International Journal of Electronic Commerce Studies Vol.4, No.2, pp.203-212, 2013 doi: 10.7903/ijecs.1128

REALISTIC 3D FACE MODELING AND APPLICATION RESEARCH

Shuqin Li

Computer and network management center NorthChinaUniversity of Technology, Beijing, China shuqin66886@163.com

Huibai Wang College of Information Engineering North China University of Technology, Beijing, China wanghuibai@gmail.com

ABSTRACT

At present, methods of face modelling are not satisfactory, requiring high costs and a complex algorithm. We present an algorithm for 3D specific face modelling from a front view image of a person's face. First, we globally reshape the 3D neutral face model by constructing s linear elasticity model grid. Then, local deformation is performed on the facial feature vertices to obtain a more realistic 3D model for the person. Finally, we apply texture mapping to the deformed face model with cylindrical projection to produce a realistic 3D specific face.

Keywords: Feature Point Extraction, Linear Elastic Model, Feature Point Edit, Texture Mapping

1. INTRODUCTION

Face modeling can be roughly divided into three types of methods: parameter model, physiological muscle model and visual modeling based on image. Park used the face parameter model to generate the face model; Platt and Water used the muscle model to build a virtual face; Horce generated a face model with two pieces of orthogonal photos; and Blanz put forward a face modeling method based on statistic and induction¹. Among these methods, visual modeling has many advantages, including its relatively simple process, accessible face data, and stronger sense of reality. As a result, we choose the visual modeling method using a front face photo to realize fast personalized face modeling. Our main researchfocus on face feature point extraction, feature points editor and texture mapping.

2. FACE FEATURE POINT EXTRACTION

Face feature point extraction is the first step of 3D face reconstruction. We adopt a three-step face detection algorithm based on skin color. The algorithm starts by calculating the similarity information and obtaining the binary image in order to locate the face region. Then, it applies feature extraction to the face region and obtains the specific position of the eyes, nose and mouth.

2.1 Face Region Detection

Based on the method of similarity degree, we define r = R/(R + G + B), b = B/(r + G + B), which can reduce the three-dimensional RGB to two dimensions. Based on Gauss distribution, we first use the training method to calculate the Gauss distribution center². Then, we obtain the desired similar distribution according to the distance between the request pixel and the distribution center. Finally, we apply binarization to the distribution and locate the face region. In the training set, we need to determine the mean M and variance C. Formulas are defined as follows:

$$M = E(x), C = E((x - M)(x - M)^{T}), x = [r, b]^{T}$$

In the similarity calculation process, we use the formula:

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

2.2 Face Feature Extraction

To make full use of priori knowledge, the eyes and mouth are expressed with a piecewise polynomial curve. We use deformation template to obtain an accurate outline.

(a) Eye contour extraction



Figure 1. Eye contour model

The eye contour model shown in Figure 1 is made of four sections of quadratic curves. The curve have eight parameters: the center of the eye (x_0, y_0) , the angle between two lines connected inside and outside canthus θ , the height of upper eye h_1 and lower eye h_2 , the distance between inner canthus and upper eye center w_1 , the distance between outside canthus and upper eye center w_2 , and the distance between the inner canthus and lower eye center w_3 . Expressions of the four sections of curve are defined as follows:

$$y = h_{1} \left(1 - \frac{x^{2}}{w_{1}^{2}}\right) W_{1} \le x \le 0$$

$$y = h_{1} \left(1 - \frac{x^{2}}{w_{2}^{2}}\right) \qquad 0 \le x \le w_{2}$$

$$y = h_{2} \left(\frac{\left(x + w_{1} - w_{3}\right)^{2}}{w_{3}^{2}} - 1\right) \qquad -w_{1} \le x \le w_{3} - w_{1}$$

$$y = h_{2} \left(\frac{\left(x + w_{1} - w_{3}\right)^{2}}{\left(w_{1} + w_{2} - w_{3}\right)^{2}} - 1\right) \qquad w_{3} - w_{1} \le x \le w_{2}$$

Eyes do not have consistent color information, but edge information is abundant. First edge extraction is performed and then appropriate closure operation, which form a high brightness area inside the eyes. In the processed image, the eye interior is a high brightness area and the eye external a low brightness area. The evaluation function can be defined as follows:

$$\min C = \oint_{\partial D^+} I(x) dx - \oint_{\partial D^-} I(x) dx$$

where D represents the eye area, ∂D^+ is the band area outside of the eye contour, and ∂D^- is the band area inside the eye contour. According to the above four sections of curve expressions, we can initialize the eye contour and then use the steepest descent iteration arithmetic until the outline converges.

(b) Mouth contour extraction

The contour model and curve expressions of the mouth are similar to those of the eyes, so we give only an evaluation function here. Because internal lip color is almost consistent (assuming a closed mouth), we can distinguish the mouth area by segmenting lip color and skin color. The evaluation function is expressed as:

$$\min C = \oint_{\partial D^-} I(x) dx - \oint_{\partial D^+} I(x) dx$$

3. FACE MODELING

After feature point extraction, we can deform the general 3D face model to a specific human face. In this paper, we use a polygon to present the face modeland define the eyes, nose, forehead, eyebrows and jaw feature points, since such feature points define the person's face shape and position. The positions of non-feature or free points are influenced by the feature points.

3.1 Construct Linear Elasticity Model

In the construct linear elasticity model, a group of feature points needs to move with the corresponding non-feature points; the whole move must accord with elastic characteristics³. When the feature point is moved back to its original position, the non-feature point can also be restored to its original position. We present a smooth interpolation function which can calculate the displacement vector of each vertex.

Given a set of known feature points displacement $\mu_i = P_i - P_i(0)$, a function is constructed which can determine each non-feature point's displacement μ_j . We use the following formula:

$$f(p) = \sum_{i} c_{i} \phi(||p - p_{i}||) + Mp + t$$
(1)

Here, ϕ is the radial basis function, affine component M is a 3×3 matrix and affine component t is a 3×1 vector. Adopting the above method, we not only ensure minimized grid curved surface energy but also resolving expression.

$$\begin{cases} \mu_{i} = f(P_{i}) \\ \sum_{i} c_{i} = 0 \\ \sum_{i} c_{i} p_{i}^{T} = 0 \end{cases}$$
(2)

By computing the above equations, we get values of c_i , M and t, then get the interpolation function ϕ .

In practice, considering all grids as an elastic mesh is not reasonable. If the eye feature points are moved, then allvertices on the mouth may follow. Moreover, the method requires that, when moving a feature point, the position of other feature points cannot be changed⁴.

In this paper, we present an improved algorithm that divides the whole grid into small elastic meshes according to different function divisions. For each non-feature point, according to its area, two or four control points are fixed in the X and Y direction. The control points must be feature points, and the ratio of distance between free points and control points in one direction must be unchanged. No matter how the face former shapes, the relative position of avertex to the surrounding feature area (such as eyes, nose, and chin) will remain unchanged.

3.2 Feature Point Editor

The revision process from a neutral face model to a specific face model includes three transforms. First, take overall transformation to a neutral face model matching with the location of general facial features. The purpose is to achieve whole facial profile modification. Second, in order to make local modification, take local transformation to the transformed neutral face model according to the shape and size of face organs such as eyebrow, eye, mouth and nose. In order to accurately simulate facial characteristics, we need an accurate adjustment to the modified face model.

Obtain the smallest cube of face height h, width w, and depth d in pixel units which contains the original model. We obtain a certain face model according to the following steps.

(a) Create a rectangle on the front photo which contains the smallest

possible face region. The size of the rectangle is $h_f \times w_f$, and the height and width of the minimum specific head cube is $h' = h_f, w' = w_f$, respectively.

(b) Transform all points of the original model, $x' = x \cdot w'/w$, $y' = y \cdot h'/h$. Then, project the front grid of the face model onto the front photo. (c) Adjust the face model to match the photograph. The adjust process uses the linear elastic algorithm mentioned above; i.e., when a feature point moves, all other free points move following the elastic characteristic. Finally, achieve the x, y coordinates of each 3D face vertex and then construct the specific 3D face model. The deformed face model is shown in Figure 2.



Figure 2. 3D face model feature point editor result

4. TEXTURE MAPPING

To make the rendering result more realistic, we must give texture information to the 3D face model. Besides face model reshaping, we extract the two-dimensional texture coordinates of the model. The texture coordinates are the points that the face model vertices project on the photograph.

4.1 Cylindrical Projection



Figure 3. Cylinder mapping principle

Use the front photograph to render the face model, and adopt cylindrical projection to spread the rendering result out into a new map⁵. The principle of cylindrical projection is shown in Figure 3. Project a ray after point P from the cylindrical axis. The cylindrical projection coordinate of point P is the intersection of the ray with the cylindrical surface, and the intersection coordinate is $(\mu, \nu)^6$. The coordinate of point

$$\mu = \begin{cases} \arcsin(|x|/\sqrt{x^{2} + z^{2}}) & x \le 0, z \le 0\\ \pi - \arcsin(|x|/\sqrt{x^{2} + z^{2}}) & x \le 0, z > 0\\ \pi + \arcsin(|x|/\sqrt{x^{2} + z^{2}}) & x < 0, z \ge 0\\ 2\pi - \arcsin(|x|/\sqrt{x^{2} + z^{2}}) & x < 0, z < 0 \end{cases}$$

$$\nu = y \qquad (3)$$

^{*P*} is ^(*a*,*b*) in photo ^I. Hypothesis of the radius of the cylinder is *r*, if z = -r, x = 0, then $\mu = 0$. To the point P(x, y, z), we can compute the texture coordinate ^(μ, ν) as follows:

To facilitate processing and display, further transform (μ, ν) into (x, y) in the new $w \times h$ rectangular map, which means spreading the cylindrical surface out into a plane map.

If
$$\mu = 0, x' = 0; \mu = 2\pi, x' = w;$$

$$v = -r, y' = 0; v = r, y' = h$$

Then, we can obtain:

$$x' = w \cdot \mu / 2\pi, y' = (v + r) \cdot h / 2r$$
 (4)

Using (3) and (4), we can determine all texture coordinates of the 3D face model projected on the photo. The final texture mapping effect is shown in Figure 4.



Figure 4. 3D face model texture mapping result

5. CONCLUSIONS

We globally reshape the 3D neutral face model by constructing a linear elasticity modelgrid. Local deformation is performed on the facial feature vertices to obtain a specific face model. Texture mapping is applied to the deformed face model with cylindrical projection in order to make the model more realistic. The experimental results prove that the Feature Point Extraction, Feature Point Edit and Texture Mapping algorithm simple and convenient.

6. APPENDIX

This work was supported by the funding project for academic human resources development in institutions of higher learning under the jurisdiction of the Beijing municipality (PHR20100509).

7. REFERENCES

- [1] F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, and D. Salesin, Synthesizing realistic facial expressions from photographs. In S. Cunningham, W. Bransford, and Michael F. Cohen (Eds.), *Proceedings of the 25th annual conference on Computer graphics and interactive techniques* (p75-84). New York, NY, USA: ACM, 1998. http://dx.doi.org/10.1145/280814.280825.
- S.Y. Lee, K.Y. Chwa, S.Y. Shin, and G. Wolberg, Image metamorphosis using snakes and free-form deformations. In Susan G. Mair and R. Cook (Eds.), *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques* (p439-448). New York, NY, USA: ACM, 1995. http://dx.doi.org/10.1145/218380.218501.

- [3] R. Lienhart, and J. Maydt, An extended set of Haar-like features for rapid object detection. In G. Sharma and Ricardo de Queiroz (Eds.), *Proceedings of International Conference on Image Processing* (p900-933). Rochester, New York, USA: IEEE Press, 2002. http://dx.doi.org/10.1109/ICIP.2002.1038171.
- [4] M. Dimitrijevic, S. Ilic, and P. Fua, Accurate face models from uncalibrated and ill-lit video sequence. In D. Jacobs (Ed.), *Proceedings* of the 2004 IEEE Computer Society Conference (p1034-1041). Washington, DC, USA: IEEE Press, 2004. http://dx.doi.org/10.1109/CVPR.2004.1315278.
- [5] Z. Zhang, Flexible camera calibration by viewing a plane from unknown orientations. In B. Werner (Ed.), *Proceedings of the Seventh IEEE International Conference on Computer Vision*, (p666-673). Kerkyra, Greece: IEEE Press, 1999. http://dx.doi.org/10.1109/ICCV.1999.791289.
- [6] Y.J.G. Wen, Specific face synthesis based on the general face models modify. *Journal of Computer aided design and graphics*, 11(5), p389-394, 1999.