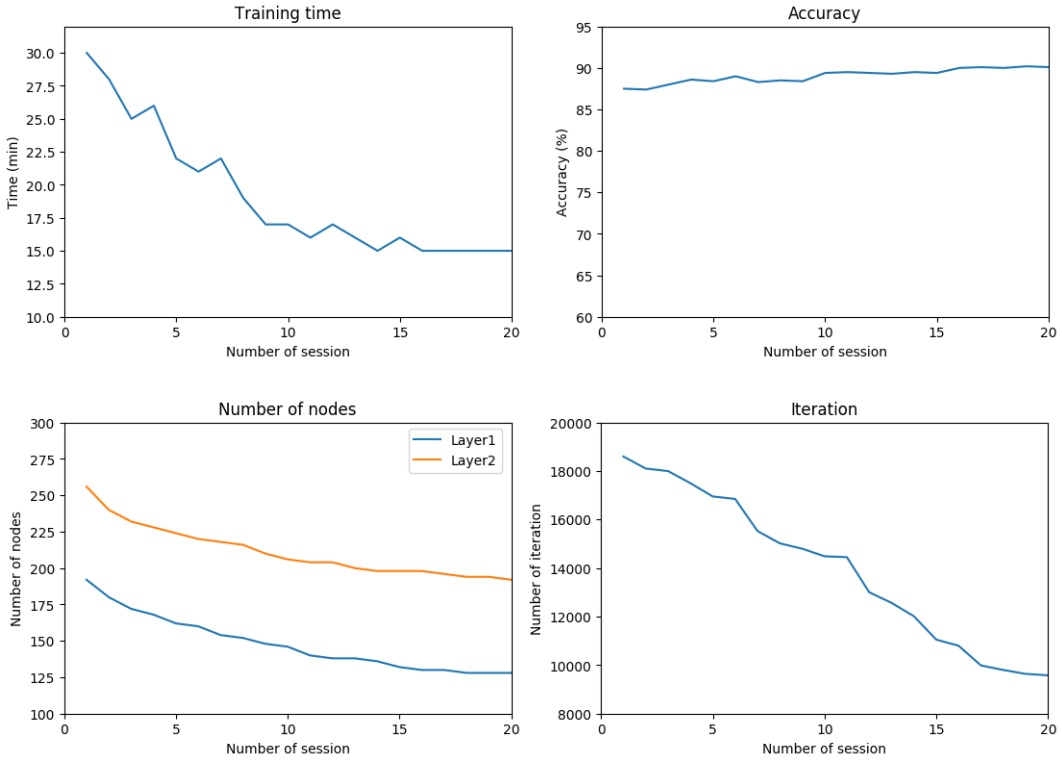


Figure 34: Evaluation of the interactive network simplification. Graphs shows the training time, accuracy, node number and maximum iteration with respect to the modification sessions.

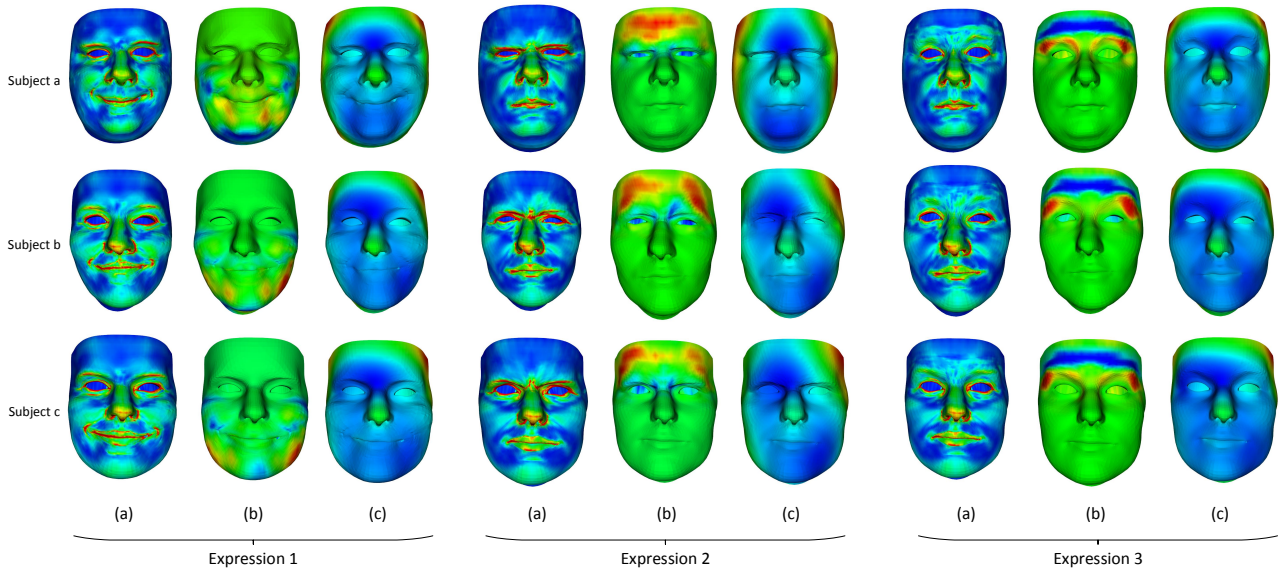


Figure 35: The geometry signatures on the 3D face. Each row shows different expressions of the same subject. Expression 1, 2, 3 shows three expressions: Happiness, Anger and Surprise. In each expression, column (a) shows the mean curvature, column (b) shows conformal factor and column (c) shows the heat kernel signature.

We start with 3 layers of convolution layer, ReLU layer and Pooling layer followed by two fully connected layers and a SoftMax classification layer. We set the number of nodes to 192 for convolution layer 1, 256 for convolution layer 2 and 3, respectively, which we consider as sufficient numbers for neurons. We set the weight update ratio to 0.0005 and run for 20000 iterations. Once we obtain initial classification network, we use our node clustering and visualization method to selectively remove low activation neurons and retrain the network with the selected high activation neurons as the initial status. An example of the optimization process is discussed in the following section. Fig. 34 shows the evaluation results of the network simplification sessions. After 20 sessions of modification of the network based on our visual analytics approach, the training time, the number of nodes and the number of maximum iterations are reduced while keeping a similar prediction accuracy. The final nodes is set to 128 for convolution layer 1, 192 for layer 2 and 256 for layer 3. Using the final network, we test on the public database and the results are discussed in Section 5.7.

5.6.3 Case Study

In this section, we provide further details on the analysis performed with our visual analytics approach. As we discussed in Section 5.6.2, we optimize the network by reducing the redundancy. Since we start with a sufficient number of neurons, it usually generates many low activation neurons, which increases the computational cost without contribution to the classification result. These neurons often have low connectivity to the subsequent layers as well. Our goal is to reduce the number of neurons with low activation values and low connectivity. We use the histogram plots to observe the distributions of the activation

values. Fig. 36 shows the activation histograms of the convolutional layers. Activation values were scaled to a range between -1 and 1, and 21 bins were selected to generate the histograms. Typical feature maps of the clusters are displayed accordingly on the histograms. The detailed lists of the neurons can be shown by selecting the histogram bins. The upper row of Fig. 36 shows the initial stage before the network tuning, and the lower row shows the final stage after optimization. Initially, there is a large number of low activation nodes in the convolution layer 1 as shown in the histogram. We further inspect the learned features via the filter list. Fig. 37 shows the list of the selected cluster of the neurons in an increasing order of the activation value. The connectivity (out) shows how many neurons in the subsequent layer are connected and the connectivity (in) shows how many preceding neurons are connected. The list can be rearranged by selecting different orders. We remove those neurons with low activation and connectivity, and then, re-train the network by keeping the rest of the feature maps. The neuron selection and removal can be reversed if the result shows unexpected changes. The bottom row of Fig. 36 shows the activation histograms in the final stage, where the number of the low activation neurons were greatly reduced compared to other neurons. Based on the modification using our visual analytics approach, we optimize the network by reducing the first convolutional layer from 192 to 128 filters, and the second convolutional layer from 256 to 192. The optimized result provides a compact network with reduced training time and the similar prediction accuracy.

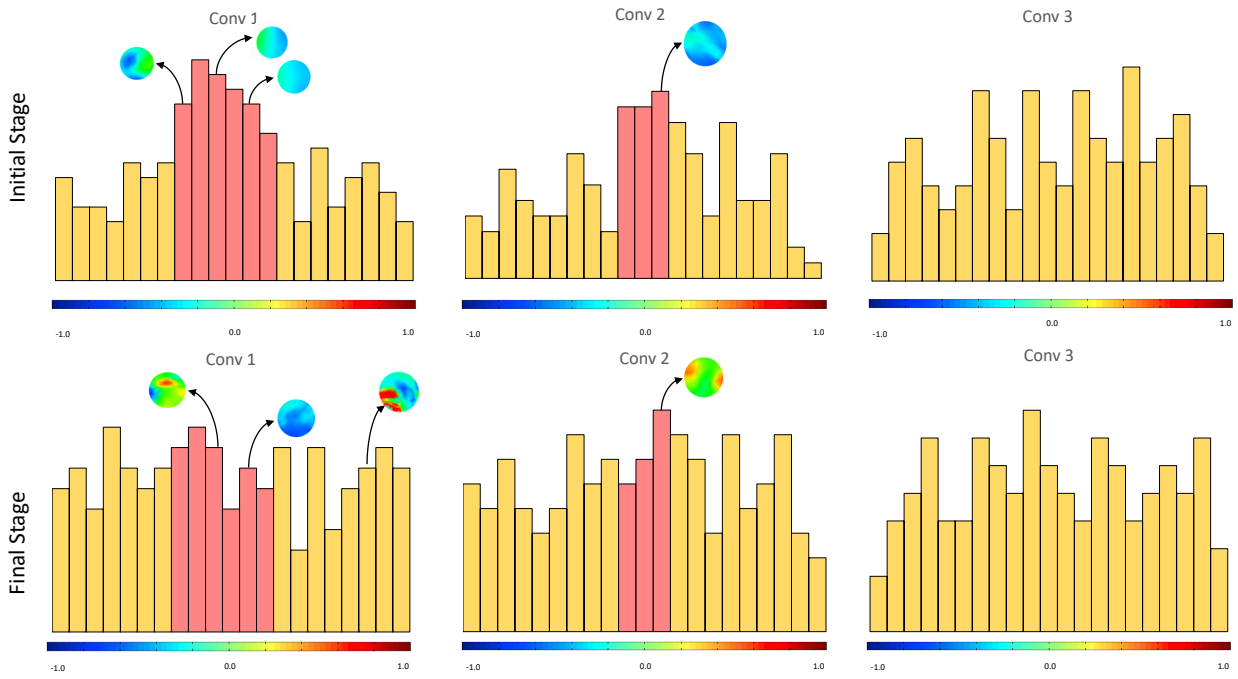


Figure 36: The activation histograms of the convolution layers. The upper row shows the histograms of the initial stage and the lower row shows histograms of the final optimized stage. Some sample filters are displayed on the histograms.

5.7 Comparisons and Evaluation

We evaluate our 3DMCNN framework by selecting different geometric signatures separately and as well as their combinations to train a 3DMCNN. To fairly compare their performance in terms of providing sufficient features for training, we use the same 3DMCNN architecture and the same number of training data. Table 2 shows the test accuracy for each expression and as well as the average. Mean curvature achieves the highest accuracy for 3DMCNN using the single feature only, followed by the conformal factor while heat kernel achieves lowest accuracy. This is because mean curvature and conformal factor carry more detailed information than heat kernel in local areas. Combining the three

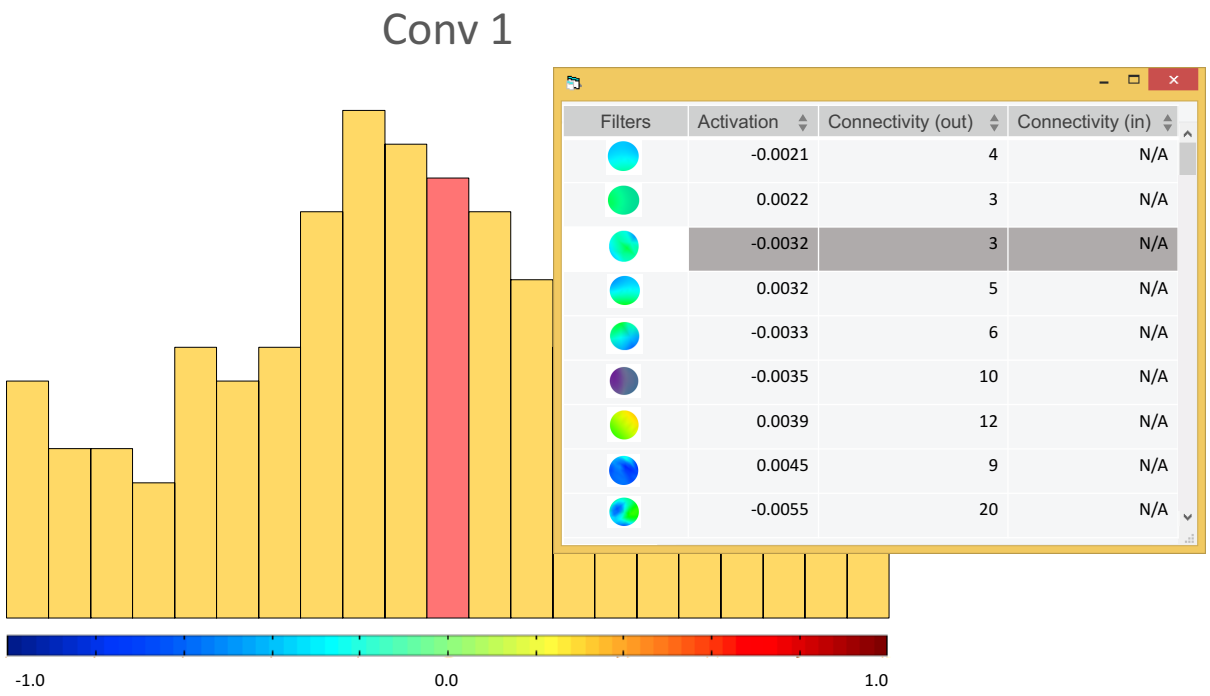


Figure 37: The detailed inspection on the selected group of neurons. The table shows the list of the filters with low activations.

Features	Neutral(%)	Happiness(%)	Sadness(%)	Angry(%)	Surprise(%)	Fear(%)	Average (%)
Mean Curvature	85.2	83.5	82.3	85.6	83.1	83.3	83.8
Conformal Factor	80.1	82.6	80.2	75.1	77.5	78.5	79.0
Heat Kernel Signature	60.1	56.2	62.1	66.7	51.6	53.6	58.4
Combined	92.3	90.1	89.5	88.1	90.2	89.6	89.8

Table 2: Training accuracy based on single geometry signature and their combination for each expression.

signatures as three channels of the input significantly improved the classification accuracy.

We also compare our method with several existing methods: (1) Traditional image based CNN [50], which uses image to train the CNN. (2) Geometry image CNN [65], which maps the 3D shape to 2D image and generates synthetic geometric images for training regular CNN. (3) Volumetric CNN [76], which extended 2D image CNN to 3D cubic CNN. These methods are proposed for general 3D shape recognition purpose, and we employ them to our face expression recognition application for comparisons. We also implement

extra two types of methods derived from our method. (4) The 3D faces with their geometric signatures are rendered to projection images with virtual cameras and take the images for training and testing. We denote this method as projection image based method: (5) Since we capture the RGB information with the depth value simultaneously, we are actually able to obtain 3D faces with RGB textures. We combine these color information with geometric signatures to train a 6 channel 3DMCNN, denoted as 3D Mesh&Image CNN. We compare these method under the same configuration of the network architecture, i.e., same number of layers and neurons. Fig. 38 summarizes the results. Our 3DMCNN and 3D Mesh&Image CNN achieved the highest accuracy. Geometric image CNN [65] also achieves a high classification accuracy (85%), since it also focuses on the geometric features. However, the 2D authelic mapping used in the method does not guarantee a distance preserving mapping, since the convolution on the 3D surface domain is non-uniform. Projection image based method achieves slightly higher accuracy than the RGB image-based method, but lower than our mesh based method. This is because they both learn other face related features, such as face contour, color patch and so on. Volumetric CNN achieves the lowest accuracy, since it needs extra depth of the network to reach its best performance. As a result, our method achieves the best accuracy (90%) under a shallow and compact network configuration.

5.7.1 Knowledge from Visual Analytics of CNN

Our visualization system provides an interactive way of neuron cluster selection for visualizing the learned features. By selecting the highest activation cluster, the features are deconvolved back to the input 3D face space and the areas are highlighted. These areas

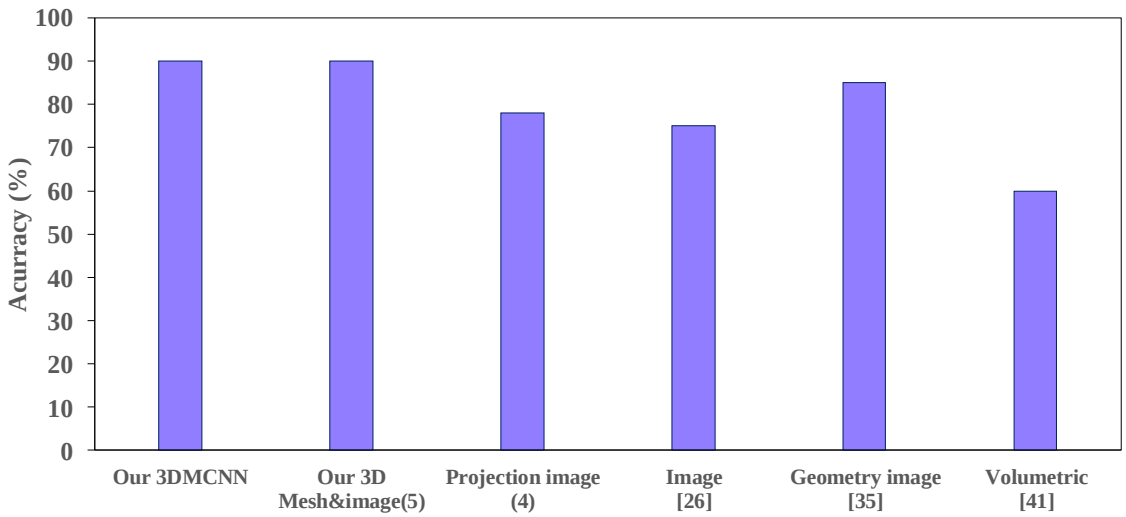


Figure 38: The comparison result between our method and other CNN-based methods. Our method achieves highest recognition accuracy.

are important since they lead to the strongest responses in the CNN. Fig. 39 shows the high activation areas for two different expressions. The top nine high activation areas are selected for demonstration in each expression. Fig. 39 (a) shows the features of “happiness” class and (b) shows features of “anger” class. High activation areas for happiness are mostly located around mouth corners as shown in the result. Furthermore, these features have strong reaction on mean curvature and only one feature area reacts on the conformal factor (lower left of (a)). On the other hand, high activation areas for “anger” are distributed around eyes and forehead. Contrast to “happiness” class, four areas reacts to conformal factor for “anger” class. From the visualization result, we know different geometric signatures carry different information for certain expressions. Fig. 40 shows the high activation areas for “happiness” and “anger” for comparison. We map these areas to the template face domain for consistent visualization purpose. Features are located around mouth and eyes for “happiness”. Since features round eyes are also considered as important as mouth cor-

ner, 2D image based method may not work well for some ambiguous expressions. Fig. 41 shows three examples where image based method fails to correctly classify the expressions, however, we successfully classify it using our method. Fig. 41 (a) shows the subject unintentionally closed her eyes while smiling, which leads to misclassification in 2D image based method. (b) shows the case is classified to “anger” using image-based method under extreme illumination, whereas was classified to “happiness” using our method. (c) shows the failed case for image-based method under large rotation. Our method provides a robust and stable classification outcome to some ambiguous expressions and also under some uncontrolled environment conditions and extreme head poses.

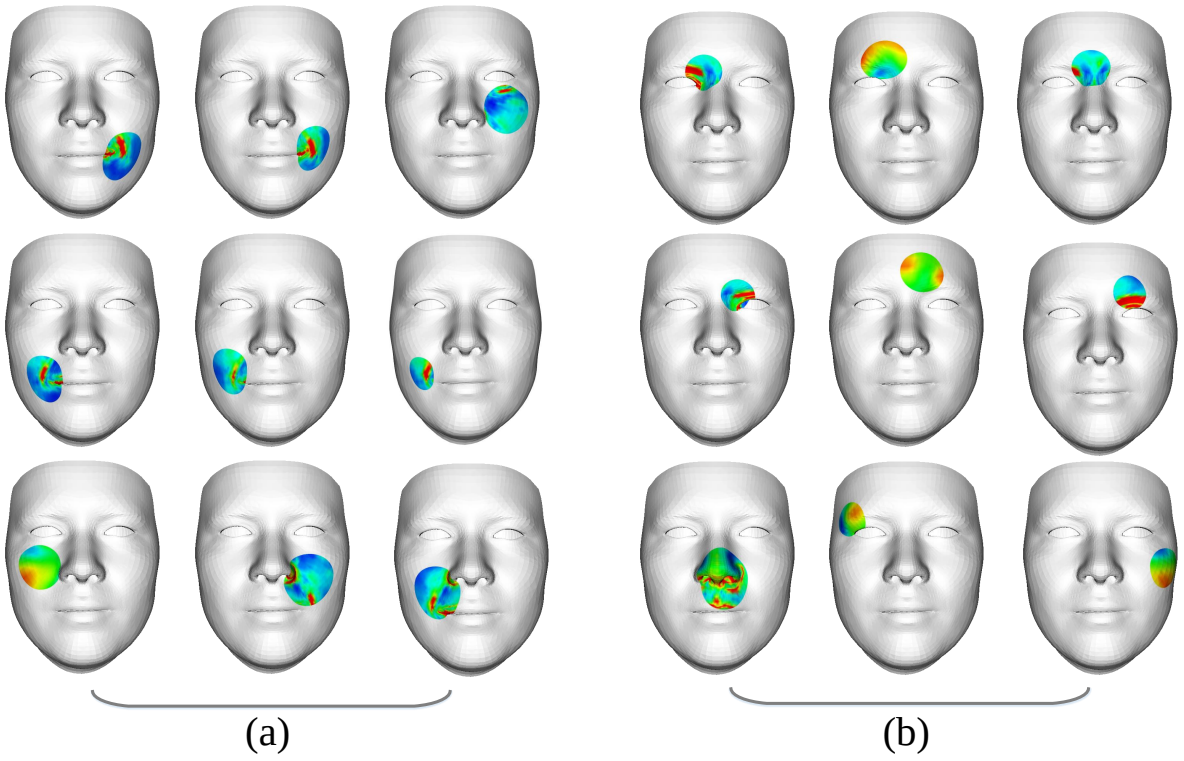
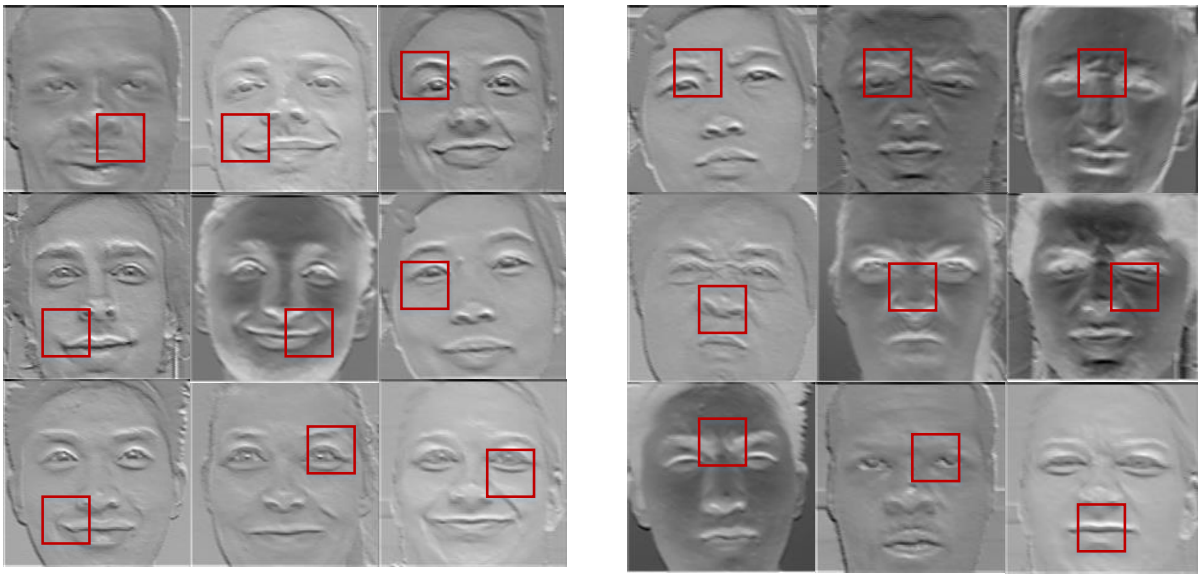


Figure 39: High activation feature areas on 3D face surface of two expression classes: (a) is happiness and (b) is anger. All the feature areas are mapped to the template face for consistent visualization.



(a) (b)

Figure 40: High activation feature areas on 2D images of two expression classes: (a) is happiness and (b) is anger.

5.7.2 Limitations

Although the 3D Mesh CNN can achieve a high performance on facial expression recognitions, there are two main limitations. One is that our method needs a uniform sampling standard across the 3D shapes. This is because, similar to the image-based CNN, the computations of convolution and pooling need to be performed consistently on the 3D surface under a uniformly defined structure. Further studies need to be done to apply the 3DMCNN to general classification task. The second limitation is that the 3D deformable face model contains limited high frequency details. Fine local details such as wrinkles are not properly modeled, therefore, result in the fitted model did not contain these information either. This disadvantage would affect the potential peak performance of the 3DMCNN to

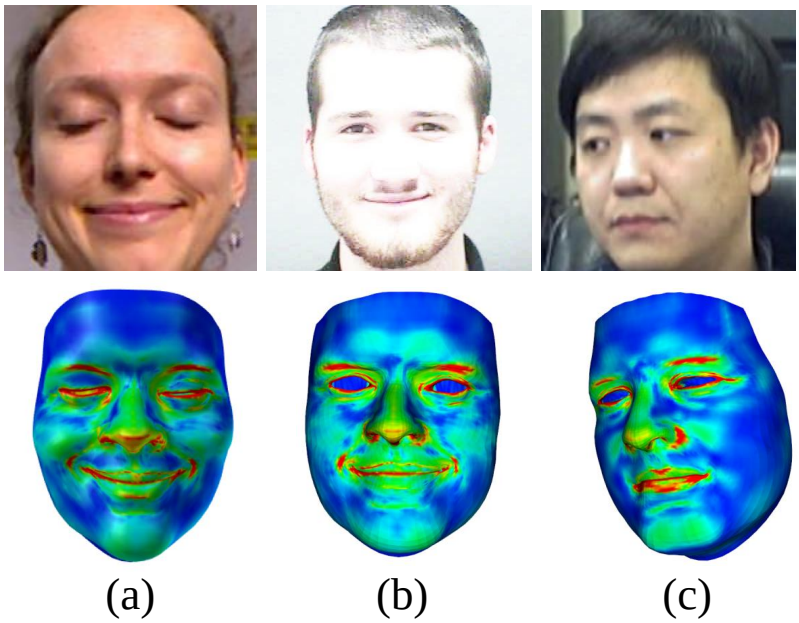


Figure 41: Successful classification cases of using our method while 2D image based method misclassified.

learn deeper features using smaller size of convolution filters.

CHAPTER 6 CONCLUSION

3D face reconstruction and facial expression analytics using 3D faces are important topics in computer graphics and computer vision, which enables many useful applications. Inspired by the recent research, we applied a NMF-based decomposition approach to the 3D face database to create a part-based 3D face model, which improved the reconstructed details of the 3D faces. We extract features directly from the reconstructed 3D face models to train regressors to estimate the emotion states. We also propose to use the most state-of-art CNN framework to improve the estimation accuracy. The main contributions include:

- We have presented a novel 3D face capture and reconstruction solution that can robustly and accurately acquire 3D face models using a single smartphone camera. In this solution, a deformable NMF part-based 3D face model, learned from a 3D face database, is developed to facilitate robust global and adaptive detail data fitting alternatively through the variations of weights. The part-based 3D face representation serves as a better morphable model for data fitting and reconstruction under geometry and illumination constraints, as the NMF bases are corresponding to localized features that correlate better with the parts of 3D faces. In our system, self-portrait frontal and side photographs are used as the input to the fully automated iterative reconstruction process. It permits fully automatic registration of the deformable part-based 3D face model to the input images and the detailed geometric features as well as illumination constraints to reconstruct a high-fidelity 3D face model. The system is flexible as it allows users themselves to conduct the captures in any uncontrolled environment. The capability of our method is demonstrated by several users to capture and reconstruct their 3D faces using a smart phone camera.

- We have presented a 3D morphable face model-based approach for emotion analysis and information visualization in VA space. We have built a NMF part-based morphable 3D face model for reconstructing. Based on the input image, a 3D face with expression is reconstructed iteratively using the morphable 3D face model, from which basis parameters and a displacement map are extracted as features for emotion analysis. We have trained two Support Vector Regressions for the fuzzy Valence and Arousal values, respectively, using the composed feature vectors. The states of the continuous emotions can be effectively visualized by plotting them in the VA space. Our method is fully automatic to compute the VA values from images or a sequence of video with various expressions. And our visualization system also provides the expression details such as the image frames and generated 3D faces interactively by interacting with the VA-plot. The experiment results have shown that our method has achieved a remarkable emotion estimation accuracy and our visualization method can provide a clear understanding of continuous emotion data. We use a standard SVR as our emotion regression model and there is a potential improvement by applying other regressions such as fuzzy neural network. The current 3D face reconstruction step in our system normally takes about 3 seconds, which does not allow our system to process real-time streaming data on the fly.
- We have presented a 3D Mesh Convolutional Neural Network for facial expression recognition and an interactive visual analytics method for the purpose of designing and modifying the networks. Based on the depth surface scanning via a RGBD camera, we have reconstructed a 3D face model by fitting a deformable face model to the raw surface. We have adopted three types of geometric signatures, including mean

curvature, conformal factor and heat kernel, as feature values of the shape surface. These signatures can comprehensively describe the shape surface both locally and globally. Using the geometric signatures the 3DMCNN is trained. To uniformly convolve the sampling points on the face surface, we proposed a geodesic distance-based convolution scheme. This geodesic distance-based convolution and pooling method can prevent dislocated false features and preserve actual local features. We have trained and tested the 3DMCNN using two public 3D face expression databases and analyzed the effectiveness of our method by interactively visualizing the learned features on the neurons. Through the visualization results, we have demonstrated some high activation features that affect the recognition result most. We have compared our method with the traditional image-based CNN and our method achieves higher recognition accuracy. The visual analytics of the learned features shows that the geometric signatures are more sensitive and effective in learning facial expressions than image features.

APPENDIX

Journal Publications

1. Hai Jin, Xun Wang, Yuanfeng Lian, Jing Hua, "Emotion Information Visualization through Learning of 3D Morphable Face Model." *The Visual Computer*, 2018.
2. Hai Jin, Yuanfeng Lian, Jing Hua, "Visualizing and Learning Facial Expression with 3D Mesh Convolutional Neural Networks." *ACM Transactions on Intelligent Systems and Technology*, 2018.
3. Hai Jin, Xun Wang, Zichun Zhong and Jing Hua, "Robust 3D Face Modeling and Reconstruction from Frontal and Side Images," *Computer-Aided Geometric Design*, 2017.
4. Ali Shahini, Hai Jin, Zhixian Zhou, Yang Zhao, Pai-Yen Chen, Jing Hua, Mark Cheng, "Toward individually tunable compound eyes with transparent graphene electrode" *Bioinspiration & Biomimetics*, 2017.

Conference Publications

1. Abhimanyu Gosain, Mark Berman, Marshall Brinn, Thomas Mitchell, Chuan Li, Yuehua Wang, Hai Jin, Jing Hua, and Hongwei Zhang, "Enabling Campus Edge Computing using GENI Racks and Mobile Resources." *In Proceedings of IEEE/ACM Symposium on Edge Computing*, 2016.
2. Hai Jin, Zichun Zhong and Jing Hua, "Robust 3D Face Modeling and Reconstruction from Frontal and Side Images." *International Conference on Geometric Modeling and Processing*, 2016.

Demos and Posters

1. Jing Hua, Shaofeng Shu, Hai Jin, Hongwei Zhang, "Real-Time Wireless-Networked 3D Vision for Public Safety," *In the Smart Cities Connect Conference and Expo* (Demo), 2017.
2. Hongwei Zhang, Jing Hua, Shaofeng Shu, Hai Jin, Chuan Li, and Yu Chen, "Predictable Wireless Networking and Collaborative 3D Vision for Real-Time Cyber-Physical-Human (CPH) Systems," *In the US-Ignite Conference* (Poster), 2017.

Book Chapter

1. Hongwei Zhang, Le Yi Wang, George Yin, Shengbo Eben Li, Keqiang Li, Jing Hua, Yeuhua Wang, Chuan Li, Hai Jin, "Trustworthy Foundation for CAVs in an Uncertain World: From Wireless Networking, Sensing, and Control to Software-Defined Infrastructure," *Road Vehicle Automation, Springer*, 2016.

REFERENCES

[1] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," *IEEE transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 2037–2041, 2006.

[2] B. Amberg, A. Blake, A. Fitzgibbon, S. Romdhani, and T. Vetter, "Reconstructing high quality face-surfaces using model based stereo," in *IEEE International Conference on Computer Vision*, 2007, pp. 1–8.

[3] A. B. Ashraf, S. Lucey, J. F. Cohn, T. Chen, Z. Ambadar, K. M. Prkachin, and P. E. Solomon, "The painful face–pain expression recognition using active appearance models," *Image and Vision Computing*, vol. 27, no. 12, pp. 1788–1796, 2009.

[4] P. Axelsson, "Processing of laser scanner data-algorithms and applications," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 54, no. 2, pp. 138–147, 1999.

[5] M. S. Bartlett, G. Littlewort, I. Fasel, and J. R. Movellan, "Real time face detection and facial expression recognition: Development and applications to human computer interaction." in *IEEE Conference on Computer Vision and Pattern Recognition Workshop*, vol. 5, 2003, pp. 53–53.

[6] T. Beeler, B. Bickel, P. Beardsley, B. Sumner, and M. Gross, "High-quality single-shot capture of facial geometry," *ACM Transactions on Graphics*, vol. 29, no. 4, p. 40, 2010.

[7] T. Beeler, F. Hahn, D. Bradley, B. Bickel, P. Beardsley, C. Gotsman, R. W. Sumner, and M. Gross, "High-quality passive facial performance capture using anchor frames," *ACM Transactions on Graphics*, vol. 30, no. 4, p. 75, 2011.

[8] M. W. Berry, M. Browne, A. N. Langville, V. P. Pauca, and R. J. Plemmons, "Algorithms and applications for approximate nonnegative matrix factorization," *Computational Statistics & Data Analysis*, vol. 52, no. 1, pp. 155–173, 2007.

[9] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3D faces," in *SIGGRAPH*. ACM, 1999, pp. 187–194.

[10] ——, "Face recognition based on fitting a 3D morphable model," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 9, pp. 1063–1074, 2003.

[11] K. W. Bowyer, K. Chang, and P. Flynn, "A survey of approaches and challenges in 3D and multi-modal 3D+ 2D face recognition," *Computer Vision and Image Understanding*, vol. 101, no. 1, pp. 1–15, 2006.

[12] M. M. Bronstein and I. Kokkinos, "Scale-invariant heat kernel signatures for non-rigid shape recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 1704–1711.

[13] T. Campbell, C. Williams, O. Ivanova, and B. Garrett, "Could 3D printing change the world," *Technologies, Potential, and Implications of Additive Manufacturing*, 2011.

[14] C. Cao, Q. Hou, and K. Zhou, "Displaced dynamic expression regression for real-time facial tracking and animation," *ACM Transactions on Graphics*, vol. 33, no. 4, p. 43, 2014.

[15] C. Cao, Y. Weng, S. Lin, and K. Zhou, "3D shape regression for real-time facial animation," *ACM Transactions on Graphics*, vol. 32, no. 4, p. 41, 2013.

[16] C. Cao, Y. Weng, S. Zhou, Y. Tong, and K. Zhou, "Facewarehouse: A 3D facial expression database for visual computing," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 3, pp. 413–425, 2014.

[17] C. Cao, H. Wu, Y. Weng, T. Shao, and K. Zhou, "Real-time facial animation with image-based dynamic avatars," *ACM Transactions on Graphics*, vol. 35, no. 4, p. 126, 2016.

[18] Y.-L. Chen, H.-T. Wu, F. Shi, X. Tong, and J. Chai, "Accurate and robust 3D facial capture using a single RGBD camera," in *IEEE International Conference on Computer Vision*, 2013, pp. 3615–3622.

[19] T. F. Cootes, G. J. Edwards, C. J. Taylor *et al.*, "Active appearance models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 681–685, 2001.

[20] T. F. Cootes, G. V. Wheeler, K. N. Walker, and C. J. Taylor, "View-based active appearance models," *Image and Vision Computing*, vol. 20, no. 9, pp. 657–664, 2002.

[21] D. Cristinacce and T. F. Cootes, "Feature detection and tracking with constrained local models," in *British Machine Vision Conference*, vol. 1, no. 2, 2006, p. 3.

[22] P. Ekman, "An argument for basic emotions," *Cognition & Emotion*, vol. 6, no. 3-4, pp. 169–200, 1992.

[23] P. Ekman and W. V. Friesen, "Facial action coding system," 1977.

[24] G. Fanelli, M. Dantone, J. Gall, A. Fossati, and L. Van Gool, "Random forests for real time 3D face analysis," *International Journal of Computer Vision*, vol. 101, no. 3, pp. 437–458, 2013.

[25] B. Fasel and J. Luettin, "Automatic facial expression analysis: a survey," *Pattern Recognition*, vol. 36, no. 1, pp. 259–275, 2003.

[26] X. Geng, Z.-H. Zhou, and K. Smith-Miles, "Automatic age estimation based on facial aging patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 12, pp. 2234–2240, 2007.

[27] S. Granger and X. Pennec, "Multi-scale EM-ICP: A fast and robust approach for surface registration," in *International Conference on Computer Vision*, 2002, pp. 418–432.

[28] H. Gunes and M. Pantic, "Automatic, dimensional and continuous emotion recognition," *International Journal of Synthetic Emotions*, vol. 1, no. 1, pp. 68–99, 2010.

[29] J. Hua, Z. Lai, M. Dong, X. Gu, and H. Qin, "Geodesic distance-weighted shape vector image diffusion," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, 2008.

[30] A. E. Ichim, S. Bouaziz, and M. Pauly, "Dynamic 3D avatar creation from hand-held video input," *ACM Transactions on Graphics*, vol. 34, no. 4, p. 45, 2015.

[31] S. V. Ioannou, A. T. Raouzaiou, V. A. Tzouvaras, T. P. Mailis, K. C. Karpouzis, and S. D. Kollias, "Emotion recognition through facial expression analysis based on a neurofuzzy network," *Neural Networks*, vol. 18, no. 4, pp. 423–435, 2005.

[32] A. Irschara, C. Zach, J.-M. Frahm, and H. Bischof, "From structure-from-motion point clouds to fast location recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 2599–2606.

[33] H. Jin, X. Wang, Z. Zhong, and J. Hua, "Robust 3D face modeling and reconstruction from frontal and side images," *Computer Aided Geometric Design*, vol. 50, pp. 1–13, 2017.

[34] P. Joshi, W. C. Tien, M. Desbrun, and F. Pighin, "Learning controls for blend shape based realistic facial animation," in *SIGGRAPH*. ACM, 2005, p. 8.

[35] M. Kahng, P. Andrews, A. Kalro, and D. H. Chau, "Activis: Visual exploration of industry-scale deep neural network models," *arXiv preprint arXiv:1704.01942*, 2017.

[36] A. Kapoor, W. Burleson, and R. W. Picard, "Automatic prediction of frustration," *International Journal of Human-Computer Studies*, vol. 65, no. 8, pp. 724–736, 2007.

[37] K. Khoshelham and S. O. Elberink, "Accuracy and resolution of kinect depth data for indoor mapping applications," *Sensors*, vol. 12, no. 2, pp. 1437–1454, 2012.

[38] B.-K. Kim, J. Roh, S.-Y. Dong, and S.-Y. Lee, "Hierarchical committee of deep convolutional neural

networks for robust facial expression recognition," *Journal on Multimodal User Interfaces*, vol. 10, no. 2, pp. 173–189, 2016.

[39] J. Kittler, A. Hilton, M. Hamouz, and J. Illingworth, "3D assisted face recognition: A survey of 3D imaging, modelling and recognition approachest," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2005, pp. 114–114.

[40] K.-E. Ko and K.-B. Sim, "Development of the facial feature extraction and emotion recognition method based on ASM and bayesian network," in *IEEE International Conference on Fuzzy Systems*, 2009, pp. 2063–2066.

[41] H. Kobayashi and F. Hara, "Facial interaction between animated 3d face robot and human beings," in *IEEE International Conference on Computational Cybernetics and Simulation*, vol. 4, 1997, pp. 3732–3737.

[42] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

[43] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[44] ——, "Algorithms for non-negative matrix factorization," in *Advances in Neural Information Processing Systems*, 2001, pp. 556–562.

[45] Z. Lei, Q. Bai, R. He, and S. Li, "Face shape recovery from a single image using cca mapping between tensor spaces," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2008, pp. 1–7.

[46] H. Li, T. Weise, and M. Pauly, "Example-based facial rigging," in *ACM Transactions on Graphics*, vol. 29, no. 4, 2010, p. 32.

[47] X. Li and S. Iyengar, "On computing mapping of 3D objects: A survey," *ACM Computing Surveys*, vol. 47, no. 2, p. 34, 2015.

[48] C. P. Lim, D. Nonis, and J. Hedberg, "Gaming in a 3D multiuser virtual environment: Engaging students in science lessons," *British Journal of Educational Technology*, vol. 37, no. 2, pp. 211–231, 2006.

[49] M. Liu, J. Shi, Z. Li, C. Li, J. Zhu, and S. Liu, "Towards better analysis of deep convolutional neural networks," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 91–100, 2017.

[50] A. T. Lopes, E. de Aguiar, A. F. De Souza, and T. Oliveira-Santos, "Facial expression recognition with

convolutional neural networks: Coping with few data and the training sample order," *Pattern Recognition*, vol. 61, pp. 610–628, 2017.

[51] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, "The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2010, pp. 94–101.

[52] M. Meyer, M. Desbrun, P. Schröder, A. H. Barr *et al.*, "Discrete differential-geometry operators for triangulated 2-manifolds," *Visualization and Mathematics*, vol. 3, no. 2, pp. 52–58, 2002.

[53] R. V. Mohamed Daoudi, Anuj Srivastava, *3D face modeling, analysis and recognition*. WILEY, 2013.

[54] A. Mollahosseini, D. Chan, and M. H. Mahoor, "Going deeper in facial expression recognition using deep neural networks," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2016, pp. 1–10.

[55] L.-P. Morency, A. Rahimi, and T. Darrell, "Adaptive view-based appearance models," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, 2003, pp. I–803.

[56] R. A. Newcombe, D. Fox, and S. M. Seitz, "Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 343–352.

[57] M. A. Nicolaou, H. Gunes, and M. Pantic, "Continuous prediction of spontaneous affect from multiple cues and modalities in valence-arousal space," *IEEE Transactions on Affective Computing*, vol. 2, no. 2, pp. 92–105, 2011.

[58] N. Pezzotti, T. Höllt, J. Van Gemert, B. P. Lelieveldt, E. Eisemann, and A. Vilanova, "Deepeyes: Progressive visual analytics for designing deep neural networks," *IEEE Transactions on Visualization and Computer Graphics*, 2017.

[59] G. Sandbach, S. Zafeiriou, M. Pantic, and D. Rueckert, "Recognition of 3d facial expression dynamics," *Image and Vision Computing*, vol. 30, no. 10, pp. 762–773, 2012.

[60] D. Scharstein and R. Szeliski, "High-accuracy stereo depth maps using structured light," in *Computer Vision and Pattern Recognition*, vol. 1, 2003, pp. I–195.

[61] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 815–823.

[62] B. Schuller, M. Valster, F. Eyben, R. Cowie, and M. Pantic, "Avec 2012: the continuous audio/visual emotion challenge," in *ACM International Conference on Multimodal Interaction*, 2012, pp. 449–456.

[63] N. Sebe, M. S. Lew, Y. Sun, I. Cohen, T. Gevers, and T. S. Huang, "Authentic facial expression analysis," *Image and Vision Computing*, vol. 25, no. 12, pp. 1856–1863, 2007.

[64] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms," in *Computer Vision and Pattern Recognition*, vol. 1, 2006, pp. 519–528.

[65] A. Sinha, J. Bai, and K. Ramani, "Deep learning 3D shape surfaces using geometry images," in *European Conference on Computer Vision*, 2016, pp. 223–240.

[66] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3D shape recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 945–953.

[67] J. Sun, M. Ovsjanikov, and L. Guibas, "A concise and provably informative multi-scale signature based on heat diffusion," in *Computer Graphics Forum*, vol. 28, no. 5, 2009, pp. 1383–1392.

[68] S. Suwajanakorn, I. Kemelmacher-Shlizerman, and S. M. Seitz, "Total moving face reconstruction," in *European Conference on Computer Vision*, 2014, pp. 796–812.

[69] J. R. Tena, F. De la Torre, and I. Matthews, "Interactive region-based linear 3D face models," *ACM Transactions on Graphics*, vol. 30, no. 4, p. 76, 2011.

[70] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *IEEE Conference on Computer Vision and Pattern Recognition*, 1991, pp. 586–591.

[71] M. Valstar, B. Schuller, K. Smith, T. Almaev, F. Eyben, J. Krajewski, R. Cowie, and M. Pantic, "Avec 2014: 3D dimensional affect and depression recognition challenge," in *ACM International Workshop on Audio/Visual Emotion Challenge*, 2014, pp. 3–10.

[72] M. F. Valstar and M. Pantic, "Combined support vector machines and hidden markov models for modeling facial action temporal dynamics," in *International Workshop on Human-Computer Interaction*. Springer, 2007, pp. 118–127.

[73] H. Wang and N. Ahuja, "Facial expression decomposition," in *IEEE International Conference on Computer Vision*, 2003, pp. 958–965.

[74] Z. Wen and T. S. Huang, “3D face modeling,” *3D Face Processing: Modeling, Analysis and Synthesis*, pp. 11–17, 2004.

[75] M. Westoby, J. Brasington, N. Glasser, M. Hambrey, and J. Reynolds, “âĂŸstructure-from-motionâĂŹphotogrammetry: A low-cost, effective tool for geoscience applications,” *Geomorphology*, vol. 179, pp. 300–314, 2012.

[76] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, “3D shapenets: A deep representation for volumetric shapes,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1912–1920.

[77] L. Yin, X. Wei, Y. Sun, J. Wang, and M. J. Rosato, “A 3D facial expression database for facial behavior research,” in *Automatic Face and Gesture Recognition*, 2006, pp. 211–216.

[78] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” in *European Conference on Computer Vision*, 2014, pp. 818–833.

[79] Z. Zhang, P. Luo, C.-C. Loy, and X. Tang, “Learning social relation traits from face images,” in *IEEE International Conference on Computer Vision*, December 2015.

[80] Z. Zhang, “A flexible new technique for camera calibration,” *IEEE transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[81] ——, “Microsoft Kinect sensor and its effect,” *MultiMedia*, vol. 19, no. 2, pp. 4–10, 2012.

[82] X. Zhu and D. Ramanan, “Face detection, pose estimation, and landmark localization in the wild,” in *Computer Vision and Pattern Recognition*, 2012, pp. 2879–2886.

ABSTRACT

3D FACE RECONSTRUCTION AND EMOTION ANALYTICS WITH PART-BASED MORPHABLE MODELS

by

HAI JIN

May 2018

Advisor: Dr. Jing Hua

Major: Computer Science

Degree: Doctor of Philosophy

3D face reconstruction and facial expression analytics using 3D facial data are new and hot research topics in computer graphics and computer vision. In this paper, we first review the background knowledge for emotion analytics using 3D morphable face model, including geometry feature-based methods, statistic model-based methods and more advanced deep learning-bade methods. Then, we introduce a novel 3D face modeling and reconstruction solution that robustly and accurately acquires 3D face models from a couple of images captured by a single smartphone camera. Two selfie photos of a subject taken from the front and side are first used to guide our Non-Negative Matrix Factorization (NMF) induced part-based face model to iteratively reconstruct an initial 3D face of the subject. Then, an iterative detail updating method is applied to the initial generated 3D face to reconstruct facial details through optimizing lighting parameters and local depths. Our iterative 3D face reconstruction method permits fully automatic registration of a part-based face representation to the acquired face data and the detailed 2D/3D features to build a high-quality 3D face model. The NMF part-based face representation learned from a 3D face database facilitates effective global and adaptive local detail data fitting alternatively. Our system is flexible and it allows users to conduct the capture in any uncontrolled environment. We demonstrate the capability of our method by allowing users to capture and reconstruct their 3D faces by themselves.

Based on the 3D face model reconstruction, we can analyze the facial expression and the related emotion in 3D space. We present a novel approach to analyze the facial expressions from images and a quantitative information visualization scheme for exploring this

type of visual data. From the reconstructed result using NMF part-based morphable 3D face model, basis parameters and a displacement map are extracted as features for facial emotion analysis and visualization. Based upon the features, two Support Vector Regressions (SVRs) are trained to determine the fuzzy Valence-Arousal (VA) values to quantify the emotions. The continuously changing emotion status can be intuitively analyzed by visualizing the VA values in VA-space. Our emotion analysis and visualization system, based on 3D NMF morphable face model, detects expressions robustly from various head poses, face sizes and lighting conditions, and is fully automatic to compute the VA values from images or a sequence of video with various facial expressions. To evaluate our novel method, we test our system on publicly available databases and evaluate the emotion analysis and visualization results. We also apply our method to quantifying emotion changes during motivational interviews. These experiments and applications demonstrate the effectiveness and accuracy of our method.

In order to improve the expression recognition accuracy, we present a facial expression recognition approach with 3D Mesh Convolutional Neural Network (3DMCNN) and a visual analytics guided 3DMCNN design and optimization scheme. The geometric properties of the surface is computed using the 3D face model of a subject with facial expressions. Instead of using regular Convolutional Neural Network (CNN) to learn intensities of the facial images, we convolve the geometric properties on the surface of the 3D model using 3DMCNN. We design a geodesic distance-based convolution method to overcome the difficulties raised from the irregular sampling of the face surface mesh. We further present an interactive visual analytics for the purpose of designing and modifying the networks to analyze the learned features and cluster similar nodes in 3DMCNN. By removing low activity nodes in the network, the performance of the network is greatly improved. We compare our method with the regular CNN-based method by interactively visualizing each layer of the networks and analyze the effectiveness of our method by studying representative cases. Testing on public datasets, our method achieves a higher recognition accuracy than traditional image-based CNN and other 3D CNNs. The presented framework, including 3DMCNN and interactive visual analytics of the CNN, can be extended to other applications.

AUTOBIOGRAPHICAL STATEMENT

Hai Jin is a Ph. D. student in the Computer Science Department of Wayne State University. He is a member of the Graphics and Imaging Group, led by Dr. Hua Jing. His primary interest lies in computer graphics, computer vision and visualization, especially with learning-based 3D face modeling and visual analytics on facial expressions. He also worked on the system design and implementation of the integrated 3D Wayne State surveillance project, which is supported by the National Science Foundation. Prior to joining Wayne State University, he received his bachelor degree in Biomedical Engineering from Shanghai Jiao Tong University, China and master degree in Computer Science from Nagoya University, Japan. He has published in premier journals and conferences such as Computer Aided Geometric Design, The Visual Computer, Bioinspiration & Biomimetics. In addition, he has led a student team to present live demonstrations in the GENI Engineering Conference and received the Best Demo Awards in 2014 and 2015 respectively. He is a recipient of the Outstanding Graduate Teaching Assistant Award in Computer Science Department at Wayne State University.