GENERATION OF 3-D HEAD MODELS FROM MULTIPLE IMAGES USING ELLIPSOID APPROXIMATION FOR THE REAR PART.

N. Grammalidis, N. Sarris, C. Varzokas and Michael G. Strintzis

Department of Electrical and Computer Engineering
University of Thessaloniki
Thessaloniki 540 06, GREECE
E-mail: ngramm@panorama.ee.auth.gr

ABSTRACT

In this paper, a system for building a three dimensional (3-D) human head model from two camera views is proposed. A generic 3-D face model is adapted to a human face of which the frontal and profile views are given. Then, a 3-D ellipsoid approximation technique is used to fit an ellipsoid to the 3-D wireframe model. This ellipsoid is then used to approximate the back of the head, while the the initial 3-D face model provides information for the high-detail front part of the head. The part of the ellipsoid model that corresponds to the back of the head is then identified and an algorithm to merge the front(face) and the back part is proposed. A cylindrical texture map is finally built covering the entire area of the head by exploiting the inherent face symmetry. The final result is a complete, textured model of a specific person's head.

1. INTRODUCTION

Estimation of the position, shape and motion of the head from image sequences has been the focus of considerable recent research effort [1, 2, 3]. In this paper, a technique for generating 3-D head models from a set of images obtained from different viewpoints is presented. In the first stage, a generic 3-D face model is adapted to a human face of which the frontal and profile views are given. Assuming that a set of feature points have been detected on both views the adaptation procedure initialises with a rigid transformation of the model aiming to minimise the distances of the 3-D model feature nodes from the calculated 3-D coordinates of the 2-D feature points. Then, a non-rigid transformation ensures that the feature nodes are displaced optimally close to their exact calculated positions, dragging their neighbours in a way that does not deform the facial model in an unnatural way. Then, a novel 3-D ellipsoid approximation technique is used to approximate an ellipsoid to the 3-D wireframe model. This ellipsoid is used to approximate the back of the head, while the the initial 3-D wireframe provides information for the high-detail front part of the head. Algorithms to identify the part of the ellipsoid model that corresponds to the back of the head and to merge the front(face) and the back part have also been developed. A cylindrical texture map is finally built from the two image views covering the entire area of the head by exploiting the inherent face symmetry. The final result is a complete, textured model of a specific person's head.

This work was supported by the GSRT projects "PAVE99" and "AMEA99".

2. THREE DIMENSIONAL MODEL ADAPTATION

The purpose of this stage is to transform a generic 3-D face model in a way that it adapts to the characteristics of an actual face observed in its front and profile image views. In theory, any 3-D model of the human face may be used. In practice however, the closer the characteristics of the model are to the characteristics of the image (e.g. model of female face for a female face) the fewer feature points will be needed for accurate adaptation. In order to project the coordinates of the feature points to the 3-D space, more information is needed than that provided by the single 2-D image frame. In the proposed scheme we utilise the combined information from two views of the face at right angles to each other. These may be obtained either by a stereoscopic system of two cameras at right angles to each other, or by a rotating single camera.

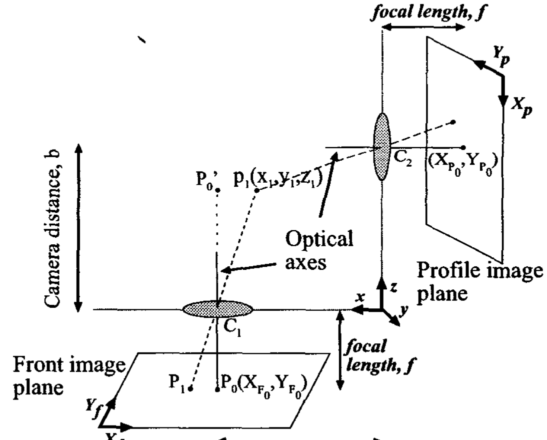


Fig. 1. Camera Geometry

Assuming the use of a perspective projection camera system, as the one shown in Figure 1, we can calculate the 3-D coordinates of every feature point:

$$X_F = f\frac{-b+x}{z}$$
$$Y_F = f\frac{y}{z} \quad (1)$$

0-7803-6297-7/00/$10.00 © 2000 IEEE

$$X_P = f\frac{b-z}{z}$$
$$Y_P = f\frac{y}{x}$$

where (X_F, Y_F) are the 2-D coordinates of the feature point on the front image view, (X_P, Y_P) the relevant coordinates on the profile image view, b the distance of the two cameras and f the focal length. The above equations yield a system of four equations for each feature point with three unknowns (the 3-D position of the feature point x, y, z). This system is solved by using least squares, to provide the 3-D position (x, y, z) of every feature point, given the projections of the feature points on the frontal (X_F, Y_F) and profile view (X_P, Y_P).

Having calculated the required positions of the feature points in 3-D, the whole model is stretched, rotated and translated optimally minimising the distance of the feature nodes (the corresponding feature points in the 3-D model) from these required positions. This task is an extension of the *"spatial resection problem"* in photogrammetry [4], which is solved using a modification of the Levenberg - Marquardt algorithm.

In the second step of the adaptation process the model is transformed in a non-rigid way so that the feature nodes are displaced even closer to their exact calculated positions while the facial parts retain their natural characteristics. To perform this adaptation the model is split into face parts (left eye, right eye, mouth, etc.) and the previous rigid adaptation is performed on every part separately. This means that every 3-D face part is rotated, translated and stretched optimally, by minimising the distances of the feature points (belonging to that face part) from their required positions. Thus, the final deformed model adapts to the particular characteristics implied by the feature points (e.g. bigger nose or smaller eyes) keeping the generic characteristics of a human face (e.g. smoothness of the skin and symmetry of the face).

3. 3-D ELLIPSOID APPROXIMATION

In [5], an efficient method was presented for fitting ellipses to scattered 2-D data. The method is based on fitting the general equation of a conic to the given data, subject to a constraint forcing the conic to be an ellipse. In this paper, this technique is generalised in the 3-D case and is used for the approximation of a general set of 3-D points by a 3-D ellipsoid.

The general equation of a 3-D conic is:

$$F(\mathbf{a},\mathbf{x}) = a_1x^2 + a_2xy + a_3y^2 + a_4xz + a_5yz + a_6z^2 + a_7x + a_8y + a_9z + a_{10} = \mathbf{a}^T \cdot \mathbf{x} = 0 \quad (2)$$

where $\mathbf{a} = [a_1\ a_2\ \ldots\ a_{10}]^T, \mathbf{x} = [x^2\ xy\ y^2\ xz\ yz\ z^2\ x\ y\ z\ 1]^T$. Eq. (2) may be written in the form of:

$$F(\mathbf{A},\mathbf{b},c,\mathbf{x}) = \mathbf{x}^T\mathbf{A}\mathbf{x} + \mathbf{b}^T\mathbf{x} + c = 0 \quad (3)$$

where:

$$\mathbf{A} = \begin{bmatrix} a_1 & a_2/2 & a_4/2 \\ a_2/2 & a_3 & a_5/2 \\ a_4/2 & a_5/2 & a_6 \end{bmatrix} = \begin{bmatrix} \mathbf{U} & \mathbf{v} \\ \mathbf{v}^T & a_6 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} a_7 \\ a_8 \\ a_9 \end{bmatrix},$$

$$c = a_{10},\ \mathbf{U} = \begin{bmatrix} a_1 & a_2/2 \\ a_2/2 & a_3 \end{bmatrix},\ \mathbf{v} = \begin{bmatrix} a_4/2 \\ a_5/2 \end{bmatrix}.$$

The following Lemma, provides simple constraints forcing a general 3-D conic to be an ellipsoid.

Lemma 1 **A** *is either positive definite or negative definite iff*

i. $det(\mathbf{U}) > 0 \Longleftrightarrow 4a_1a_3 - a_2^2 > 0$ *and*

ii. $(a_1 + a_3)det(\mathbf{A}) > 0$

The problem of fitting an ellipsoid into N data points $\mathbf{x}_i$ can be solved by minimising the sum of squares of the "algebraic distances"

$$E = \sum_{i=1}^{N} F^2(\mathbf{a},\mathbf{x}_i) = ||\mathbf{D}\mathbf{a}||^2 \quad (4)$$

subject to the constraint $\mathbf{a}^T\mathbf{C}\mathbf{a} = 1$ where $\mathbf{D} = [\mathbf{x}_1\ \mathbf{x}_2\ \cdots\ \mathbf{x}_N]^T$. Since a_1 is a free parameter, the first condition of Lemma 1 can be converted into an equality, as in [5] for the 2-D case. Thus, the constraint can be written as $\mathbf{a}^T\mathbf{C}\mathbf{a} = 1$, where:

$$\mathbf{C} = \begin{bmatrix} 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (5)$$

The constraint minimization problem posed by equations (4) and (5) can be directly solved using a Lagrange multiplier, which yields:

The solution $\mathbf{a}$ is the generalized eigenvector of

$$\mathbf{D}^T\mathbf{D}\mathbf{a} = \lambda\mathbf{C}\mathbf{a} \quad (6)$$

corresponding to the smallest positive eigenvalue.

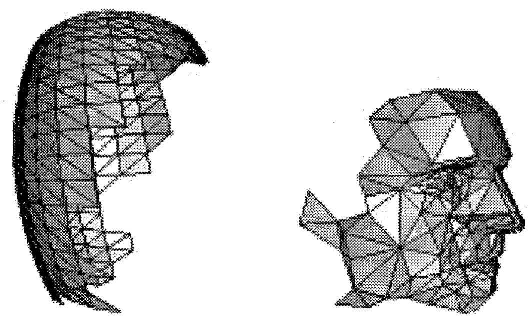
In order to verify that the conic is an ellipsoid, a check has to be made as to whether the second condition of Lemma 1 is satisfied. Note that use of Lemma 1 is computationally very advantageous for the actual determination of the eigenvalues of **A** which are needed to verify if it is positive or negative definite. This is particularly useful when repeated such verification is needed, as is the case when the above method is used alongside the Median of Least Squares method to achieve better robustness to noise.

4. GENERATION OF THE 3-D HEAD MODEL

In order to identify the part of the ellipsoid that corresponds to the back of the head, a check is performed as to whether the line that connects each node with the ellipsoid center intersects with any triangle of the wireframe corresponding to the front(face) wireframe. If so, the corresponding node is removed from the ellipsoid wireframe. Furthermore, each triangle in the ellipsoid wireframe containing this node is also removed.

At this stage, two wireframe models corresponding to the front and the back part of the head have been generated, as seen in Figure 2.

Since these parts are not connected, an algorithm was developed to automatically connect them. This is achieved, by first detecting the list of nodes that define the *largest wireframe boundary chain* in each wireframe model. The *largest wireframe boundary chain* is simply the set (chain) of nodes corresponding to the



(a) (b)

Fig. 2. wireframe models corresponding to a) the back and b) the front part of the head

largest path that can be defined using wireframe edges (edges belonging to only one triangle). Then, an algorithm is used to create additional triangles using the nodes of these two sets, in order to connect the two wireframes in a natural way. The procedure of creating triangles between the two chains A and B can be summarized as follows:

1. Two consecutive nodes are selected from set A, namely nodes A1 and A2.
2. The node from chain B closest to node A1 is identified as node B1.
3. The node that is consecutive to node B1 in chain B and has the smallest distance to node A2 is identified as node B2
4. The following steps are iterated until all chain nodes from chain A or B have been processed: If distance(A1,B2)¡ distance(B1,A2), then
   - the triangle (A1,B1,B2) is added to the mesh
   - Node B2 becomes the new B1 node, and the node in chain B after node B2, becomes the new B2 node.

   If distance(A1,B2)¿ distance(B1,A2), then
   - triangle (A1,A2,B1) is added to the mesh
   - Node A2 becomes the new A1 node, and the node in chain A after node A2, becomes the new A2 node.

The above procedure is schematically illustrated in Figure 3.

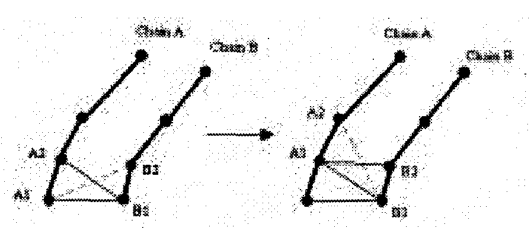


Fig. 3. Procedure for merging two wireframe models.

Finally, in order to be able to visualise a textured head model from any viewpoint, it is necessary to blend the images into a single texture map. As in other approaches [6, 7], we define a mapping between the 3-D coordinates on the face model and the 2-D positions on the texture map using a cylindrical projection. Then, each triangle of the wireframe is projected on the cylindrical surface to define the texture coordinates for each wireframe node. Furthermore, depending on its geometrical position, each triangle of the wireframe may be visible in the frontal view, in the profile view, or in both views. For each position in the cylindrical surface, the corresponding position in the wireframe triangle is found. Then, the texture value T which is stored to the texture map is produced as a weighted average of the corresponding colour values I_F, I_P obtained by projecting the corresponding 3-D point to the frontal and the profile images respectively:

$$T = kI_F + (1-k)I_P. \quad (7)$$

At the above equation, $k = 1$ is used if the 3-D point is projected only in the front view and $k = 0$ is used if the 3-D point is projected only in the profile view. If the 3-D point is projected in both the front and the profile views, k is selected to be proportional to the area of the 2-D triangle projections on the front and profile views of the wireframe 3-D triangle. The reason for this is that the greater this area is, the better the resolution of the corresponding texture image, in the vicinity of this 3-D point.

However, many triangles from the ellipsoid (back part of the head) may not be projected neither in the front nor in the profile view. The area in the texture map corresponding to such triangles is filled with texture from the hair region, which can be semi-automatically identified in the profile view.

5. EXPERIMENTAL RESULTS

In Figures 4(a-b), the adaptation of the initial wireframe to the face feature points is presented, while two views of generated head model are presented in Figures 5(a-b)(front and rear view respectively).

The texture map generated from the front and the profile view is shown in Figure 6, while a view of the texture mapped head model is shown in Figure 7.

6. REFERENCES

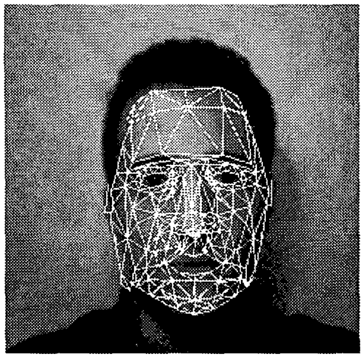
[1] I. Essa et al., "Coding, analysis, interpretation and recognition of facial expressions," *IEEE Trans. Pattern Anal. and Mach. Intell.*, vol. 19, no. 7, pp. 757–763, July 1997.

[2] S. Basu, I. Essa, and A. Pentland, "Motion Regularization for Model-based Head Tracking," in *Proceedings, International Conference on Pattern Recognition*, Vienna, Austria, August 1996.

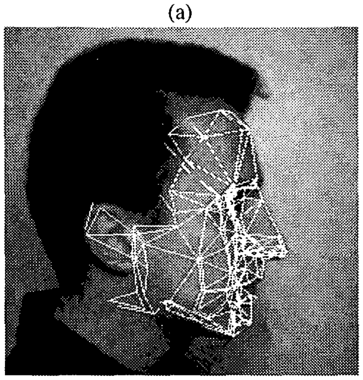
[3] M.J.T. Reinders, P.J.L. van Beek, B. Sankur, and J.C.A. van der Lubbe, "Facial feature localization and adaptation of a generic face model for model based-coding," *Signal Processing : Image Communication*, vol. 7, pp. 57–74, March 1995.

[4] R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision, Volume II*, Addison Wesley, 1993.

[5] A. W. Fitzgibbon, M. Pilu, and R.B. Fisher, "Direct Least Squares Fitting of Ellipses," in *Proc. International Conference on Pattern Recognition*, Vienna, Austria, August 1996.



(a)



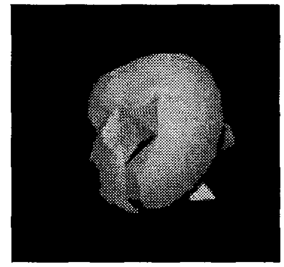
(b)

Fig. 4. The adapted 3-D face model projected to the two available images. a) front view b) profile view.

[6] Frédéric Pighin, Jamie Hecker, Dani Lischinski, Richard Szeliski, and David H. Salesin, "Synthesizing realistic facial expressions from photographs," *Computer Graphics*, vol. 32, no. Annual Conference Series, pp. 75–84, August 1998.

[7] David T. Chen, Andrei State, and David Banks, "Interactive shape metamorphosis," in *1995 Symposium on Interactive 3D Graphics*, Pat Hanrahan and Jim Winget, Eds. ACM SIGGRAPH, April 1995, pp. 43–44, ISBN 0-89791-736-7.

(a)



(b)

Fig. 5. The generated head model, viewed from two different viewpoints a) front view b) rear view.

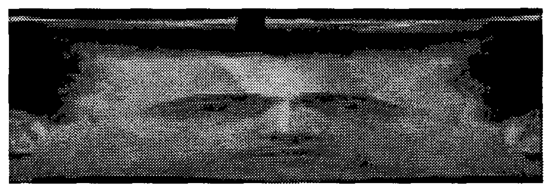


Fig. 6. The texture map generated from the front and the profile view.

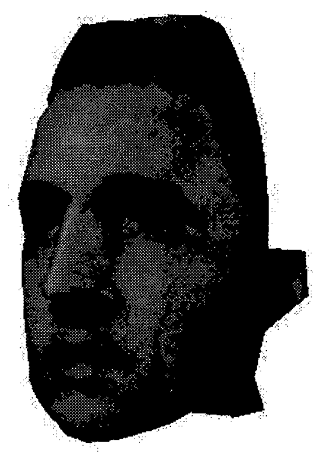


Fig. 7. The textured generated head model