



AUTHOR(S):

TITLE:

YEAR:

Publisher citation:

OpenAIR citation:

Publisher copyright statement:

This is the _____ version of proceedings originally published by _____
and presented at _____
(ISBN _____; eISBN _____; ISSN _____).

OpenAIR takedown statement:

Section 6 of the “Repository policy for OpenAIR @ RGU” (available from <http://www.rgu.ac.uk/staff-and-current-students/library/library-policies/repository-policies>) provides guidance on the criteria under which RGU will consider withdrawing material from OpenAIR. If you believe that this item is subject to any of these criteria, or for any other reason should not be held on OpenAIR, then please contact openair-help@rgu.ac.uk with the details of the item and the nature of your complaint.

This publication is distributed under a CC _____ license.

Automatic Features Characterization from 3D Facial Images

Eyad Elyan
Robert Gordon University/
School of Computing,
Aberdeen, AB251HG, UK
Email: e.elyan@rgu.ac.uk

Hassan Ugail
University of Bradford/
School of Computing,
Bradford, BD71DP, UK
Email: h.ugail@brad.ac.uk

Abstract— this paper presents a novel and computationally fast method for automatic identification of symmetry profile from 3D facial images. The algorithm is based on the concepts of computational geometry which yield fast and accurate results. In order to detect the symmetry profile of a human face, the tip of the nose is identified first. Assuming that the symmetry plane passes through the tip of the nose, the symmetry profile is then extracted. This is undertaken by means of computing the intersection between the symmetry plane and the facial mesh, resulting in a planner curve that accurately represents the symmetry profile. Experimentation using two different 3D face databases was carried out, resulting in fast and accurate results.

Keywords: 3D Images, Features Extraction, Facial Symmetry, Computational Geometry.

I. INTRODUCTION

In many applications, acquiring facial data and fitting a facial surface is usually based on 2D intensity images known as Image based facial modeling. These techniques, involve the utilization of one or more 2D images of a given face to construct 3D facial surface. Therefore, such techniques usually involve two steps. The first is to extract certain features from the facial image[1] and the second step involves the construction of the 3D facial model[2-5].

The use of 3D images acquired by scanning devices has increased significantly in the past few years. This is mainly due to the recent developments leading to the availability of high quality 3D data capture systems. It is noteworthy to point out that 3D face images captured from 3D scanning devices usually tend to be more accurate and is capable of overcoming the problems inherent in geometry constructed from 2D image/s. However, images captured through 3D scanning devices usually require some form of preprocessing stages before it can be utilized for various purposes. For example, in the case of 3D images of human face, it is usually necessary to identify the central region of the face known as the facial mask. Such region, is important for wide range of applications such as 3D face recognition and authentication[3, 6-13], facial expression simulation[14-17] and facial surgery simulation[18]. A common method for determining the facial mask from the raw 3D facial data is based on the surface curvature[19], such as Gaussian curvature. The drawback of such approach is that the

computation of accurate Gaussian curvatures requires sufficiently accurate data which should result in relatively smooth surfaces and such data cannot be always made available through scanning devices.

In this paper we present a novel and computationally fast method for processing 3D facial images and automatic identifications of the symmetry profile of a human face. It is important to note that no assumption about the pose of the face has been made. In addition, no preprocessing steps are required.

In the remaining parts of this paper, we discuss our algorithm in detail. In section 2 we review related literature. In section 3 we outline the main functions of our algorithm and explain some concepts and terminologies that are used in this paper. In the following section we will present experiments and results and will provide analysis to our results in terms of accuracy in recovering facial features and discuss computation time. Finally conclusions and future work will be presented.

II. RELATED WORK

One of the main challenges in processing and determining certain facial features for a given raw 3D facial mesh is due to the resulting scanned image, which usually contains unwanted geometry that need to be identified and discarded at a pre-processing stage as shown in Figure 1. In certain applications semi-automatic approaches have been introduced to overcome this problem[6, 7, 10, 12, 20, 21,22]. For example, BenAbdelkader et al. [20], used seven manually selected land-mark points. Similarly in [21] it is required to identify several landmark points e.g. nose tip and eye corners which can then be used to register the face.

A key component within facial data is the symmetry characteristic that is defined by a symmetry plane which divides the face into two similar halves. Wide range of methods are available in the literature that deals with symmetry detection, in particular for 3D face shapes[6, 11, 23-26].

Sun et. al.[24], for example, assume that the symmetry plane passes through the center of mass of a given object and uses Extended Gaussian Image (EGI) based technique to detect reflection, and rotational symmetry of objects. For facial data such assumption might not hold, especially that 3D facial data acquired by laser scanners might be highly

asymmetric since it would contain noise, and undesired geometry such as neck and the shoulder.

Pan et. al.[23] uses a similar approach to detect the symmetry plane of facial data and reported 95% of good results using two different databases of 3D facial scans [24]. Here, they had to simplify the facial data to 2000 vertices in order to obtain efficient computations. Zhang et al.[6] detected the pose of a raw mesh by means of Principle Component Analysis technique (PCA), and then detected the symmetry plane by determining certain facial features (e.g. nose ridge points). They reported that of 120 facial data utilized in their experiments 117 model were correctly characterized by its symmetry profiles and few feature points along the nose area, with an average processing time of 10 seconds. Wu et. al.[27] used a profile matching approach for face authentication. For symmetry analysis, an initial position of the symmetry plane needs to be interactively identified. Colbry and Stockman[11] identified the symmetry plane of a facial scan by matching that scan with a mirror image of itself using face surface alignment algorithm assuming that pose variation is up to 10 degree in roll and pitch and up to 30 degree in yaw.



Figure 1. Centroid point position for a facial scan almost always lies within the boundary of region of interest of a facial scan regardless of the unwanted parts that might exist in the images.

Other techniques for processing and characterizing facial features from 3D images include: the use of principal axes of inertia of the object[28], point signature techniques [9, 29], and PCA based techniques [13].

III. MAIN METHOD

A. Definitions

3D images are either produced as point clouds or polygonal meshes (usually triangular). A point cloud is simply a set of n vertices $V = \{p_i \mid p_i \in R^3, 1 \leq i \leq n\}$. A triangular mesh S on the other hand, includes the set of vertices and adjacency information and is defined as $S = \{V, E, F\}$, where E is a set of edges defined as $E = \{(p_i, p_j) \mid p_i, p_j \in V\}$ and F is a set of facets defined as $F = \{(p_i, p_j, p_k) \mid p_i, p_j, p_k \in V\}$.

The Euclidean distance between two points v_1, v_2 denoted by $v_1 = (x_1, y_1, z_1)$, and $v_2 = (x_2, y_2, z_2)$ is defined as

$$d(v_1, v_2) = \|v_1 - v_2\| = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

. If we Let $f_i \in F$ be a facet on the surface mesh defined by the triplets $f_i = \{v_0, v_1, v_2\}$ then the circumference of the f_i is defined as $d(f_i) = d(v_0, v_1) + d(v_1, v_2) + d(v_2, v_0)$. Based on this arrangement we could approximate the tolerance value of the surface mesh as,

$$S_i = \frac{1}{C} \sum_{i=1}^{nf} d(f_i) \quad (1)$$

where nf represents the number of facets in the triangular mesh, and $d(f_i)$ is the circumference of the i^{th} facet and C is a constant computed based on an average estimation of the number of common edges between adjacent facets on the surface mesh.

A normalized and registered raw mesh means that all values of the vertices are scaled to be in the range between 0.0 and 1.0. In addition, the facial data is aligned with the Cartesian coordinate system, such that the nose tip is located at the origin and the face is looking towards the positive z -axis.

A plane is defined by a point and its normal vector, hence a plane will be denoted in the form of $\Pi(p_0, \mathbf{n})$ where p_0 is a point on the plane, and \mathbf{n} is its unit normal vector. A reference depth plane is a plane that is used as the reference for measuring the depth of a given surface point on the mesh. The depth of any point denoted by $p_0 = (x_0, y_0, z_0)$ on the surface mesh is measured as the distance between that point and its projection on depth plane Π which is defined as

$$d(p_0, \Pi) = \frac{n_x x_0 + n_y y_0 + n_z z_0}{\sqrt{n_x^2 + n_y^2 + n_z^2}} \quad (2)$$

where the normal vector of the plane is defined as $\mathbf{n} = (n_x, n_y, n_z)$.

A planner curve is defined as a set of points on the 3D space that belongs to the mesh, and intersects a certain plane. The length of a planner curve is defined as $\sum_{i=1}^m d_i(v_i + v_{i+1})$, where $d_i(v_i + v_{i+1})$ is the Euclidean distance between the two positions points $v_i + v_{i+1}$, where v_i, v_{i+1} , are points positions on the planner curve, and v_1, v_m are the first and last point respectively on the curve.

The 3D face is said to be symmetric, if there is a plane, such that the face is invariant under reflection about it. Essentially a symmetry plane will pass through the tip of the nose. Thus, if the tip of the nose and another two position points are identified on the face then one could define the symmetry plane. In fact there are several techniques that address the identification of nose feature points within range data (e.g. [30]).

The centroid position of a facial surface mesh with n vertices is denoted by $c = (c_x, c_y, c_z)$ where

$c_x = \frac{1}{n} \sum_{i=1}^n x_i, c_y = \frac{1}{n} \sum_{i=1}^n y_i, c_z = \frac{1}{n} \sum_{i=1}^n z_i$. For a well characterized facial data set, the centroid point of a mesh usually lies within region of interest which includes the nose, eyes and mouth features. Thus, it is highly unlikely that such point would lie, outside such region, for example near the neck area, or the facial hair. Figure 1 shows various 3D facial scans[31] with irregular outliers where the above assumption about the centroid position is still true.

B. Method outline

The tip of the nose is considered as one of the easiest feature points to recover from a facial image. In addition, we assume that the symmetry plane of the face passes through the tip of the nose. For human faces this is a very reasonable assumption which is widely accepted in research community[6, 13, 32]. Our methodology is focused on determining the symmetry plane based on the determination of the tip of the nose. The basic structure of the proposed algorithm is as follows,

1. The central region of a 3D scan is initially approximated based on the center of mass and few extreme points.
2. The tip of the nose is determined as the point on the facial surface with maximum perpendicular distance from a certain depth plane.
3. Symmetry plane that passes through the pre-determined nose tip is then determined.
4. A planner curve that accurately represents the symmetry profile is then extracted.
5. Few feature points are then automatically determined on the symmetry profile. These feature points include the nose bridge, lower part of the nose.
6. The central region is then extracted, based on approximating the positions of the outer corners of the eyes.

1) Nose tip Identification

The first step in this process is to identify the tip of the nose. This is considered as the easiest point to recover on a facial scan. In order to determine this point, we fit a bilinear blended Coon's surface patch. Coon's patch is simply parametric surface defined by a given four boundary curves. For more information the reader is referred to[33]. The four boundaries of the Coon's patch are determined based on the boundary curves that enclose an approximated central region of the face.

In order to approximate the region of interest we take the centroid and all points that lie within a pre-determined distance from that point. It is important to highlight that the central region identified here is not an accurate representation of central region of the face. Rather it is an approximation which can be used to identify a "minimum"

region of the face which can provide a smooth boundary on which it includes certain facial features and in particular the nose region. Once this region is approximated, its boundary is sorted and organized so that it represents the four boundary curves of a Coon's patch. Finally, a surface patch within the boundary curves is interpolated based on Coon's patch definition (see [33] for more information).

Having the Coon's surface generated as a reference to the facial points on an approximated central region, it becomes straightforward to recover an initial estimation of the nose tip as the one with the maximum depth from the patch. If we let V to denote the set of all vertices within the approximated region of interest of the facial data and let C denote the set of vertices of the Coon's surface patch, then the initial approximation of the nose tip could be formulated as follows,

$$NTIP_{init} = \max \{d(p_i, e_j) : \forall p_i \in V, e_j \in C\} \quad (3)$$

Provided that, $e_j = \min \{d(p_i, e_j) : \forall e_j \in C\}$. Since the Coon's surface is composed of relatively small number of vertices in order to keep computation to minimum, the above formulation only gives an approximation to the nose tip position. To improve our approximation we fit a plane using the points e_j , recovered in Equation (3) and its neighbors e_{j0}, e_{j1} and compute the nose tip position as the point with maximum depth from the constructed plane. Figure 2(b) illustrates this concept. Assuming that the nose tip is denoted by N_{TIP} , the constructed depth plane fitted is defined as Π_{depth} and n is the normal unit vector to the plane, then the tip of nose is formulated as follows,

$$N_{TIP} = \max \{d(v_i, \Pi_{depth}) : \forall v_i \in V\} \quad (4)$$

where $d(v_i, \Pi_{depth})$ is the Euclidean distance between a point v_i on the surface of the face and the constructed depth plane Π_{depth} .

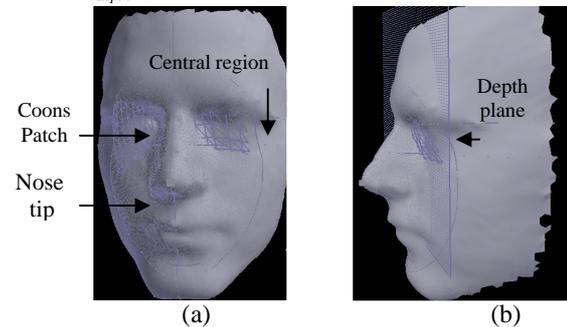


Figure 2. Nose tip identification. (a) Initial estimation of nose tip based on depth measured relative the Coon's patch. (b) Improving accuracy of nose tip positioning based on fitting a plane.

This procedure enables us to neutralize the facial data with the tip of the nose residing at the origin of a right hand

coordinate system. In addition, the facial data can now be transformed in the Cartesian coordinate system with a rotation vector \mathbf{r} defined by two points $N_{TIP}, Nproj_{\mathbf{u}}$ that respectively represents the nose tip and its projection on the depth plane with normal unit vector \mathbf{u} . Thus, once we identify the nose tip correctly, we then rotate the facial data such that \mathbf{r} becomes aligned with the z-axis of the Cartesian coordinate system.

2) Symmetry Plane Detection

To identify the symmetry plane, we assume that N_{TIP} point lies on the symmetry plane. In addition, we let a point p_{s1} be any arbitrary point that lies on the depth plane such that $\mathbf{n}_s = (N_{TIP} - p_{s1}) \times (N_{TIP} - Nproj_{\mathbf{u}})$ where $(N_{TIP} - p_{s1}), (N_{TIP} - Nproj_{\mathbf{u}})$ are two vectors such that $Nproj_{\mathbf{u}}$ is the projection of N_{TIP} into the depth plane and \mathbf{n}_s is the normal unit vector resulting from their cross-product. Figure 3 illustrates this arrangement. Clearly both depth plane and the initial symmetry plane with normal \mathbf{n}_s are perpendicular to each other.

Assuming that the initial symmetry plane defined by the point p_{s1} and its normal unit vector \mathbf{n}_s denoted as $\Pi(p_{s1}, \mathbf{n}_s)$ and recalling that p_{s1} is one of the points lying on the depth plane then we make the following observations,

1. for a human face, the height dimension of the face is greater than its width
2. it is clear that if the upper part of the face was considered, and the initial symmetry plane was rotated around the z-axis, then the planner curve that is identified as the intersection between the facial points and the initial symmetry plane with the minimum length will be the symmetry profile.

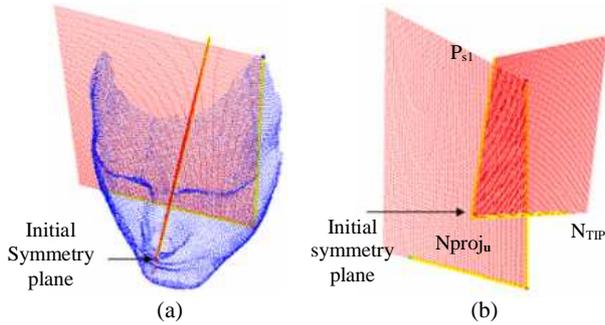


Figure 3 Symmetry plane identification. (a) Facial surface with depth plane, and initial symmetry plane. (b) Initial symmetry plane results from the nose tip, its projection into depth plane, and an arbitrary point on the depth plane.

Based on this arrangement, the initial symmetry plane is rotated by 2π around the z-axis and computation is performed to verify the correct allocation of the symmetry

plane. In order to perform the rotation, we compute an angle θ where θ is the angle by which the symmetry plane should be rotated each time. Recall that, the facial data is already aligned within the Cartesian coordinate system with the nose tip residing at the origin. Therefore, if we assume that the initial symmetry plane is defined by the three points $N_{TIP}, p_{s1}, Nproj_{\mathbf{u}}$ and recalling that p_{s1} is one point on the depth plane then θ could be defined as

$$\theta = \cos^{-1}\left(\frac{d^2}{S_t^2 + d^2}\right), \text{ where } d = d(p_{s1}, NTIP), \text{ and } S_t \text{ is}$$

the tolerance value of the mesh. Defining θ to be dependant on the tolerance value of the mesh makes it more accurate regardless of how meshes vary in terms of its density. In addition, rotation of the initial symmetry plane based on a very small value for θ minimizes the error value of the detected symmetry plane. Based on θ , the number of rotations that need to be performed is then approximated as

$$n = \frac{2\pi}{\theta}, \text{ Working out the degree of rotations and the}$$

number of rotations to validate the symmetry plane, the algorithm proceeds as shown in Algorithm 1:

Algorithm 1: Approximating Symmetry Plane

Let $height = -100.0, length = 100.0$

Let V' be a subset of the facial data that represents an approximated central region of the face.

While Number of rotation $\leq n$

1. Find $v_0, v_1 \in V'$ such that they both intersect initial symmetry plane at both ends of the central region.

2. Let v_{0p}, v_{1p} be the projected points of v_0, v_1 respectively into the depth plane, and construct an initial symmetry plane based on the three points v_{0p}, N_{TIP}, v_{1p}

3. Find the planner curve $p(l)$ that is resulting from the intersection of the Facial points of the central region with the initial symmetry plane, and let P_{length} be the length of its upper part

4. If

$d(v_{0p}, v_{1p}) > height$ and $P_{length} < length$ then set

$$height = d(v_{0p}, v_{1p}),$$

$$length = P_{length},$$

and store v_0, v_1 , as possible candidate for symmetry plane points.

End if

rotate the initial symmetry plane by θ

Repeat step 1.

Figure 4, provides an illustration for symmetry plane detection algorithm.

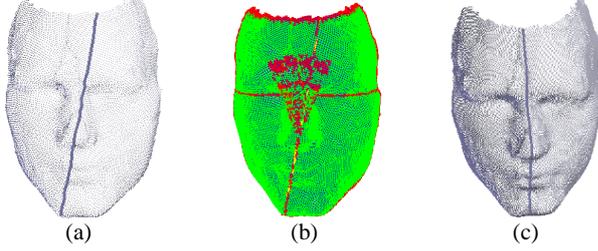


Figure 4 Improving symmetry plane identification. (a) initial symmetry plane, (b) rotating symmetry plane and computing length of the curve. (c) the final symmetry profile.

In order to analyze the symmetry profile extracted from the facial data, we fit a spline of the form $P_i = \sum C_i B_i$ where B_i is a cubic polynomial and C_i are the corresponding control points. This process of curve fitting to the extracted discrete symmetry profile data enables us to have a smooth curve passing through the discrete data. Once we have a smooth symmetry profile, we analyze the profile by identifying local extreme points that corresponds to the nose bridge and the lower point of the nose Figure 5.

Based on the symmetry profile, a profile that passes through the nose bridge and through the eyes area can be extracted. Figure shows the relation between the cross-sectional eyes profile which passes through the point NB (nose bridge), and the symmetry profile.

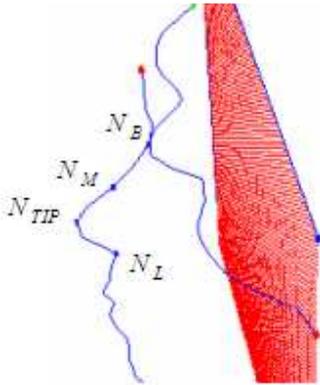


Figure 5 Symmetry profile analysis.

IV. EXPERIMENTS AND RESULTS

Testing the accuracy of the algorithm for detecting symmetry profile was carried out based on reflective symmetry. Figure 6 illustrates this approach, which is similar to the one used in [23] to detect a symmetry plane. As shown in Figure 6, it is assumed that $\mathbf{n} = (n_x, n_y, n_z)$ is the unit normal vector of the detected symmetry plane. If we define a set of points $P = \{p_i\}$ to be the set of vertices that exist at

one side of the symmetry profile, and reflect these points around the symmetry plane, another set of Points Q , will be obtained. Assuming the facial data is perfectly symmetric and the identified symmetry plane is the correct one, then the average mid points of each point and its image $\{p_i, q_i\}$ which is denoted by m_i could be computed using the parametric equation of line $m_i = p_i + \alpha(q_i - p_i)$ where $\alpha = 0.5$ and the average error value is computed as $Err = \frac{1}{n} \sum_1^n d(\Pi, m_i)$, where $d(\Pi, m_i)$ is the distance between the i^{th} mid point and the detected symmetry plane Π in the Euclidean space.

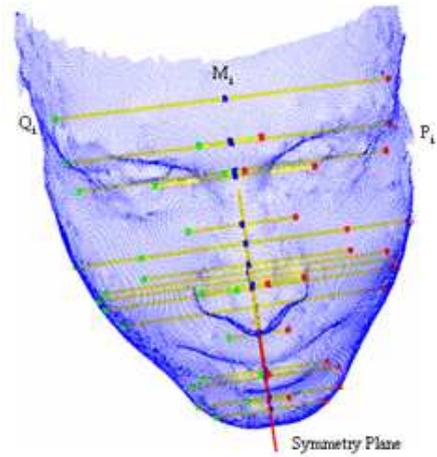


Figure 6 symmetry accuracy by calculating the midpoint between a surface point and its image around the symmetry plane.

In order to test our algorithm, two experiments were carried out on a Pentium 4 machine with 512 MB of RAM and a CPU processor of 2.8GHz. The density of facial models varies from models with 2000 vertices up to 60,000 vertices.

In the first experiment we used a proprietary database with 95 different facial data models with various densities. 80 models were correctly characterized (Figure 7).

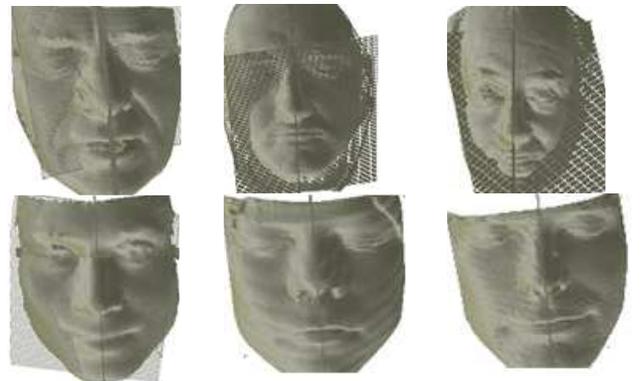


Figure 7 Visualizing correct identification of symmetry profile in different sample images.

Correctness of characterization of the face was based on objective and subjective comparisons. Subjectively a symmetry profile and cross-sectional profiles could be evaluated according to the face model where such profiles must pass through central region of the face (see Figure 7). Based on the testing algorithm using reflective symmetry, it has turned out that detected symmetry plane with error values $>10.0 \times 10^{-3}$ are inaccurate. Figure 8 shows the error rates computed for 10 different facial models with varying mesh densities.

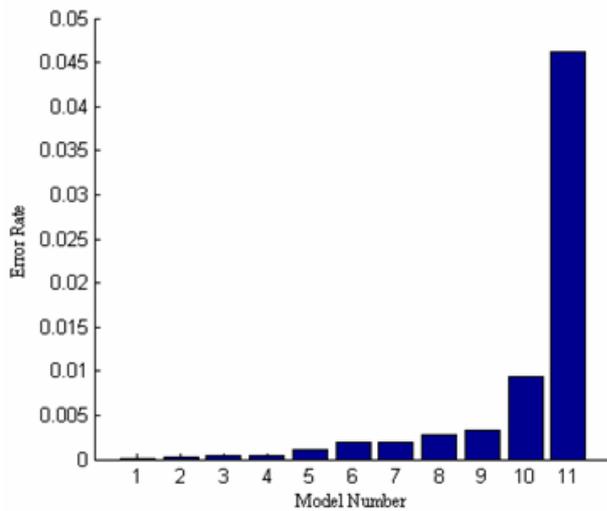


Figure 8 Error rates of the detected symmetry profile for 10 different images with varying mesh densities.

In the second experiment and in order to further test and verify the robustness of our algorithm especially for extreme cases we used 30 different images from GavabDB 3D database [31]. The images used from this database were chosen intentionally to contain irregular outliers such as part of the shoulders and hair. Figure 9 shows sample images of this database that were used in the second experiment. Applying our algorithm yields promising results where all of 30 images were correctly characterized.



Figure 9 Sample images from GavaDB database with correct identification of symmetry profile.

In both experiments the average processing time of facial models to extract nose tip, symmetry profile and cross-sectional profiles was 2.75 second/image. This time doesn't

include loading the image. Large images of more than 50,000 vertices require 3.9 second to be processed.

V. CONCLUSIONS AND FUTURE WORK

In this paper we have described a technique to automatically process raw scanned data using non-iterative and fast computational algorithm.

In addressing this problem we have decomposed the whole process into a set of sub problems that performs as a sequence. The process starts by characterizing the tip of the nose, as one of the most characterizing features on a face and relatively easy to recover. The symmetry plane is then extracted and analyzed to recover other facial feature points.

In this work no simplification algorithms were used to simplify and reduce the data sets of the facial models for efficient computation. Instead, the algorithm is based on reducing the search space by discarding certain areas of the facial data based on some restrictions values. Average processing time for a given face is 2.5 seconds and 3.79 seconds for a facial model of 60,000 vertices makes our approach acceptable in practices where timing is a critical issue. In addition, this relatively short-time processing leaves room for further improvement to our algorithm where noise in data could be addressed and further improve the results.

One of the limitations of our algorithm is the method of approximating the central region. In most of the images that were incorrectly characterized, the problem was mainly due to the incorrect approximation of the central region. As part of our future plan, we will address this limitation. In addition we will further investigate the possibility of combining this technique with a proper detection and comparison algorithm and verify results using a larger scale database (e.g. Face Recognition Grand Challenge Database).

REFERENCES

- [1] T. Goto, W.-S. Lee, and N. Magnenat-Thalmann, "Facial feature extraction for quick 3D face modeling," *Signal Processing: Image Communication*, vol. 17, pp. 243, 2002.
- [2] W. S. Lee and N. Magnenat-Thalmann, "Fast head modeling for animation," *Image and Vision Computing*, vol. 18, pp. 355-364, 2000.
- [3] D. Jiang, Y. Hu, S. Yan, L. Zhang, H. Zhang, and W. Gao, "Efficient 3D reconstruction for face recognition," *Pattern Recognition*, vol. 38, pp. 787-798, 2005.
- [4] H. H. S. Ip and L. Yin, "Constructing a 3D individualized head model from two orthogonal views," *Visual Computer*, vol. 12, pp. 254, 1996.
- [5] T. Akimoto and Y. Suenaga, "3D facial model creation from front and side views of a face," *NTT R&D*, vol. 42, pp. 455, 1993.
- [6] L. Zhang, A. Razdan, G. Farin, J. Femiani, M. Bae, and C. Lockwood, "3D face authentication and recognition based on bilateral symmetry analysis," *Visual Computer*, vol. 22, pp. 43, 2006.
- [7] V. Blanz and T. Vetter, "Face recognition based on fitting a 3D morphable model," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, pp. 1063-1074, 2003.
- [8] T. H. Lin, W. P. Shih, W. C. Chen, and W. Y. Ho, "3D face authentication by mutual coupled 3D and 2D feature extraction," in

- Proceedings of the 44th annual southeast regional conference.* Melbourne, Florida: ACM Press, 2006.
- [9] C. Chin-Seng, H. Feng, and H. Yeong-Khing, "3D Human Face Recognition Using Point Signature," in *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition* IEEE Computer Society, 2000.
- [10] T. Tan, T. Kuhnappel, A. Wongso, and F.-L. Lim, "A solution for illumination challenged face recognition using exemplar-based synthesis technique," Bangkok, Thailand, 2005.
- [11] D. Colbry and G. Stockman, "Canonical Face Depth Map: A Robust 3D Representation for Face Verification," presented at Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on, 2007.
- [12] Y. Xue Dong, W. Xu, and Y. Boting, "3D characteristic facial contours," presented at Electrical and Computer Engineering, 2005. Canadian Conference on, 2005.
- [13] Y. Wang, G. Pan, Z. Wu, and Y. Wang, "Exploring facial expression effects in 3D face recognition using partial ICP," presented at Asian Conference on Computer Vision Hyderabad, India 2006.
- [14] Y. Zhang, E. C. Prakash, and E. Sung, "Face alive," *Journal of Visual Languages & Computing*, vol. 15, pp. 125, 2004.
- [15] T. D. Bui, D. Heylen, and A. Nijholt, "Improvements on a Simple Muscle-Based 3D Face for Realistic Facial Expressions," in *Proceedings of the 16th International Conference on Computer Animation and Social Agents (CASA 2003)*: IEEE Computer Society, 2003.
- [16] J. x. Chai, J. Xiao, and J. Hodgins, "Vision-based control of 3D facial animation," in *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation.* San Diego, California: Eurographics Association, 2003.
- [17] P. Wang, C. Kohler, F. Barrett, R. Gur, R. Gur, and R. Verma, "Quantifying Facial Expression Abnormality in Schizophrenia by Combining 2D and 3D Features," presented at Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on, 2007.
- [18] R. M. Koch, S. H. M. Roth, M. H. Gross, A. P. Zimmermann, and H. F. Sailer, "A Framework for Facial Surgery Simulation," Budmerice, Slovakia, 2002.
- [19] E. Trucco and A. Verri, *Introductory Techniques for 3-D Computer Vision*: Prentice Hall PTR, 1998.
- [20] C. BenAbdelkader and P. A. Griffin, "Comparing and combining depth and texture cues for face recognition," *Image and Vision Computing*, vol. 23, pp. 339-352, 2005.
- [21] T. Nagamine, T. Uemura, and I. Masuda, "3D facial image analysis for human identification," in *International Conference on Pattern Recognition*, 1992, pp. 324-327.
- [22] A. Mian, M. Bennamoun, and R. Owens, "Automatic 3D Face Detection, Normalization and Recognition," presented at 3D Data Processing, Visualization, and Transmission, Third International Symposium on, 2006.
- [23] G. Pan, Y. Wang, Y. Qi, and Z. Wu, "Finding Symmetry Plane of 3D Face Shape," in *Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06) - Volume 03*: IEEE Computer Society, 2006.
- [24] S. Changming and J. Sherrah, "3D symmetry detection using the extended Gaussian image," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 19, pp. 164, 1997.
- [25] K. Hattori, S. Matsumori, and Y. Sato, "Estimating pose of human face based on symmetry plane using range and intensity images," presented at Pattern Recognition, 1998.
- [26] B. Gokberk, M. O. Irfanoglu, and L. Akarun, "3D shape-based face representation and feature extraction for face recognition," *Image and Vision Computing*, vol. 24, pp. 857, 2006.
- [27] Y. Wu, G. Pan, and Z. Wu, "Face Authentication Based on Multiple Profiles Extracted from Range Data," in *Audio- and Video-Based Biometric Person Authentication*, 2003, pp. 515-522.
- [28] A. V. Tuzikov, O. Colliot, and I. Bloch, "Brain symmetry plane computation in MR images using inertia axes and optimization," presented at ICPR, Quebec Canada, 2002.
- [29] I. Mpiperis, S. Malasiotis, and M. G. Strintzis, "3D Face Recognition by Point Signatures and Iso-Contours," presented at 4th IASTED International Conference on Signal Processing, Pattern Recognition, and Applications (SPPRA2007), Innsbruck, Austria, 2007.
- [30] C. Xu, Y. Wang, T. Tan, and L. Quan, "Robust nose detection in 3D facial data using local characteristics," presented at International Conference on Image Processing, ICIP, Singapore, 2004.
- [31] A. B. Moreno and A. Sanchez, "GavabDB: A 3D Face Database," presented at 2nd COST Workshop on Biometrics on the Internet: Fundamentals, Advances and Applications, Vigo (Espana), 2004.
- [32] C. Beumier and M. Acheroy, "Automatic 3D face authentication," *Image and Vision Computing*, vol. 18, pp. 315-321, 2000.
- [33] G. Farin and D. Hansford, "Discrete Coons patches," *Computer Aided Geometric Design*, vol. 16, pp. 691-700, 1999.