# 3D FACE LASER SCANS NORMALIZATION USING OPENGL Z-BUFFER

*Krystian Ignasiak, Marcin Morgoś, Władysław Skarbek*

Multimedia Laboratory, Institute of Radioelectronics, Warsaw University of Technology

## ABSTRACT

3D data processing operations which were applied to 3D laser scans of human heads are described. The preprocessing starts with spikes removal using median filtering. Then scans are registered exploiting mapping of 3D mesh to the reference 3D mesh based only on four FAP points. Finally 3D data is rescanned using OpenGL Z-buffer. The rescanning process eliminates errors and perturbations bounded to different position of the head while scanning. The rescanned models are used for building 3D eigenfaces both in shape and color domains.

## 1. INTRODUCTION

3D face modeling is widely used in multimedia research community. There are many reasons, e.g.:

— Face animation on personal computers, can be now performed in real time even if the mesh model is very detailed. This is due to a fact that sophisticated graphics cards with built-in accelerators of OpenGL and its extensions, have become affordable because of the high demand for real time 3D interactive games.

— Face recognition using 3D face shape models is now possible as efficient algorithms have emerged within many competing projects. This is due to a current political demand for more reliable and cheap biometric recognition systems.

— Forecasts are made for high demand (in a near future) on distance learning systems based on full immersive virtual classrooms. In such environments video-conference paradigm is replaced by a concept of avatars similar to a real man in terms of visual, aural and interactive behavior. This approach imposes much less bit-rate requirement for communication system supporting e-learning.

On the other hand there are many solutions to acquire 3D models for human heads. One of them is to use laser scanning system, which measures in cylindrical coordinates $(h,\varphi)$ the depth $d$ of facial points and their color components $(r,g,b)$ with intrinsic resolutions $h_{res}$ and $\varphi_{res}$. The height range $\Delta H$ is intrinsic parameter of the scanner as well. The angle range $\Delta\Phi=2\pi$.

Our source data was produced by 3D laser scanner by Cyberwave machinery (http://www.cyberware.com/).

In this paper we describe 3D data processing operations which were applied to 3D laser scans of human heads prior to 3D eigenfaces could be obtained both in shape and color spaces.

## 3. SPIKES REMOVAL

Spikes are large depth values returned by scanner for points of poor reflectance. They rarely occur on facial skin which is of importance for modeling and therefore median filtering in small windows of depth data component, seems to be a good candidate method for their removal.

In previous approach [3] we need to define a reference mesh to obtain the reference 3D face model. Using Z-buffer eliminates importance of reference mesh since it can be chosen arbitrarily. This is described in the next section.

## 4. RESCANNING WITH OPENGL Z-BUFFER

First, the reference face model is chosen. Our experiments show that it is enough to choose one of the original models with its inertia axis closest to the scanner's axis. Chosen reference face is depicted in Figure 1.

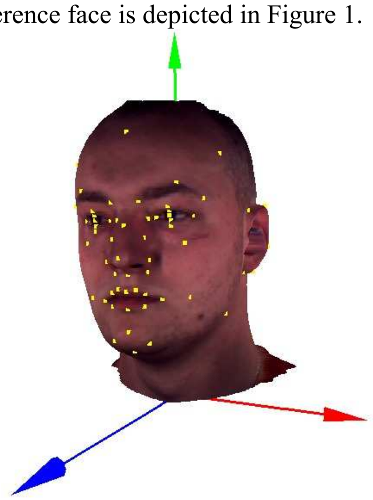

**Figure 1 Chosen reference face with FAP points marked.**

The are four arbitrarily chosen Facial Action Points (FAP) of MPEG-4 [5] which are manually selected. The selection is conducted on 2D texture image as well as on displayed 3D model wherever it is more accurate.

**Figure 2 Screen shots from rescanning process, images taken at 0, 45, 90, 135 and 180 degrees during the rotation of the model.**

The chosen four reference points are: nose tip (9.3), middle lower edge of nose bone (9.12), lower contact point between left/right lobe and face (10.7, 10.8).

These four points constitute the reference mesh instead of a mesh based on all visible FAPs. Such a four-points-based mesh is easier to maintain and does not require selection of all visible FAP point within the model.

The same four point are selected on each face and transformation matrices are calculated. The given matrix transforms the given model so that the four FAP points are mapped exactly into the same points on the reference model, i.e. $\boldsymbol{P} = \mathrm{A}\boldsymbol{Q}$, where $\boldsymbol{P} = [P_1–P_4, P_2–P_4, P_3–P_4]$ and $P_k$, ($k$ = 1,2,3,4) denote points selected on 3D reference model (see Figure 3), and $\boldsymbol{Q}$ represents the points selected on the given model being transformed. The transformation matrix $\mathrm{A} = \boldsymbol{PQ}^{-1}$.

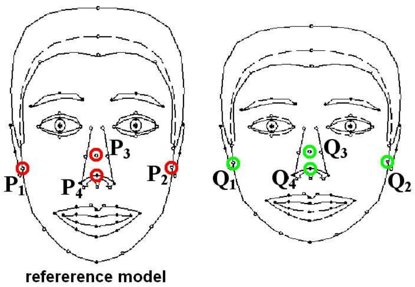


**Figure 3 Four FAP points selected on reference and given 3D model.**

This transformation (translation, rotation and scaling at various rates in all directions) may distort original shape of a given face model, however this process can be reversed applying an inverse transformation matrix. The transformation matrices need to be stored along a normalized 3D face model to perform denormalization after face model is reconstructed with 3D eigenfaces method.

Normalized 3D face model is re-scanned virtually. This process uses OpenGL and 3D acceleration graphic card. This method was chosen instead of manual recalculation of depth-map, because it is simple, accurate and has good performance with current graphic cards [6].

The normalized 3D face model is rendered in OpenGL window using orthogonal projection. No perspective projection is used since it introduces the distortion of the displayed 3D model. It is important to render the model in such a way that it occupies as large part of the window as it is possible with no clipping. The vertical resolution, i.e. the size of an OpenGL window and the size of a rendered model in pixels, influence the vertical resolution of the rescanned model.

At the given time the normalized 3D face model is rendered at a different rotation angle over $y$ axis. The observation point is defined in such a way that the model is displayed exactly in front of it. The rotation angle step results from the desired horizontal resolution of re-scanned model, see Figure 2.

In each step, i.e. after the model has been rendered at the given angle, depth buffer information about middle column of pixels in the OpenGL window is read from the graphic card, i.e. form OpenGL Z-buffer. These values are interpolated by a graphic card prior to rendering based on the information about position and color of the vertices that form the original 3D model.

The values read from the Z-buffer allow for calculation of the distance between observation point and the points situated on the surface of rendered face model. Color information about measured points, i.e. pixels, is also read from the adequate OpenGL buffer.

Once the distances between observation point and points on the surface of the rendered model are known, values for the new depth-map can be calculated as a difference between this distance and the distance between rotation axle and observation point, multiplied by an appropriate scaling factor to maintain original aspect ratio of width/depth and height of the newly 3D face model being created.

## 5. PCA MODELING OF FACE SHAPE AND COLOR

It appears that the collection of facial mesh clouds, obtained for a big group $L$ persons ($L$>50), and

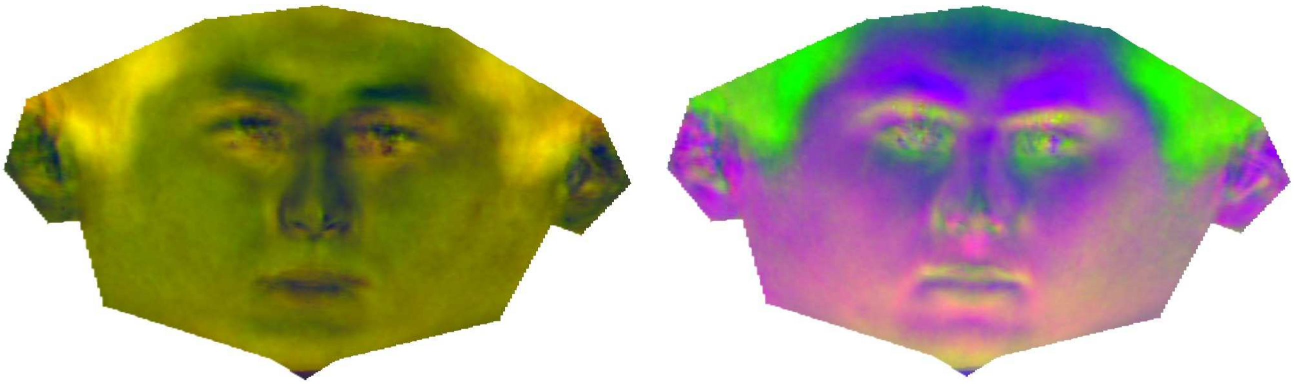

Figure 4 Color component of the first and the second eigenfaces.

considered as a set of points in a high $N$ dimensional space (here $N>4\cdot10^4$) can be approximated by a hyperplane (subspace) of relatively low dimension $M$ (typically M≈50). One of possible linear algebraic bases spanning this subspace is obtained by Principal Component Analysis (PCA [4]) as eigenvectors of covariance matrix for the given training set of vectors.

In case of 2D facial images PCA eigenvectors are called eigenfaces [7] and by the analogy for 3D facial cloud of points, PCA eigenvectors are recognized as 3D eigenfaces [1] (cf. Figure 4). Having $M$ 3D eigenfaces $F_1,\ldots,F_M$ and average face $F_0$ (see Figure 5) any 3D face $F$ can be approximated by a linear combination of 3D eigenfaces where coefficients $\alpha_i$ are appropriate dot products:

$$F \approx F_0 + \sum_{i=1}^{M} \alpha_i F_i, \qquad \alpha_i \stackrel{\Delta}{=} (F - F_0)^t F_i$$

3D eigenfaces are built by Singular Value Approximation (SVA) [2] what allows to avoid building prohibitively large covariance matrices. Each 3D eigenface consists like other original 3D scan of the depth and the color maps.

Figure 5 The mean face visualization.

## 5. CONCLUSIONS

Operations on 3D scan data were described which are necessary to register data from different human head scans. OpenGL Z-buffer approach simplifies the normalization step eliminating tedious and error prone manual selection of all visible FAP points.

The quality of 3D eigenfaces design confirms the validity of presented approach of preprocessing steps applied to laser scans.

**Acknowledgement** The work presented was developed within VISNET, a European Network of Excellence (http://www.visnet-noe.org), funded under the European Commission IST FP6 programme.

## 6. REFERENCES

[1] Blanz V., S. Romdhani, T. Vetter, "Face Identification across Different Poses and Illumination with a 3D Morphable Model," Proc. FG'02, pp.202-207, 2002.

[2] Golub G., C. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, 1996.

[3] Ignasiak K., M. Morgos, W. Skarbek, M. Tomaszewski "3D data processing for 3D face modeling," Proc. *International Conference on Computer Vision and Graphics ICCVG 2004*, Warsaw, Poland, September 22-24, 2004.

[4] Jolliffe I.T., *Principal Component Analysis*, Second Edition, Springer, 2002.

[5] Ostermann J., "Animation of synthetic faces in MPEG-4," *Computer Animation*, pp. 49-51, June 1998.

[6] Shreiner D., M. Woo, J. Neider, T. Davis, *OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 1.4, Fourth Edition,* Addison-Wesley, 2003.

[7] Turk M., A. Pentland A., "Eigenfaces for Recognition," *Journal of Cognitive Neuroscience*, pp. 71-86, 3(1), 1991.