Automatic 3D Face Detection, Normalization and Recognition

Ajmal Mian, Mohammed Bennamoun and Robyn Owens
School of Computer Science and Software Engineering
The University of Western Australia
35 Stirling Highway, Crawley, WA 6009, Australia
{ajmal, bennamou, robyn.owens}@csse.uwa.edu.au

Abstract

A fully automatic 3D face recognition algorithm is presented. Several novelties are introduced to make the recognition robust to facial expressions and efficient. These novelties include: (1) Automatic 3D face detection by detecting the nose; (2) Automatic pose correction and normalization of the 3D face as well as its corresponding 2D face using the Hotelling Transform; (3) A Spherical Face Representation and its use as a rejection classifier to quickly reject a large number of candidate faces for efficient recognition; and (4) Robustness to facial expressions by automatically segmenting the face into expression sensitive and insensitive regions. Experiments performed on the FRGC Ver 2.0 dataset (9,500 2D/3D faces) show that our algorithm outperforms existing 3D recognition algorithms. We achieved verification rates of 99.47% and 94.09% at 0.001 FAR and identification rates of 98.03% and 89.25% for probes with neutral and non-neutral expression respectively.

1. Introduction

Face recognition is a challenging problem because of the ethnic diversity of faces and variations caused by expressions, gender, pose, illumination and makeup. Appearance based (2D) face recognition algorithms were the first to be investigated due to the wide spread availability of cameras. One of the classic face recognition algorithms uses the eigenface representation of Turk and Pentland [15] which is based on the Principal Component Analysis (PCA). Linear Discriminate Analysis (LDA) [16], Independent Component Analysis (ICA) [3], Bayesian methods [11] and Support Vector Machines (SVM) [14] have also been successfully used for appearance based face recognition. Zhao et al. [17] give a detailed survey of 2D face recognition algorithms and conclude that existing algorithms are sensitive to illumination and pose. Therefore, researchers are now investigating other data acquisition modalities of the face to overcome these limitations. One of the most promising modalities is the 3D shape of the face. A 3D face is a three dimensional vector $[x_i, y_i, z_i]^\top$ of the x, y and z coordinates of the pointcloud of a face ($i = 1 \dots n$, where n is the number of points). A 2D face on the other hand is a five dimensional vector $[u_i, v_i, R_i, G_i, B_i]^\top$ where u, v are the pixel coordinates and R, G and B are their corresponding red, green and blue components. When the 3D face and the 2D face are registered, the pixel coordinates u, v of the 2D face can be replaced with the absolute coordinates x, y of its corresponding 3D face.

Bowyer et al. [6] present a comparative survey of 3D face recognition algorithms and conclude that 3D face recognition has the potential to overcome the limitations of its 2D counterpart. Especially, the 3D shape of a face can be used to correct the pose of its corresponding 2D facial image which is one of the major contributions of our paper. We present a fully automatic algorithm for pose correction of a 3D face and its corresponding 2D colored image. Existing techniques perform pose correction by manually identifying landmarks on the faces (e.g. [7]). Our approach is to automatically detect the nose tip and correct the pose using the Hotelling transform [8]. The pose correction measured from the 3D face is also used to correct the 3D pose of its corresponding 2D face. Since 2D face recognition is a well studied area [17], we will only demonstrate the pose correction of the 2D faces (along with the 3D faces) and then focus on 3D face recognition alone.

Another major contribution of our paper is an efficient 3D Spherical Face Representation (SFR) based rejection classifier which quickly eliminates a large number of ineligible candidate faces from the gallery. The remaining faces are then verified using a novel recognition algorithm which is robust to facial expressions. Robustness to facial expressions is achieved by automatically segmenting the face into expression sensitive and insensitive regions and using the latter for recognition.

2. Three-Dimensional Face Normalization

Blanz et al. [5] used morphable models to deal with pose variations in 2D facial images. We use 3D face data

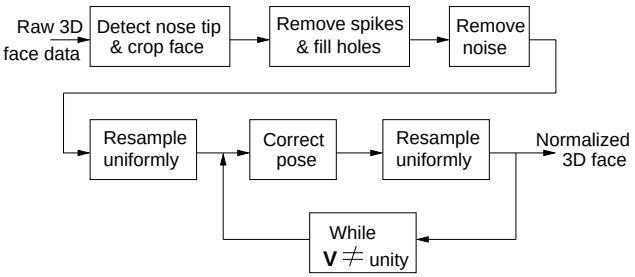

Figure 1. Face normalization block diagram.

acquired with a range scanner to automatically correct the facial pose. A block diagram of our algorithm is given in Fig. 1) and its different components are explained below.

2.1. Face Detection and Denoising

We performed our experiments on the Face Recognition Grand Challenge (FRGC) version 2.0 [13] dataset which comprises 9,500 2D and 3D frontal views of faces mostly acquired from the shoulder level up. Therefore, an important preprocessing step was to detect the face. Moreover, the 3D faces are noisy and contain spikes (see Fig. 3). Since processing 3D data is computationally expensive, we detect the nose tip in the first step in order to crop out the required facial area from the 3D face for further processing. The nose tip is detected using a coarse to fine approach as follows. Each 3D face is horizontally sliced at multiple steps d_v. Initially a large value is selected for d_v to improve speed and once the nose is coarsely located the search is repeated in the neighboring region with a smaller value of d_v. The data points of each slice are interpolated at uniform intervals to fill in any holes. Next, circles centered at multiple horizontal intervals d_h on the slice are used to select a segment from the slice and a triangle is inscribed using the center of the circle and the points of intersection of the slice with the circle as shown in Fig. 2. Once again a coarse to fine approach is used for selecting the value of d_h for performance reasons. The point which has the maximum altitude triangle associated with it is considered to be a potential nose tip on the slice and is assigned a confidence value equal to the altitude. This process is repeated for all slices resulting in one candidate point per slice along with its confidence value. These candidate points form the nose ridge. Points that do not correspond to the nose ridge are outliers and are removed using RANSAC. Out of the remaining points, the one which has the maximum confidence is taken as the nose tip and the above process is repeated at smaller values of d_v and d_h in the neighboring region of the nose tip for a more accurate localization.

A sphere of radius r centered at the nose tip (see Fig. 2) is then used to crop the 3D face and its corresponding registered 2D face. A constant value of $r = 80\ mm$ was selected in our experiments. This process crops an elliptical region (when viewed in the xy plane) from the face with vertical major axis and horizontal minor axis. The ellipse varies with the curvature of the face. For example, the more narrow a face is, the greater is the major axis to minor axis ratio. Once the face is cropped, outlier points causing spikes (see Fig. 3) in the 3D face are removed. We defined outlier points as the ones whose distance is greater than a threshold d_t from any one of its 8-connected neighbors. d_t is automatically calculated using $d_t = \mu + 0.6\sigma$ (where μ is the mean distance between neighboring points and σ is its standard deviation). After removing spikes the 3D face and its corresponding 2D face are resampled on a uniform square grid at 1 mm resolution. Removal of spikes may result in holes in the 3D face which are filled using cubic interpolation. Resampling the 2D face on a similar grid as the 3D face ensures a one-to-one correspondence is maintained between the two. Since noise in 3D data generally occurs along the viewing direction (z-axis) of the sensor, the z-component of the 3D face (range image) is denoised using median filtering (see Fig. 3).

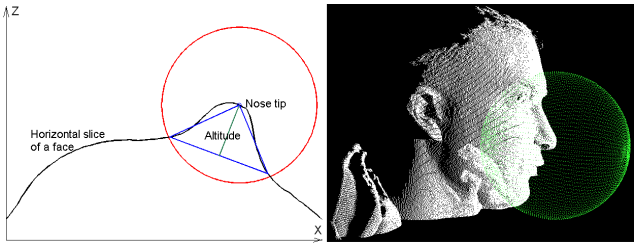

Figure 2. Left: Nose tip detection. Right: A sphere centered at the nose tip of a 3D face is used to crop the face.

Figure 3. Left: A pointcloud of a face with spikes. Center: Shaded view of the same face after removing the spikes shows that it is noisy. Right: Shaded view of the face after complete preprocessing (i.e. cropping, hole filling, denoising and resampling).

2.2. Pose Correction and Resampling

Once the face is cropped and denoised, its pose is corrected using the Hotelling transform [8]. Let $\mathbf{P}$ be a $3 \times n$

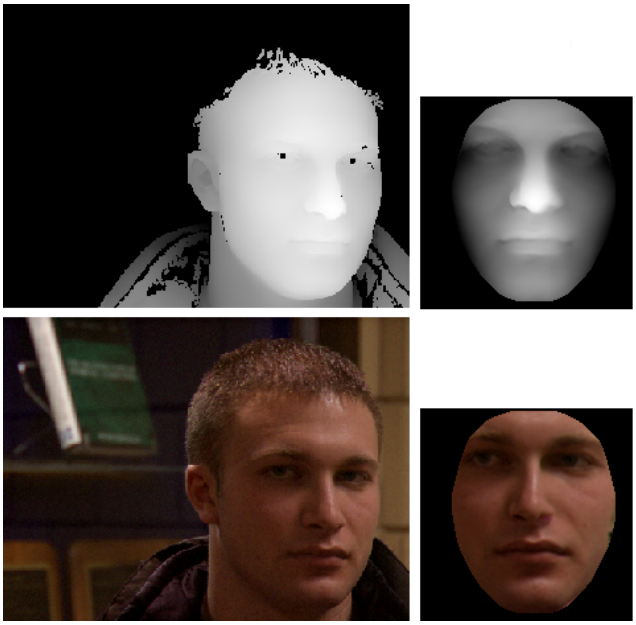

Figure 4. A 3D face and its corresponding 2D face (colored) before and after pose correction and normalization.

matrix of the x, y and z coordinates of the pointcloud of a face (Eqn. 1).

$$\mathbf{P} = \begin{bmatrix} x_1 \, x_2 \, \ldots \, x_n \\ y_1 \, y_2 \, \ldots \, y_n \\ z_1 \, z_2 \, \ldots \, z_n \end{bmatrix} \quad (1)$$

The mean vector and covariance matrix of $\mathbf{P}$ are given by Eqn. 2 and Eqn. 3 respectively.

$$\mathbf{m} = \frac{1}{n} \sum_{k=1}^{n} P_k \quad (2)$$

$$\mathbf{C} = \frac{1}{n} \sum_{k=1}^{n} P_k P_k^T - \mathbf{m}\mathbf{m}^T \quad (3)$$

Where P_k is the kth column of $\mathbf{P}$. The matrix of eigenvectors $\mathbf{V}$ of the covariance matrix $\mathbf{C}$ are given by Eqn. 4.

$$\mathbf{CV} = \mathbf{DV} \quad (4)$$

Where $\mathbf{D}$ is the matrix of the eigenvalues of $\mathbf{C}$. $\mathbf{P}$ can be aligned with its principal axes using Eqn. 5 known as the Hotelling transform [8].

$$\mathbf{P}' = \mathbf{V}(\mathbf{P} - \mathbf{m}) \quad (5)$$

Pose correction may expose some regions of the face (especially around the nose) which are not visible to the 3D scanner. These regions have holes which are interpolated using cubic interpolation. The face is resampled once again on a uniform square grid at 1 mm resolution and the above process of pose correction and resampling is repeated until $\mathbf{V}$ converges to an identity matrix (see Fig. 1).

Figure 5. Sample 3D faces and their corresponding 2D (colored) faces after pose correction and normalization.

$\mathbf{V}$ is also used to correct the *3D pose* of the 2D face corresponding to the 3D face. The R, G and B pixels are mapped onto the pointcloud of the 3D face and rotated using $\mathbf{V}$. This may also result in missing pixels which are interpolated using cubic interpolation. To maintain a one-to-one correspondence with the 3D face as well as for scale normalization, the 2D colored image of the face is also resampled in exactly the same manner as the 3D face. It is important to note that this scale normalization of the 2D face is different from the one found in existing literature. Previous methods (e.g. [7]) are based on manually identifying two points on the face (generally the corners of the eyes) and normalizing their distance to a prespecified number of pixels. As a result, the distance (measured in pixels) between the eyes of all individuals ends up the same irrespective of the absolute distance. This brings the faces closer in the feature space hence making classification more challenging. On the other hand, with our 3D based normalization algorithm, the distance between the eyes of each individual may be different as it is a function of their absolute distance. Thus the faces remain comparatively far in the feature space which results in a more accurate classification.

2.3. Pose Correction Results

Fig. 4 shows some sample 3D and their corresponding 2D faces from the FRGC v2.0 dataset after pose correction. A qualitative analysis of these results show that our algorithm is robust to facial expressions and hair that covers the face. For quantitative analysis, the pose of each face must be compared with some ground truth. Since ground truth was not available, we pairwise registered the 3D faces belonging to the same identities with each other (all possible combinations C_2^n where n is the number of 3D faces belonging

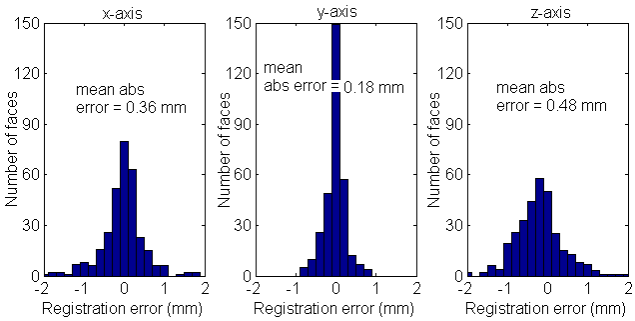

Figure 6. Translation errors between the 3D faces of the same identities after automatic pose correction.

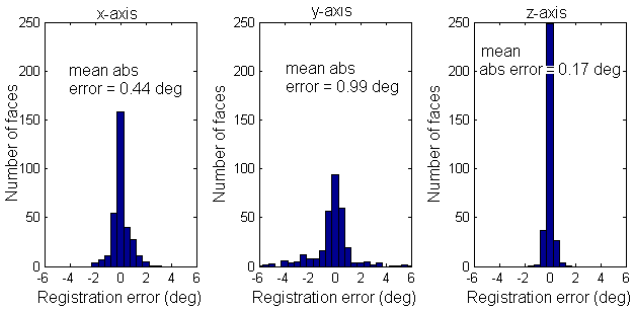

Figure 7. Rotation errors between the 3D faces of the same identities after automatic pose correction.

to the same identity) using the Iterative Closest Point (ICP) [4] algorithm. The translation and rotation errors between these faces are presented in Fig. 6 and Fig. 7 respectively. The maximum absolute mean translation and rotation errors between the faces were $0.48\ mm$ and 0.99^o respectively.

3. SFR: A Low Cost Rejection Classifier

A rejection classifier is defined as the one which quickly eliminates a large percentage of the candidate classes with high probability [2]. A rejection classifier is "an algorithm ψ that given an input, $x \epsilon S$, returns a set of class labels, $\psi(x)$, such that $x \epsilon W_i \Rightarrow i \epsilon \psi(x)$" [2]. Where x is a measurement vector, $S = \Re^d$ is a classification space of d measurements and W_i is the ith class such that $W_i \subseteq S$. The effectiveness $\mathrm{Eff}(\psi)$ of a rejection classifier is the expected cardinality of the rejector output $E_{x \epsilon S}(\mid \psi(x) \mid)$ divided by the total number of classes M (Eqn. 6) [2].

$$\mathrm{Eff}(\psi) = \frac{E_{x \epsilon S}(\mid \psi(x) \mid)}{M} \tag{6}$$

In our case, M is the size of the gallery. The smaller the value of $\mathrm{Eff}(\psi)$, the better is the rejection classifier. The use of a rejection classifier was unavoidable in our experiments

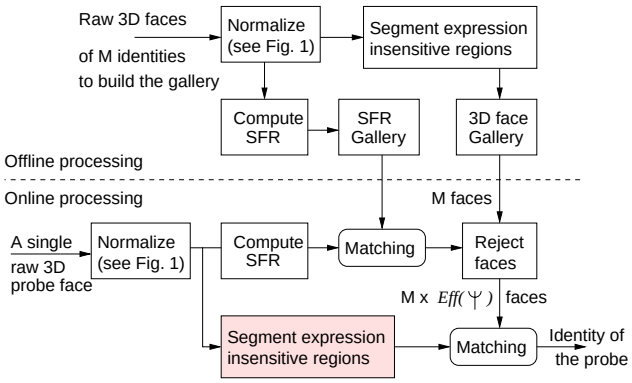

Figure 8. Block diagram of our recognition algorithm. The probe is segmented (shaded block) only in the case of uniform face segmentation.

as the data set was enormous. There were 4393 probes and 557 faces in the gallery. A brute force matching approach would have required 557 x 4393 = 2446901 comparisons. A rejection classifier of $\mathrm{Eff}(\psi) = 0.25$ would reduce this to only 611725 comparisons.

We present a rejection classifier based on a novel Spherical Face Representation (SFR) and intuitively compare it to the spin image representation [9]. Fig. 8 shows the block diagram of our complete 3D face recognition algorithm including the rejection classifier. Intuitively, an SFR can be imagined as the quantization of the pointcloud of a face into spherical bins centered at the nose tip. Fig. 9-a graphically illustrates an SFR of three bins. A spin image is generated by spinning an image (e.g. of size 6x6 in Fig. 9-b) around the normal of a point (the nose tip in our case) and summing the face points as they pass through the bins of the image. Intuitively, the spin image representation appears to be more descriptive compared to the SFR. The recognition performance of a representation is directly related to its descriptiveness. However, the more descriptive the representation, the more sensitive it becomes to facial expressions. Moreover, in terms of computational complexity the spin image representation is more complex than the SFR. Therefore, for use as a rejector, the SFR should be a better choice.

We quantitatively compared the performance of the SFR (15 bins) to the spin images (size 15×15) [9] when used as rejection classifiers. The SFRs were matched using Euclidean distance whereas the spin images were matched using a linear correlation coefficient as described in [9]. Fig. 10 shows the ROC curves of the SFR and the spin images. For probes with neutral expression, the spin images perform slightly better whereas for probes with non-neutral expression the SFR performs slightly better. This supports our argument that representations with higher descriptiveness

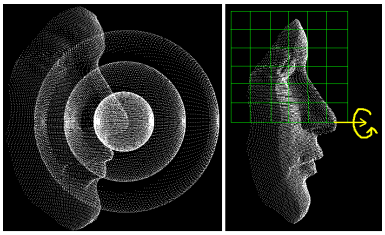

Figure 9. Illustration of the (a) SFR and (b) spin image representation.

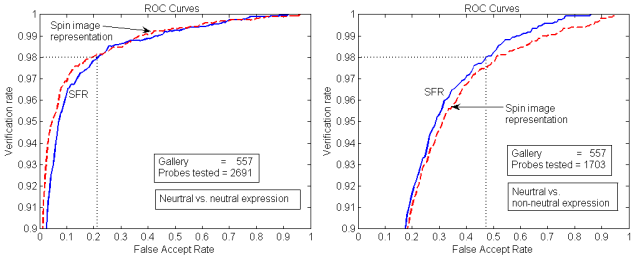

Figure 10. Rejection classification results.

are more sensitive to facial expressions. However, the SFR based classifier is computationally much more efficient than the spin image classifier. Using a Matlab implementation on a 2.3 GHz Pentium IV machine, takes $6.8\ msec$ to construct a SFR of a probe, match it with the 557 SFRs in the gallery and reject a subset of the gallery, whereas the spin images take $2,863\ msec$ for the same purpose. At 98% verification rate, the effectiveness of the SFR based rejection classifier as per Eqn. 6 is 0.21 and 0.48 for probes with neutral and non-neutral expressions respectively.

4. Face Segmentation and Recognition

Fig. 8 shows the block diagram of our complete 3D face recognition algorithm including the rejection classifier and the final recognition process. During offline processing the gallery is constructed from raw 3D faces. A single 3D face per individual is used. Each input 3D face is normalized as described in Section 2 and its SFR is computed. Although 3D face recognition has the potential to achieve higher recognition rates, it is more sensitive to facial expressions compared to 2D face recognition [12]. To overcome this limitation, we segment the 3D faces into expression sensitive and insensitive regions. Two different approaches are used for this purpose. Both approaches are fully automatic, however the first approach segments faces non-uniformly based on the properties of individual faces whereas the second approach performs a uniform segmentation i.e. the same features are segmented from all faces.

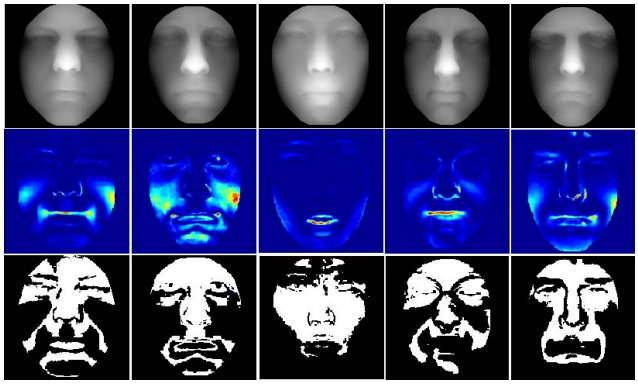

Figure 11. Top: Sample 3D faces. Center: Variance of the 3D faces with expressions. Bottom: The expression insensitive mask of the 3D faces.

4.1. Non-uniform Face Segmentation

The probes with non-neutral expressions were divided into training and test sets. The training set was used during offline processing to automatically determine the regions of the face which are the least affected by expressions. A maximum of three training faces per gallery face were used. The variance of all training faces (with non-neutral expression) from their corresponding gallery faces (with neutral expression) was measured. The regions of the gallery faces whose variance was less than a threshold were then segmented for use in the recognition process. The threshold was dynamically selected in each case as the median variance of the face pixels. Fig. 11 shows some sample 3D faces (first row), their variance due to facial expressions (second row) and the derived mask (third row) for expression insensitive regions. In Fig. 11 second row, bright pixels correspond to greater facial expressions. It is noticeable that generally the forehead, the region around the eyes and the nose are the least affected by expressions (in 3D) whereas the cheeks and the mouth are the most affected.

During online recognition, a probe from the test set is first preprocessed as described in Section 2. Next, its SFR is computed and matched with those of the gallery to reject unlikely faces. The matching process results in a vector of similarity scores of size M (where M is the size of the gallery). The scores are normalized to a scale of 0 to 1 (0 being the best similarity) using Eqn. 7.

$$\mathbf{s} = \frac{\mathbf{s} - min(\mathbf{s})}{max(\mathbf{s} - min(\mathbf{s})) - min(\mathbf{s} - min(\mathbf{s}))} \qquad (7)$$

Gallery faces whose similarity is above a threshold are rejected. Selecting a threshold is a trade off between accuracy and efficiency (or $\mathrm{Eff}(\psi)$). In our experiments we used

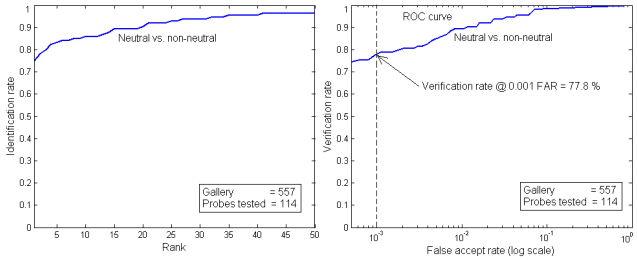

Figure 12. Identification and verification results using the SFR rejection classifier and a non-uniform expression mask (Fig. 11).

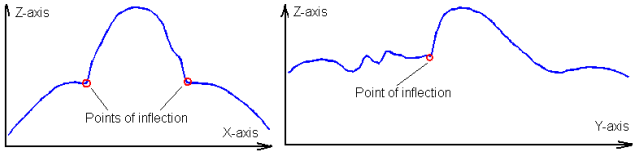

Figure 13. Inflection points detected in a horizontal (left) and a vertical (right) slice of a face.

a threshold so that $\text{Eff}(\psi) = 0.33$. The expected verification rate of the rejection classifier at this threshold was 99% and 98% for probes with neutral and non-neutral expression respectively.

The remaining gallery faces are then matched with the probe using a variant of the Iterative Closest Point (ICP) [4] algorithm (see Section 4.3 for details). Only the expression insensitive regions of the gallery faces are used for matching to avoid the effects of expressions. The probe however is not required to be segmented. The mean registration error between the probe and a gallery face is used as their similarity score (a lower score means a better match). The similarity scores are normalized to a scale of 0 to 1 using Eqn. 7. Fig. 12 shows our identification and verification results using this approach. We achieved a rank one recognition rate of 76.5% and a verification rate of 77.8% at 0.001 FAR (the FRGC benchmark) for probes with non-neutral expression. One possible reason why these results are not impressive is that many of the faces in the gallery did not have enough training images with facial expressions. In fact some of them had no training images, in which case the gallery face was not segmented at all. Another reason is that the training images did not adequately represent all facial expressions. Careful analysis of the probes that were incorrectly identified or verified revealed that their corresponding gallery faces were segmented using fewer than three training images. We believe that this approach will produce good results provided that sufficient training images representing all possible facial expressions are available for all the gallery faces.

4.2. Uniform Face Segmentation

This approach is an extension of our earlier work [10] and does not require any training images since features are consistently segmented from all faces including the gallery and the probes. Given that the nose, the forehead and the region around the eyes are the least sensitive to facial expressions in 3D faces (see Fig. 11), we segmented these features and used them during the recognition process. The features were automatically segmented by detecting the inflection points (see Fig. 13) around the nose tip. These inflection points are used to define a mask which segments the nose, eyes and forehead region from the face as shown in Fig. 14. Note that in this case no training images with facial expressions are required and therefore the probe is also segmented before matching.

The gallery faces were segmented during offline processing. During online recognition, a part of the gallery is rejected using the SFR based rejection classifier as discussed in Section 4.1. Next, the expression insensitive regions of the probe are segmented and matched with those of the gallery faces using our modified ICP [4] algorithm. Fig. 15 and Fig. 16 show our identification and verification results. We achieved a verification rate of 97.44% and 91.65% at 0.001 FAR for probes with neutral and non-neutral expression respectively. Moreover, the identification rates for the same were 97.5% and 88.7% respectively. Note that our verification rate of 91.65% at 0.001 FAR for faces with non-neutral expressions is far better than the best verification rate of 80% reported by FRGC for the same dataset [12].

Fig. 16 shows our identification and verification results without using a rejection classifier. In this case the verification rates at 0.001 FAR are 99.47% and 94.09% whereas the identification rates are 98.03% and 89.25% for probes with neutral and non-neutral expression respectively. This amounts to an average improvement of 2.24% in verification and 0.54% in identification rates. On the down side, the recognition time without using the rejection classifier is three times greater than when the rejection classifier is used.

4.3. Matching

Matching is performed using a variant of the Iterative Closest Point algorithm [4]. ICP establishes correspondences between the *closest points* of two sets of 3D point-clouds and minimizes the distance error between them by applying a rigid transformation to one of the sets. This process is repeated *iteratively* until the distance error reaches a minimum saturation value. It also requires a prior coarse

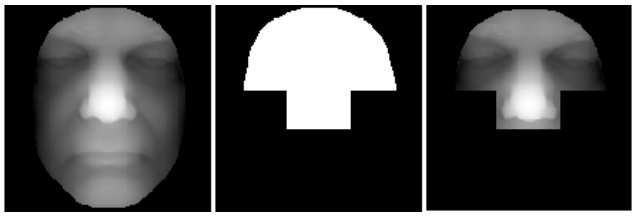

Figure 14. Left: A 3D face. Center: A mask derived from the inflection points around the nose tip. Right: The 3D face after masking.

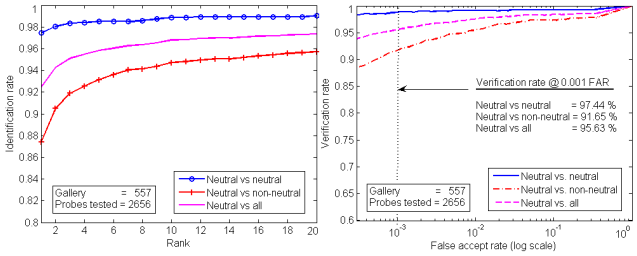

Figure 15. Identification and verification results when using the SFR rejection classifier and a uniform expression mask (Fig. 14).

registration of the two pointclouds in order to avoid local minima. We use our automatic pose correction algorithm (Section 2.2) for this purpose. Our modified version of the ICP algorithm follows the same routine except that the correspondences are established along the z-axis only. The two pointclouds are mapped onto the xy plane before correspondences are established between them. This way, points which are close in the xy but far in the z-axis are still considered corresponding points. The distance error between such points provide useful information about the dissimilarity between two faces. However, points whose 2D distance in the xy plane is more than the resolution of the faces ($1\ mm$) are not considered as corresponding points. Once the correspondences are established, the pointclouds are mapped back to their 3D coordinates and the 3D distance error between them is minimized. This process is repeated until the error reaches a minimum saturation value.

Let $\mathbf{P} = [x_k, y_k, z_k]^\top$ (where $k = 1 \dots n_P$) and $\mathbf{G} = [x_k, y_k, z_k]^\top$ (where $k = 1 \dots n_G$) be the pointcloud of a probe and a gallery face respectively. The projections of $\mathbf{P}$ and $\mathbf{G}$ on the xy plane are given by $\widehat{\mathbf{P}} = [x_k, y_k]^\top$ and $\widehat{\mathbf{G}} = [x_k, y_k]^\top$ respectively. Let $\digamma$ be a function that finds the nearest point in $\widehat{\mathbf{P}}$ to every point in $\widehat{\mathbf{G}}$.

$$(\mathbf{c}, \mathbf{d}) = \digamma(\widehat{\mathbf{P}}, \widehat{\mathbf{G}}) \tag{8}$$

Where $\mathbf{c}$ and $\mathbf{d}$ are vectors of size n_G each such that c_k

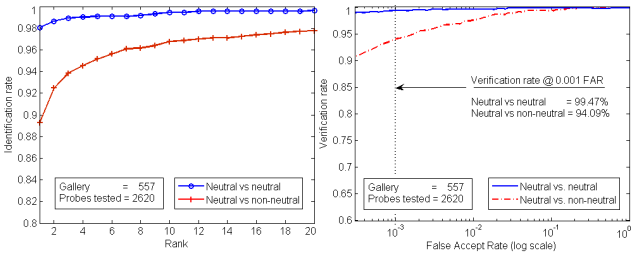

Figure 16. Identification and verification results using a uniform expression mask and without using any rejection classifier.

and d_k contain respectively the index number and distance of the nearest point of $\widehat{\mathbf{P}}$ to the kth point of $\widehat{\mathbf{G}}$. $\forall\, k$ find $g_k \in \mathbf{G}$ and $p_{c_k} \in \mathbf{P} \mid d_k < d_r$ (where d_r is the resolution of the 3D faces equal to $1\ mm$ in our case). The resulting g_i correspond to $p_i\ \forall\ i = 1 \dots N$ (where N is the number of correspondences between $\mathbf{P}$ and $\mathbf{G}$). The distance error e to be minimized is given by Eqn. 9. Note that e is the 3D distance error between the probe and the gallery as opposed to 2D distance. This error e is iteratively minimized and its final value is used as the similarity score between the probe and gallery face.

$$e = \frac{1}{N} \sum_{i=1}^{N} \|\mathbf{R} g_i + \mathbf{t} - p_i\| \tag{9}$$

The rotation matrix $\mathbf{R}$ and the translation vector $\mathbf{t}$ in Eqn. 9 can be calculated using a number of approaches including Quaternions and SVD (Singular Value Decomposition) method [1]. An advantage of the SVD method is that it can easily be generalized to any number of dimensions and is presented here for completeness. The cross correlation matrix $\mathbf{K}$ between p_i and g_i is given by Eqn. 12.

$$\mu_p = \frac{1}{N} \sum_{i=1}^{N} p_i \tag{10}$$

$$\mu_g = \frac{1}{N} \sum_{i=1}^{N} g_i \tag{11}$$

$$\mathbf{K} = \frac{1}{N} \sum_{i=1}^{N} (g_i - \mu_g)(p_i - \mu_p)^\top \tag{12}$$

$$\mathbf{U}\mathbf{A}\mathbf{V}^\top = \mathbf{K} \tag{13}$$

In Eqn. 13, $\mathbf{U}, \mathbf{V}$ are orthogonal and $\mathbf{A}$ is diagonal. The rotation matrix $\mathbf{R}$ and the translation vector $\mathbf{t}$ are given by Eqn. 14 and Eqn. 15 respectively.

$$\mathbf{R} = \mathbf{V}\mathbf{U}^\top \tag{14}$$

$$\mathbf{t} \quad = \quad \mu_p - \mathbf{R}\mu_g \tag{15}$$

$\mathbf{R}$ is a polar projection of $\mathbf{K}$. If $\det(\mathbf{R}) = -1$, this implies a reflection in which case $\mathbf{R}$ is calculated as

$$\mathbf{R} \quad = \quad \mathbf{V} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & det(\mathbf{U}\mathbf{V}^\top) \end{bmatrix} \mathbf{U}^\top \tag{16}$$

5. Limitations and Future Work

Our nose detection and pose correction algorithms assume that the input data contain a front view of a single face with small pose variations ($\pm 15^o$) along the x-axis and the y-axis. However, pose variation along the z-axis can be between $\pm 90^o$. The accuracy of our nose detection algorithm is 98.3% (only 85 failures out of 4950). The failures were mainly due to hair covering a part of the face and in a *few cases* due to exaggerated expressions (e.g. widely open mouth and inflated cheeks). The pose correction algorithm failed to correct the pose of 0.16% of the faces (only 8 out of 4,950) along the z-axis. Hair also caused problems in calculating the SFR and during the final verification process. A skin detection algorithm could be useful to overcome the limitation due to hair. However, applying it before pose correction will result in missing regions from the face (because they were covered by hair) leading to an incorrect pose. In our future work, we intend to use skin detection in our algorithm and fill the missing regions by using morphable models and facial symmetry. Moreover, we will also combine 2D facial features with 3D features for further refining the performances of our algorithms. Finally, we aim to extend our algorithms to be able to automatically determine profile views and perform fully automatic face recognition.

6. Conclusion

We presented a fully automatic 3D face recognition algorithm and introduced several novelties including (1) a 3D nose detection algorithm, (2) an automatic pose correction and normalization algorithm for 3D and their corresponding 2D colored faces, (3) a low cost rejection classifier based on our novel Spherical Face Representation, (4) two different schemes for the automatic segmentation of 3D faces into expression sensitive and insensitive regions and (5) a 3D face recognition algorithm robust to facial expressions. The performance of each algorithm was tested on the FRGC version 2.0 dataset and analyzed. Note that this is the largest available database of its kind. Our 3D face recognition algorithm significantly outperforms existing algorithms [12] particularly for large databases. Our 3D face representation, 3D/2D face normalization and face recognition algorithms make a significant contribution to the face recognition literature and have promising applications.

7. Acknowledgments

Thanks to FRGC organizers [13] for providing the face data. This research is sponsored by ARC grant DP0664228.

References

[1] K. Arun, T. Huang, and S. Blostein. Least-squares Fitting of Two 3-D Point Sets. *TPAMI*, 9(5):698–700, 1987.

[2] S. Baker and S. K. Nayar. Pattern Rejection. In *IEEE CVPR*, pages 544–549, 1996.

[3] M. S. Bartlett, H. M. Lades, and T. Sejnowski. Independent Component Representation for Face Recognition. In *SPIE*, pages 528–539, 1998.

[4] P. Besl and N. McKay. Reconstruction of Real-world Objects via Simultaneous Registration and Robust Combination of Multiple Range Images. *TPAMI*, 14(2):239–256, 1992.

[5] V. Blanz, P. Grother, J. Phillips, and T. Vetter. Face Recognition based on Frontal Views gentrated from Non-Frontal Images. In *CVPR*, pages 454–461, 2005.

[6] K. Bowyer, K. Chang, and P. Flynn. A Survey of Approaches and Challenges in 3D and Multi-Modal 3D+2D Face Recognition. In *CVIU*, volume 101, pages 1–15, 2006.

[7] K. I. Chang, K. W. Bowyer, and P. J. Flynn. Multi-Modal 2D and 3D Biometrics for Face Recognition. In *IEEE AMFG*, pages 187–194, 2003.

[8] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Addison Wesley, 1992.

[9] A. Johnson and M. Hebert. Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes. *TPAMI*, 21(5):674–686, 1999.

[10] A. S. Mian, M. Bennamoun, and R. A. Owens. 2D and 3D Multimodal Hybrid Face Recognition. In *ECCV*, volume 3, pages 344–355, 2006.

[11] B. Moghaddam and A. Pentland. Probabilistic Visual Learning for Object Representation. *TPAMI*, 19:696–710, 1997.

[12] P. J. Phillips. FRGC Third Workshop Presentation. http://www.biometricscatalog.org/documents/Phillips%20FRGC%20%20Feb2020055.pdf, 2005.

[13] P. Phillips, P. Flynn, T. Scruggs, K. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek. Overview of the Face Recognition Grand Challenge. In *CVPR*, 2005.

[14] P. J. Phillips. Support Vector Machines applied to Face Recognition. In *NIPS*, volume 11, pages 803–809, 1998.

[15] M. Turk and A. Pentland. Eigenfaces for Recognition. *J. Cogn. Neurosci*, 3, 1991.

[16] W. Zhao, R. Chellapa, and A. Krishnaswamy. Discriminant Analysis of Principal Components for Face Recognition. In *FG*, pages 336–341, 1998.

[17] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face Recognition: A Literature Survey. *ACM Computing Survey*, pages 399–458, 2003.