Face Dynamic Modeling Based on Deep Learning and Feature Extraction

Lijing Tong, Jinqiu Yang*, Yuping Lai and Zequn Xiao

School of information, North China University of Technology, No. 5 Jinyuanzhuang Road, Shijingshan District, Beijing, 100144, People’s Republic of China

*Corresponding author: 13260050901@163.com

Abstract. Facial modeling is a key step to model visual effects in special movie effects and computer games. In this paper, a method based on the combination of deep learning and feature extraction is proposed for the modeling of 3D face model. Firstly, the face region is located for the captured face image. And then, the facial feature points are extracted by the landmark algorithm and the Convolutional Neural Network (CNN) is used to classify the facial expressions. Next, a special expression 3D face model is created by the deformation of the standard 3D face model based on the facial expressions classification result. Finally, the 3D face model and the extracted facial feature points are combined to perform personalized adjustment of the 3D model to complete a 3D facial expression animation system. The experimental results show that the proposed method can effectively perform the dynamic 3D face modeling which has high reality.

Keywords. Face modeling; Feature extraction; Expression classification; CNN

1. Introduction

Facial expression recognition [1-6] and modeling techniques have broad academic and commercial values. The method of this paper is as follows: Firstly, the images in the video are pre-processed by locating the face and extracting the feature point information. Here, the landmark algorithm is used to extract facial feature points in a face. It can mark 68 feature points in a face area. These feature points can mark the face contour, eyebrow and eyes, bridge of the nose, nostril and mouth outline and other characteristic information of the profile respectively. Then, the convolutional neural network (CNN) algorithm is used to classify the facial expression according to the displacements of these feature points. At the same time, the neutral Candide-3 facial expression model library is established by manually moving the positions of the Candide-3 model vertexes. So that, the corresponding neutral Candide-3 expression model can be called according to the result of the expression classification. After that, the Radial Basis Function (RBF) interpolation algorithm is used to complete the personalized deformation on the neutral Candide-3 expression model. Finally, a facial expression animation system is obtained. The overview of the proposed method is shown in figure 1.
2. Extraction of facial feature points and expression classification

2.1. Extraction of facial feature points

The landmark method has the characteristics of fast and stable performance. Therefore, the landmark method is used in this paper to extract facial feature points. In this method, a cascading regression is used to do facial feature alignment by the Ensemble of Regression Trees (ERT) [7]. The prediction of each regression is based on the prediction of the previous image.

In this paper, Let \( x_i \in \mathbb{R}^2 \) be the \( x, y \)-coordinates of the \( i^{th} \) facial landmark in an image \( I \). The facial shape is denoted by a vector \( S \). Furthermore, the vector \( S = (x_1^T, x_2^T, \ldots, x_p^T)^T \in \mathbb{R}^p \) denotes the coordinates of all the \( p \) facial landmarks in image \( I \). And the current estimate of \( S \) is denoted by \( \hat{S}^t \). Each regressor, \( \gamma_t(\ldots) \), in the cascade predicts an update vector from the image and \( \hat{S}^t \) that is added to the current shape estimate \( \hat{S}^{t+1} \) to improve the estimate.

\[
\hat{S}^{t+1} = \hat{S}^t + \gamma_t(I, \hat{S}^t)
\]

(1)

Here, each regressor \( \gamma_t(\ldots) \), is learned using the gradient boosting tree algorithm. For the training data\((I_1, S_1), \ldots, (I_n, S_n)\), where each \( I_i \) is a face image and \( S_i \) is its shape vector, a triplets of a face image, an initial shape estimate and the target update step is created from the training data to learn the first regression function \( r_0 \) in the cascade.

Assuming the data set \( \{(i, \hat{S}_i^0, \delta_i)\}_{i=1}^n \) and the learning rate value \( \nu \), \( 0 < \nu \leq 1 \), \( f_0(I, \hat{S}^t) \) is initialized as:
\[
\hat{f}_0(I, \hat{S}^t) = \arg \min_{y \in \mathbb{R}^2} \sum_{i=1}^{N} \left\| \Delta s_i^{(t)} - y \right\|^2
\]  
(2)

For \( k = 1, \ldots, K \), a loop is used to calculate the \( r_{ik} \):

1. Letting \( i = 1, \ldots, N \), the \( r_{ik} \) is calculated as:

\[
r_{ik} = \Delta s_i^{(t)} - f_{k-1}(l_{int}, \Delta s_i^{(t)})
\]  
(3)

2. A weak regression function \( g_k(I, \hat{S}^t) \) is given to fit the target regression tree \( r_{ik} \).

3. Each \( f_k(I, \hat{S}^t) \) result is updated from the last calculation.

\[
f_k(I, \hat{S}^t) = f_{k-1}(l_{int}, \Delta s_i^{(t)}) + \nu g_k(I, \hat{S}^t)
\]  
(4)

At last, the \( r_t(I, \hat{S}^t) \) is learned as the final output result of \( f_K(I, \hat{S}^t) \).

\[
r_t(I, \hat{S}^t) = f_K(I, \hat{S}^t)
\]  
(5)

After the \( r_t(I, \hat{S}^t) \) is learned, the expressive feature points can be extracted by the iterative calculation from the captured face image. The experimental effect of extracting facial feature points by the landmark method is shown in figure 2(a). The number of facial feature points is indexed in figure 2(b).

(a) Extracted feature points  
(b) Indexed number of feature points

**Figure 2.** Extracting facial feature points.

### 2.2. Facial expression classification

Facial expression classification is very important in a wide range of applications [8-10]. Convolutional neural network [11] is a kind of deep learning algorithm, which has become a research hotspot in the field of image understanding. In this paper, the second max pooling layer is used to reduce the number of parameters. This reduces the computational complexity of the deep learning network, while the performance reduction is very small. The overview of the deep learning network architecture is shown in figure 3. After designing of the deep learning network, data training is performed based on the network structure. Then the facial expression classification can be fulfilled based on the training result of this deep learning network.
Figure 3. Overview of the deep learning network architecture.

For comparison, experiments were performed using the CNN method and the SVM method respectively. Here, three facial expressions of happy, peace and anger are categorized. Experimental results show that the CNN method can obtain more accurate classification results for expression classification. Sometimes the classification by the SVM method is inaccurate. The following is an example. Among them, figure 4 (a) are the result graphs by the CNN method. Figure 4 (b) are the result graphs by the SVM method. As shown in figure 4 (b), the actual expression is angry, but the classification result is peace. Considering that the CNN method is superior to the SVM algorithm for expression classification, the deep learning based on the CNN is used in the proposed method.

(a) Expression classification using CNN

(b) Expression classification using SVM

Figure 4. Expression classification results.

3. Dynamic modeling of 3D face

3.1. Establishment of 3D expression model library
The Candide-3 model includes 113 feature points and 168 triangular meshes. It is quite detailed to introduce information about the position of the eyes, mouth and nose, but does not include the parameter positioning of the teeth and tongue. Each vertex in the model corresponds to 3 coordinate
values $x, y, z, P_i = (x_i, y_i, z_i)$. According to the statistical experience, three neutral Candide-3 facial expression models are established by manually moving the positions of the Candide-3 model vertexes. The three neutral Candide-3 facial expression models are shown in figure 5.

Figure 5. Candide-3 facial expression models.

3.2. 3D face personalized deformation based on RBF

In order to deform the face mesh model, some feature points should be extracted out from the captured image and their displacements between the positions in the image and the in the neutral expression model should be calculated firstly. Then, the positions of the $n$ feature vertexes in the neutral expression model are moved according to these displacements. Next, the positions of the $m$ non-feature vertexes in the neutral expression model are moved according to the calculation results of the Radial Basis Function (RBF) based on a spatial interpolation function.

In theory, any function that satisfies the requirements of radial symmetry can be used as an interpolation basis function. The general choices are Gaussian function, Hardy quadratic function, and thin-spline function and so on. In fact, the Gaussian function is used in data interpolation algorithm more widely. And it is reliable for data interpolating in the high-dimensional space. Hence, the Gaussian function in equation (6) are used in the proposed method.

$$g(r) = e^{-\left(\frac{r}{r_i}\right)^2}$$ (6)

In equation (6), $r$ is the radius of influence. The Gaussian function $g(r)$ will gradually approach 0 with the increasing of $r$.

The equation (7) is the radial basis function for calculating the displacements of those non-feature vertexes in the neutral expression model.
\[ f(\tilde{x}) = \sum_{j=1}^{n} c_j g\left(\|\tilde{x} - x_j\|\right) \]  

(7)

In equation (7), \( \tilde{x} \) is the original position of a non-feature vertex in the neutral expression model, \( g() \) is the Gaussian function, \( c_j \) is the coefficient of the Gaussian function \( g() \), \( x_j \) is the known original feature vertex position, and \( n \) is the total number of the feature vertexes. \( c_j \) is calibrated by the known positions of those feature vertexes before and after an expression in the Candide-3 model.

The radial basis function network is a three-layer forward feedback network [10]. The input layer is consisted of an \( m \)-dimensional input vector. The intermediate layer is consists of the calculated Gaussian function values. And the output layer is consists of \( n \) linear elements.

The calibration process of \( c_j \) is as follows.

For the \( n \) feature vertexes, assuming the displacement before and after the movement in the neutral expression model is \( \Delta p_i \), \( \Delta p_i = p'_i - p_i = f(p_i) \) \( (1 \leq i \leq n) \),

\[ \Delta p_i = \sum_{j=1}^{n} c_j \phi\left(\|p_i - p_j\|\right) + Mp_i + t \quad (1 \leq i \leq n) \]  

(8)

And the affine transformation should satisfy these constraints:

\[ \begin{cases} \sum_{j=1}^{n} c_j = 0 \\ \sum_{j=1}^{n} (c_j \cdot p_j) = 0 \end{cases} \]  

(9)

The above two equations can be combined to obtain a linear equation as equation (10).

\[
\begin{bmatrix}
\phi_{11} & \phi_{12} & \ldots & \phi_{1n} & P_{1x} & P_{1y} & P_{1z} & 1 & c_1 & \Delta p_1 \\
\phi_{21} & \phi_{22} & \ldots & \phi_{2n} & P_{2x} & P_{2y} & P_{2z} & 1 & c_2 & \Delta p_2 \\
\vdots & \vdots & \ldots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\phi_{n1} & \phi_{n2} & \ldots & \phi_{nn} & P_{nx} & P_{ny} & P_{nz} & 1 & c_n & \Delta p_n \\
P_{1x} & P_{2x} & \ldots & P_{nx} & 0 & 0 & 0 & 0 & M & 0 \\
P_{1y} & P_{2y} & \ldots & P_{ny} & 0 & 0 & 0 & 0 & 0 & 0 \\
P_{1z} & P_{2z} & \ldots & P_{nz} & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & \ldots & 1 & 0 & 0 & 0 & 0 & 0 & 0 
\end{bmatrix}
\begin{bmatrix}
c_1 \\
c_2 \\
\vdots \\
c_n \\
M \\
r \end{bmatrix}
= \begin{bmatrix}
\Delta p_1 \\
\Delta p_2 \\
\vdots \\
\Delta p_n \\
0 \\
0 
\end{bmatrix}
\]  

(10)

In equation (8) and (10), \( \phi_{ij} = \Phi\left(\|p_i - p_j\|\right) = g\left(\|p_i - p_j\|\right) \) \( (1 \leq i, j \leq n) \) is the Gaussian function. \( (P_{ix}, P_{iy}, P_{iz}) \) \( (1 \leq i \leq n) \) are the coordinates of the feature vertex \( p_i \). The coefficients \( c_j \) of the Gaussian functions, the affine parts \( M \) and \( t \) can be solved by the linear system of equation (10).

After the coefficients \( c_j \) of the Gaussian functions, the affine parts \( M \) and \( t \) are solved, the corresponding displacements of all non-feature vertexes can be calculated as equation (8). Binding textures can make the model look more realistic. The OpenGL binding texture functions are used here for the reality of the 3D face personalized graphics.

4. Experimental results and analysis

4.1. Dynamic modeling experiment results

A 3D facial expression animation processing system is implemented using the method proposed in this paper. Figure 6 shows the results of the system. The corresponding 3D face model can be re-
constructed with the operator's expression changing. It can be seen that the method proposed in this paper can provide a realistic 3D face model.

(a) Peace expression  (b) Happy expression  (c) Happy expression (Rotate 30° to the right)

Figure 6. 3D facial expression animation processing system

4.2. Time performance experiment results
In order to further evaluate the time performance, the run times of different periods are tested during the running of the 3D facial expression animation processing system. The system is implemented in C++ and run on a PC with an Intel(R) Core(TM) i7-4770 CPU @3.40GHz and 4G memory. The graphics card is Intel(R) HD Graphics 4600. The whole processing is divided into 4 periods. They are the face detection + feature extraction, classification + emotion detection, RBF computing, and vertex position updating + model rendering. Table 1 shows the running times of each period in the system processing. The time for RBF computing is a little longer. However, it can be done offline for the facial expression animation generation.

| Type                                           | Time taken/proposed method |
|------------------------------------------------|----------------------------|
| Face detection + Feature extraction            | 0.066                      |
| Classification + Emotion detection             | 0.026                      |
| RBF Computing                                  | 4338                       |
| Vertex position updating + Model rendering     | 0.088                      |

Table 1. Time performance experiment result (sec).

4.3. Comparison of face modeling
In order to verify the effectiveness of the proposed algorithm based on facial expression classification and feature extraction, a comparative experiment on modeling effect is carried out in this paper. In the same experimental environment, for different expressions (happiness, peace, angry respectively), the experimental results of Candide-3 deformation model by the method proposed in this paper is compared with the experimental results of face models invoked from the 3D expression model library.
The experimental results show that the proposed method is more realistic. The movements of the vertexes in the Candide-3 model is more accurate.

![Figure 7. Comparison of modeling effects.](image)

As shown in Figure 7, sub-figures (a) ~ (c) show the expression images of happiness, peace, and angry. Sub-figures (d) ~ (f) represent the corresponding modeling results invoked from the 3D expression model library. Sub-figures (g) ~ (i) present the modeling results by the method proposed in this paper.

5. Conclusion

This paper proposes a method combining deep learning and feature extraction for face dynamic modeling. After the face feature points are extracted and the facial expression is classified by the CNN, the 3D face model is personalized re-constructed based on the 3D neutral face model to complete a 3D facial expression animation system. The experimental results show that the proposed method can more accurately re-construct the face models with the different expressions.
Acknowledgments
The authors would like to acknowledge the support from the National Natural Science Foundation of China (61371142) and the Beijing Natural Science Foundation (4194076, 4182018).

References
[1] Y. Li, Y. Zhao, H. Yao, and Q. Ji, "Learning a discriminative dictionary for facial expression recognition," 2015 International Conference on Affective Computing and Intelligent Interaction (ACII), IEEE, pp. 838-844, 2015.
[2] P. Cui, and T. Yan "A SVM-Based Feature Extraction for Face Recognition," International Conference of Pioneering Computer Scientists, Engineers and Educators, pp. 120-126, 2015.
[3] P. Sarakon, T. Charoenpong, and S. Charoensirivath, "Face shape classification from 3D human data by using SVM," The 7th 2014 Biomedical Engineering International Conference, pp. 1-5, 2014.
[4] HS. Devi, R. Laishram, and DM. Thounaojiam, "Face recognition using R-KDA with non-linear SVM for multi-view database," Procedia Computer Science, vol. 54, pp. 532-541, 2015.
[5] QQ. Tao, S. Zhan, XH. Li, and T Kurihara, "Robust face detection using local CNN and SVM based on kernel combination," Neurocomputing, vol. 211, pp. 98-105, 2016.
[6] M. H. Selamat and H. Md Rais, "Image face recognition using Hybrid Multiclass SVM (HM-SVM)," 2015 International Conference on Computer, Control, Informatics and its Applications (IC3INA), pp. 159-164, 2015.
[7] V. Kazemi, and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1867-1874, 2014.
[8] Z. Xie, F. Yang, Y. Zhang, C. Wu, "Face sketch recognition based on edge enhancement via deep learning," LIDAR Imaging Detection and Target Recognition 2017, vol. 10605, pp.106053N, 2017
[9] RK. Yadav, and AK. Sachan, "Application expansion inside optimized RBF kernel of SVM in robust bace recognition system," International Journal of Signal Processing, Image Processing and Pattern Recognition, vol. 8, no. 12, pp. 89-98, 2015.
[10] XH. Wang, C. Xia, M. Hu, and FJ. Ren, "Facial expression recognition under partial occlusion based on fusion of global and local features," Ninth International Conference on Graphic and Image Processing (ICGIP 2017), vol. 10615, 2018.
[11] A. Krizhevsky, I. Sutskever, and GE. Hinton, "ImageNet classification with deep convolutional neural networks," Communications of the ACM, vol. 60, no. 6, pp. 84-90, 2017.