

3D Expressional Head Creation System for Mobile Game Platform

Jiejie Zhu¹⁾, Zhigeng Pan¹⁾, David Andrain Cheok²⁾, Shawchoong Peng³⁾

¹⁾ State Key Lab of CAD&CG, Zhejiang University, Hangzhou, P.R.China

²⁾ Interaction and Entertainment research center, Nanyang Technological University, Singapore

³⁾ Fllage company, Singapre.

Contact mail: {zhujiejie,zgpan,xuguilin}@cad.zju.edu.cn

Abstract

This paper introduces a novel system of creating 3d human head models with facial expressions using one front view image taken by camera-ready phones. 3D head models can be downloaded and rendered on common 3G mobile phones. With only one front view image, our system can recognize human face regions using skin color possibility, and calculate facial feature points using gradient changes. Head model modification and texture mapping are implemented via two default generic 3D head models. 6 basic facial expressions are created based on expressional templates. Using Mobile 3D API (JSR-184), this system can render high quality 3D geometries on mobile phones with facial expressions. Constraints and potential resolutions are also described.

Keywords: Facial Expression, Texture Mapping, Mobile 3D, Gaussian Distribution.

1. INTRODUCTION

Phones are built for human-human's interaction over long years. The fact that people can only hear each other when the connection is established exists for decades. However, multimedia techniques change this situation by enriching the connection content. Such implementation used to be 2D and displayed only in mono color on mobile phones. 3D graphic rendering techniques developed in recent years for mobile phones help to implement 3D games on mobile phones. Several companies are leading techniques for real applications, such as mobile 3D framework design in Hybrid Graphics [1], mobile 3D game in Superscape [2]. Low level 3D mobile engine are also developed such as OpenGL ES, OpenMax, OpenML, Open VG [3], etc.

With the introduction of the 3G networks and the fast developing 3D graphics on mobile devices, pervasive games and human-human interactions will become more widespread and more powerful. Thus, 3D will be a stepping-stone to a console-like experience on mobile phone. And games and human-human interaction applications will be in the lead to take advantage of these new capabilities.

In most of games and interactive applications, a representation of oneself is important. In single player games, this representation comes in the form of the story hero/heroine. In multiplayer games, players have the power to uniquely change the statistics of the characters, but the pictorial representations and personal emotions are of few choices. Realistic facial expression is among the most powerful, natural, and immediate means for people to communicate their emotions and intentions. A 3D mirror of oneself enhanced with facial expressions will attract most users for entertainment.

Nowadays, mobile 3D is evolving but is still at a fairly embryonic phase in its development today. Several limitations exist for implementing 3D animation.

(1) No standard 3D engines that all phones support

Although OpenGL ES is the standard mobile 3D specification, there are a number of 3D APIs created on this base by different companies, ranging from the famous Hi Corp's Mascot Capsule API to new API's like the JSR-184 and even proprietary API like the Motorola 3D API. This causes a compatible problem for developers whose works cannot be applied to different mobile phones. This also delays the project develop time for specifying the specified type of mobile phone that developers need to use the phone built in APIs.

(2) Low memory and processor speed

Low processor speed and lack of memory in mobile devices only support low-polygon and simple models in unacceptable quality. The challenge of implementing real time rendering on mobile phone is located to balance between quality and performance. Many of tests have to be conducted among various phone models. Such tests help developers decided how many polygons should application supports. Judging the balance is difficult and also costs much of time.

(3) Delivery methods

Bandwidth and computing powers of mobile handsets are important for wireless communication. Centric server based technique have to be used for processing complex computing. For 3D graphic applications, the server acts as the computer that will render the 3D face model with associated animations and emotions. It will also compress all the data and coding them using the phone's video codec. And then it will send them over the network to the receiving party. The overhead caused by the animation and compression to a video stream will cause considerable performance degradation. The problem is that phones do not provide an API layer for talking with the phone's video coder from a J2ME application.

Due to above constraints, normal techniques of rendering realistic 3D human head models and animations in PC platform can not be simply transferred to mobile platform. In this paper, we introduce our on-going work of creating a mediate-polygon 3D head with realistic facial expressions and transmit it to mobile platform. Section 2 presents related works on 3D facial models on mobile phone and it's application. Section 3 introduces our prototype architecture and basic wireless communication skills. Key

techniques are described in section 4, including face recognition, facial feature point extraction and model generation. Preliminary results will be explained at section 4. Conclusions and future work on how to optimize the resemblance and the rendering speed are introduced in the last section.

2. RELATED WORKS

3D human model and facial expression rendering techniques on mobile phone are new research topics. However, techniques in PC are not new to researchers. The main difficult is how to transfer implemented PC rendering techniques to low power, low process capacity and no rendering hardware supported mobile phones. Naoya Miyashita [4] developed a chat system using facial image synthesis. He used flash techniques to avoid different plug-in for different mobile companies since most of them support flash players. Four layers are combined to express different facial expressions using facial action coding system. However, techniques on this work are all based on 2D image processing. S.T Worrall [5] analyzed difficulties on data transferring for very low bit rate mobile video from videos, such as 3D facial expression. His work resolved efficient encoding and decoding of 3D coded data. Facial animation data are encoded and decoded using MPEG4 FAP and FDP specifications. An application of face password uses face reorganization technology via analyzing facial features from Oki Electric company [6].

ARM [7] has implemented several types of graphic hardware accelerator for mobile phones. ARM9™, ARM10E™ and ARM11™ core-based embedded systems exceed the capabilities of PCs of 1995. And ARM has achieved compliance with the OpenGL ES standard in PowerVR MBX graphics accelerator cores. By adding OpenGL ES driver support to the PowerVR MBX product line, enhanced 3D graphics performance is available. However, such high end clients are not available for our experiment at present. Thus, our work does not based on the mobile 3D accelerator.

3. PROTOTYPE

3.1 Architecture

Main functions of a user expressional 3D head creation system involve the creation and maintaining of a pictorial representation of a person, portraiture and a visual identity of a person and his/her personality. In the simplest goal of this application is enable this “cyber personality” comes to live in a 3G mobile call. We divide our system into several parts and the entire architecture is illustrated figure 1.

Phone Client This part is implemented using J2ME APIs on the mobile phone. User can use designed UI client to capture a front view photo, editing his/her head model with accessories, such as different hairstyles, skin colors.

Phone Servlet This servlet is used for communicating with phone client. It runs on the centric server with UNIX operation system. The main task of this model is receiving phone’s requests and replying them. The server is also responsible for common mobile message transmitting.

Modeler Servlet This part is also a Java Servlet program running on the centric server, but it takes charge of the communication between centric server and 3D Modeler , which will be introduced in 3.2 and we call it PC client. It sends requests for building users’ 3D head model and replies to PC client relative input data (photo and user information) to 3D Modeler and receiving the model file (including texture files) from 3D Modeler.

User Database This database locates in the centric server and stores all users’ information and data files. Only Phone Servlet and Modeler Servlet can access it and query and store user data.

3D Modeler This Modeler is responsible for 3D model building and texture production. It is the factory to build customized head model for each user.

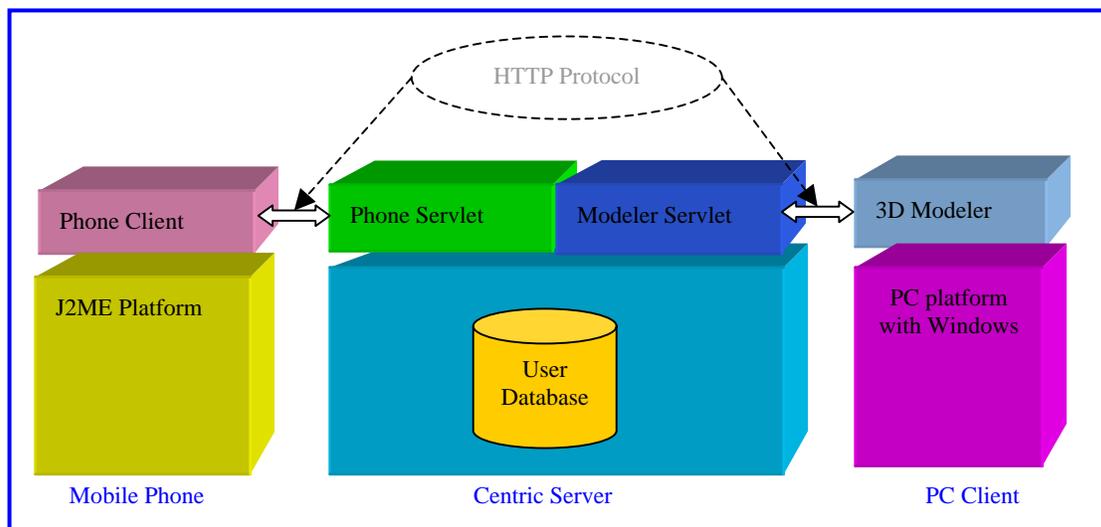


Figure 1: Client/Server Architecture

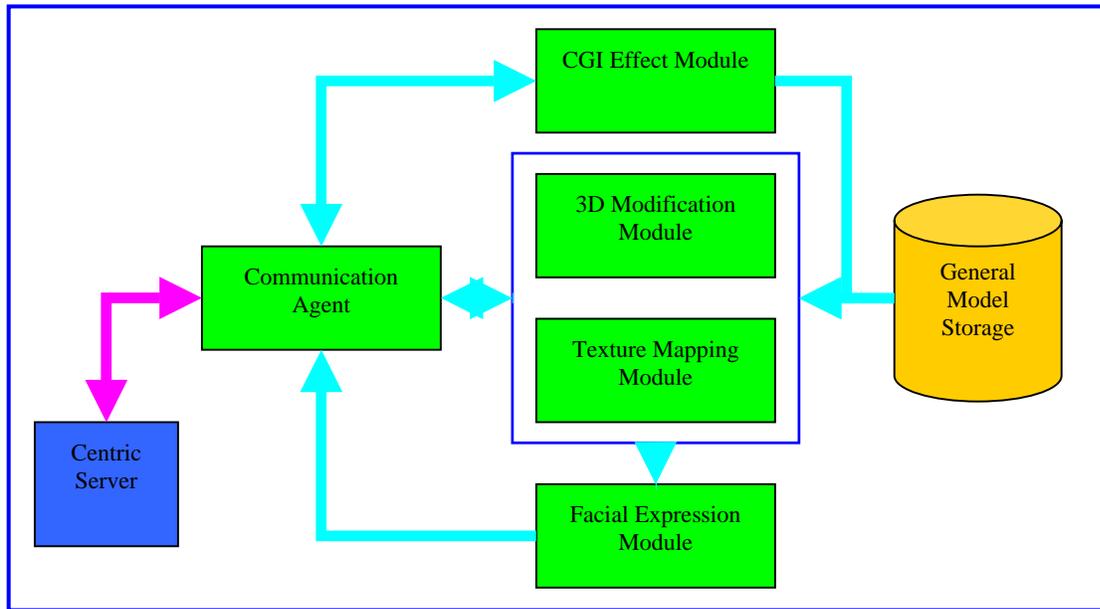


Figure 2: 3D modeler modules

3.2 Structure of 3D Modeler

3D Modeler is the key part for creating the users' expressional 3D head model. Figure 2 shows the 3D Modeler structure. Two main modules can be divided: communication model and mirror creation model.

Communication Agent This module communicates between 3D Modeler model and Centric Server model. Wireless HTTP protocol is used to send and get messages. Polling and pushing techniques are adopted. Before executing other modules, this agent should request all required data from centric server. And after creating the 3D head model, it also need send all the result data to server.

CGI Effect Module This module aims to create an out-looking nice face texture for the modified 3D head model. We take FaceGen's [8] face texture as reference. Other face texture generation methods implemented in Poser [9], Maya [10], can also be used as references. Clipping, edge smoothing, scaling image operations are implemented. Details could be found in 4.3.

3D Modification Module The modifications on general model are done in this module. At first, it should select an appropriate general model according to user gender and input coordinates of marked points. And then, several modifying operation will be executed to build the mirror model. Details could be found In 4.4.

Texture Mapping Module This module is to map new generated face texture to modified 3D model. The texture mapping with hard constraints are used to alignment the critical partitions on the texture and 3D model. For example, the mouth on the texture should be mapped to the mouth polygons on 3D model. Section 4.5 is the special one for illustrating this component.

Facial Expression Module Several general facial expressions are automatically generated based on expressional templates in our system. MPEG4's FDP and FAP specification are used. Details could be found in 4.6.

3.3 I-Message Communication

Main function of communication in our system is to do live chat with 3D avatars on different mobile phones. While engaging in voice conversation, the avatars will act as a representative of respective users. Via 3G network, users have 3D avatars based on present video call platform. Things could happen like this:

To establish a 3D call, caller has to send a request message to the receiver via server. Once the receiver confirms this 3D call, they download the opposite model from server or from end client (caller and receiver). At the same time, caller would initiate a voice call over the standard voice network. Once the voice call is established, the models will be able to extract the voice stream and perform other animations. Emoticons and hand gestures can be sent over to the other channel via the server in forms of text-based parameters. Receiver will decode those parameters and animate the models accordingly.

Real-time computing is implemented while on a voice call with other connection channels, which will ensure application could finish required task lists.

4. KEY TECHNIQUES

4.1 Facial Points Detection

Two steps are taken using image processing toolkits for automatic facial point detection. First is to recognize the face region on the input image. There are lots of face recognition algorithms implemented on PC platform [11]. Among them, calculate the possibility of skin pixel is used most widely. This is based on the research results that the color distribution of human skin colors of different people was found to be clustered in a small area of the chromatic color space. And the distribution can be regarded as a Gaussian distribution. The likelihood of a skin pixel is computed according to:

Likelihood :

$$P(r, b) = \exp[-0.5(x - m)^T C^{-1}(x - m)]$$

where: $x = (r, b)^T$

C is the covariance matrix of Cb to Cr . And m stands for the mean vector of Cb and Cr . C and m can be calculated by the skin samples [12].

Second is to locate the facial points in the detected face region. Face Recognition module generates a rectangle of face region. Extraction method [13,14,15,16] calculates the horizon and vertical gradient changes first, and then algorithm finds out the largest sum of the row and column by specified face region template. The sum pixel stands for the center of the feature region. Since the front face will not be symmetric due to users' photo taking process, for each half face, the algorithm needs to calculate the center for eyes. Figure 3 shows the results.

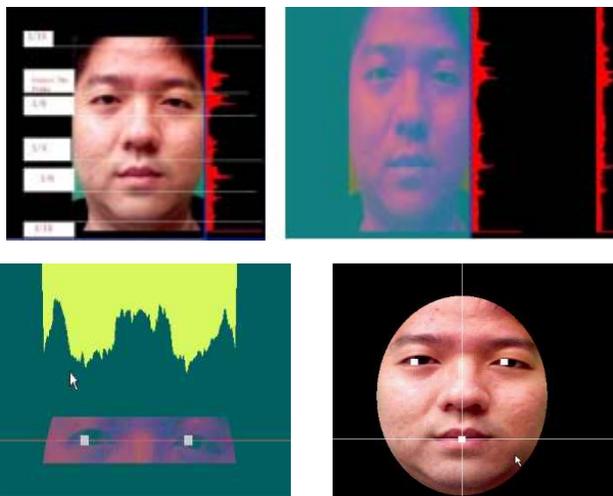


Figure 3: Eye and mouth center detection using our template in 4 Steps. (1): Template matching. (2): Left side and right side gradient calculation. (3): Eye gradient calculation in horizon and vertical. (4): Detected eye and mouth center according to our template.

4.2 User interface for marking facial points

Images taken by mobile camera are not in good quality and in high resolution. The common resolution obtained by native camera is 640*480, but if we use J2ME program, we can only obtain 160*120 size. Such resolution and quality far from the accuracy fact for facial points detection described in 4.1. Thus, a client application is designed for users selecting facial points. Figure 4 shows the marking points interface. The image on the right bottom corner is the reference image which is used to assist users' feature point selection.

Some results comparing have been made between automatically facial point generation and user marking. For high resolution image, the former one is much more preferred since no manpower needed to interfere. But for accuracy and computing speed, the latter one is preferred.



Figure 4: User marking points interface

4.3 CGI Texture Generation

Before mapping to modified 3D model, the face texture taken by mobile camera is processed by several operations. First, clip operation should be executed to achieve the face region according to facial points. We use an ellipse approximation method to get the main region of the face. Second, re-sampling method should be implemented to resize the image as the mapping module requested. Thirdly, we need generated a compatible background for wrapping the other head. We just use the mean color of the face region because the result is quite good since the mobile phone screen size is small. At last, the face region should be merged smoothly with the background. An ellipse-based interpolation method for alpha channel is implemented to smooth the edge of the two merged images. Figure 5 shows a sample of the results.



Figure 5: Regenerated face texture. Left side using 160*120 size and right side using 640*480 size.

4.4 3D Head Modification

To modify a 3D head model according to images, we adjust all key feature points and do an interpolation for 3D vertex on the head wire frame. Figure 6 shows the distribution of these feature points.



Figure 6: The feature points on the user photo

As for the general model, different sized models are pre-generated according to “guess”. Some of them are used for long face and others are used for short face. The reason for multiple general models is that one similar basic model can be select for one user and the final result of modification is much more acceptable. Selection can be done according to the outline of face gotten from the data of marked outline points. For example, if user’s face is very long, then a longer general model will be selected. After the basic model being ready, modification on this model will be executed. The modification in this system is based on the feature points. Each feature points on 3D model will be moved according to the corresponding feature points inputted from user face photo. The non-feature vertices on 3D model are also moved by the weight-driven methods.

4.5 Texture Mapping

Texture mapping is important to evaluate the result effect. We use the cylindrical texture mapping [17] to project the 3D coordinates of model onto a cylindrical surface. Then imaging the texture on cylindrical surface based on feature points, which is called hard constraints that extracted in 4.3. Non-feature points are interpolated using 2D interpolation method [18] for scattered data points. Figure 4 shows an example of hard constraints mapping.

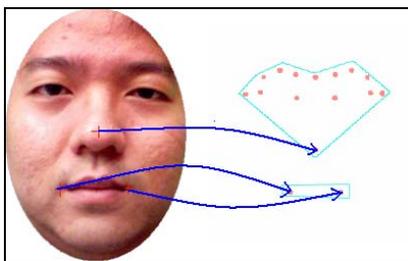


Figure 7: Texture mapping using hard constraints

4.6 Facial Expression

We use facial animation parameters (FAPs) to implement facial expressions. These facial animation parameters, which displace feature points from neutral position, allow personalization in a scalable manner. Then facial expressions can be expressed by these FAPs. For example, we can define joy expression as: the eyebrows are relaxed. The mouth is open and the mouth corners pulled back toward the ears. In this system, we use morphing methods for facial animation. There are three steps:

- (1) Get the source points come from the neutral position model,
- (2) Get a target facial expression from the neural 3D model manually,

- (3) Transform rest of the points in the facial mesh by using the following distance measure:

$$d(x) = \sum_{i=1}^N h(\|x - x_i\|)$$

where h is the Gossi distance [19] between a feature point and an input point, N is the number of feature points.



Figure 8: Generated facial expressions. Left side is sad using a high resolution and right side is smile using a low resolution.

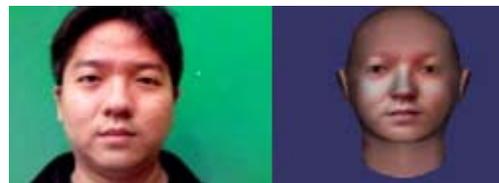
5. EXPERIMENT RESULTS

Figure 9 shows mobile client renders 1830 polygons 3D head model with 6 facial expressions and 6 textures. The mobile phone we are using is Sony Ericsson Z800. The screen size is 176*220. Memory is 64M and the resolution of the camera is 1.3 mega.



Figure 9: Six facial expressions render on mobile phone. The model used here is downloaded from FaceGen download center.

Figure 10 shows three users’ photo and their generated 3D head model.



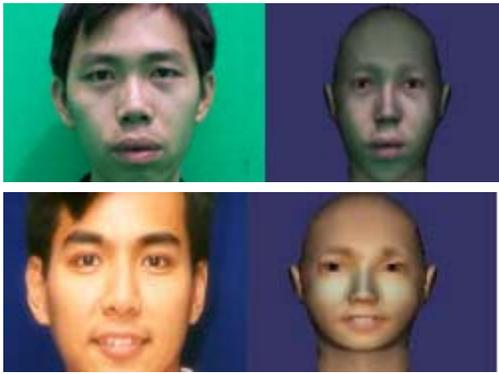


Figure 10: Three rendered results. The different color of the face image influences the final results of the model. This feature is kept because users want to build different models according to different face images. For example, they can photo it at night with rich lights, the effects will much different with the normal one.

6. CONCLUSION

In this paper, we describe our methods for creating human facial expressions automatically using a general 3D head model and render them on mobile phone using mobile 3D API. Some limitations of current facial feature point extraction prototypes are:

- No inclination is preferred.
- No long hair is preferred
- No glasses and thick moustache are preferred
- Large skin samples are preferred

Face feature points can not be stably extracted due to the image quality taken by camera-phone and the robustness of the algorithm do not satisfy the baseline for feature point extraction. We'll change this part to let users mark facial feature points and users can assemble their own face. They can choose different shapes and get a funny looking face, which is useful for market improving. Face editor can also be done on website through the Internet. And with CGI effects, users can get a nice-looking 3D head via talking with others. This screen will improve the traditional phone's idea for interacting with each other face to face.

ACKNOWLEDGEMENTS

This project is co-supported by Key NSF Project on Digital Olympic Museum (Grant no: 60533080). We'd like to give thanks to Dr. You Kin Choong, Hongwei Yang, Guilin xu, Sucianto Prasetio, Seah Peck Beng. Their contributions are invaluable for this project.

7. REFERENCES

[1] Hybrid Graphics Company. <http://www.hybrid.fi>
 [2] Superscape Company. <http://www.superscape.com>
 [3] Khronos organization. <http://www.khronos.org>
 [4] Naoya M.; Takashi S.; Norihiko M.; Tadashi N.: Development of a Chat System Using 3D Facial Image Synthesis on Flash. In proceedings of ICEC. (2005)

[5] Worrall S.T.; Sadka A.H.; Kondoz A.M.:3-D facial animation for very low bit-rate mobile video. In proceedings of 3rd International conference on 3G Mobile Communication Technology. (2002) 371-375.
 [6] Vodafone company www.vodafone.co.uk/
 [7] ARM Company www.arm.com
 [8] FaceGen Software. <http://www.facegen.com/>
 [9] Poser Software. www.e-frontier.com/
 [10] Maya Software. www.autodesk.com/alias
 [11] Gong, Y. and Sakauchi, M: Detection of regions matching specified chromatic features, Journal of Computer Vision and Image Understanding, 61(2) (1995) 263-269
 [12] Face Detection Project. www.cs.cmu.edu/~har/faces.html
 [13] Chang, T.C.; Huang, T.S.; Novak, C.: Facial feature extraction from color images. Proceedings of the 11th International Conference on Pattern Recognition. (1994) 39-43
 [14] Wang Y.Q.; Liu Z.F.; You Z.S; Jain, A.K.: Face detection and facial feature extraction in color image. Proceedings of Fifth International Conference on Computational Intelligence and Multimedia Applications. (2003) 126-130
 [15] Zhong X.; Li, S.Z.; Eam K.T.: Facial feature extraction and image warping using PCA based statistic model Proceedings of International Conference on Image Processing. (2001) 689-692
 [16] Wu, H.Y.; Chen, Q.; Yachida, M.; Facial feature extraction and face verification Proceedings of the 13th International Conference on Pattern Recognition. (1996) 484-488
 [17] Eric A. B.; Kenneth R. S.: Two-Part Texture Mapping. IEEE Computer Graphics and Application. 6(9) (1986) 542-547
 [18] Isaac A.: Scattered data interpolation methods for electronic imaging system: a survey, Journal of Electronic Image. 11(2) (2002) 157-176
 [19] Noh J.; Fidaleo D.; Neumann U.: Animated deformations with radial basis functions. In proceedings of the ACM Symposium on Virtual Reality Software and Technology. (2000) 166 174

About the authors

Jiejie Zhu is a Ph.D. student at State Key Lab of Computer Aided Design and Computer Graphics, Zhejiang University, P.R.China. His contact email is zhujejie@cad.zju.edu.cn

Zhigeng Pan is a full professor at State Key Lab of Computer Aided Design and Computer Graphics, Zhejiang University, P.R.China. His contact phone numbers is: 86-571-88206681-509. His contact email is zgpan@cad.zju.edu.cn

David Andrain Cheok is an associate professor at Interaction and Entertainment research center. He is also the director of this research center. His contact email is adraincheok@ntu.edu.sg

Shawchoong Peng is the manager of Fllage Company. His contact email is sc@fllage.com