International Journal of Computational Intelligence Systems, Vol. 6, No. 4 (July, 2013), 669-683

Robust binary neural networks based 3D Face detection and accurate face registration

Quan Ju [1 2] *,

[1] *College of Computer and Information Engineering, Henan University of Economics and Law, Zhengzhou, China 450002,* †

E-mail: juquan@gmail.com

[2] *Department of Computer Science, University of York, York, UK, YO10 5DD,* ‡

Received 22 March 2011

Accepted 18 February 2013

Abstract

In this paper, we propose a facial feature localization algorithm based on a binary neural network technique - k-Nearest Neighbour Advanced Uncertain Reasoning Architecture(kNN AURA) to encode, train and match the feature patterns to accurate identify the nose tip in 3D. Based on the results of the 3D nose tip localization, the main face area is detected and cropped from the original 3D image. Then we present a novel framework to implement the 3D face registration by several integrated phases. First we use Principal Component Analysis(PCA) to roughly correct the server misalignment. Then we exploit the symmetric of human face to reduce the misalignment about *oy* and *oz* axis. In order to reduce the effect of facial expression variations, the expression-invariant region is segmented. Using Iterative Closest Point(ICP) algorithms, the expression-invariant region of faces can be aligned according to a standard face model, the misalignment about *ox* is then eventually corrected. Our experiments performed on the FRGC v2 database which contains pose and expression variations show that our approach outperforms the current state-of-the-art techniques both in the nose tip localization and face registration.

Keywords: 3D Face, Facial Feature Extraction, Face Registration,Binary Neural Networks, Correlation Matrix Memories, Iterative Closest Point.

1. Introduction

Generally speaking, a 3D face is a group of high dimensional vectors of the *x*, *y* and *z* positions of the vertexes of a face surface. *RGB* color or grayscale information can be added into this vector if the texture values of those vertices is required. A 3D face is usually represented by a 3D shape file and 2D texture image. Face recognition based on 3D has the potential to overcome the challenging problems caused by expression and illumination variations [6]. In order to implement 3D face recognition, firstly we need to know where the main face area is, especially when the 3D face surface includes face, hair, clothing and other noise caused by objects surrounding the face. If the main face area can be found, the face area then can be cropped from the original 3D sur-

*Quan Ju. email:juquan@gmail.com

†College of Computer and Information Engineering, Henan University of Economics and Law,Zhengzhou, China 450002,

‡Department of Computer Science, University of York,York, UK, YO10 5DD,

face to reduce the effect of noise and other non-face factors. A sphere around the nose tip can be defined to crop the face area. Thus a robust and accurate nose tip localization is crucial for an automatic face recognition system. In this paper, we use a binary neural network technique - kNN AURA to precisely localize the nose tip based on a multi-shell 3D shape descriptor. Furthermore, the effect of head orientation variations is a crucial problem of the face recognition. Thereby, an effective face alignment is also required to correct the poses of all faces. Especially those faces belonging to the same individual should be in a consistent pose. To solve the alignment problem, we present an integrated approach to align faces even with expression variations.

1.1. Related work

1.1.1. Approaches of 3D face detection

The nose tip is the most important facial feature and also the center of the face. Many works [28] [10] [29] [21] perform nose tip detection and use the nose tip as the foundation to detect other features or the face itself, as the nose tip is the most prominent feature of the face. Some approaches use an assumption that the nose is the closest point to the camera or device which acquires the 3D data [14] [18]. Although this supposition is true in most cases, there is no 100% guarantee due to the noise. Various pose rotations and the complex situation of hair and clothes could make some places closer than the nose. Making use of the corresponding 2D texture information is a possible way to detect the face area first then localize the nose tip within the selected 3D face crop. That requires 2D texture and 3D shape to correspond perfectly. However, the 2D texture channel is not always precisely matched with the 3D shape channel. Using the 2D face crop method in a face with a poor 2D-3D corresponding probably will obtain the wrong 3D shape crop.

Colombo et al. [8] presented a method to identify the shape of facial features based on 3D geometrical information only by using *HK* Gaussian Curvature classification. They achieved a 96.85% identification rate on a small dataset, although only the rough nose/eye shapes are identified and no accurate locations of the nose tip are detected. Of other algorithms, Bevilacqua et al. implemented an experiment to detect the nose tip based on extending the Hough Transform to 3D point cloud. However, only 18 3D faces are involved in the experiment [5]. Spin image and support vector Machine (SVM) are used to represent and classify 3D shape [9] [30]. In [30], a 99.3% successful localization rate of the nose tip is claimed, but it was tested on a limited dataset without benchmark evaluation. The main problem of those approaches is that they only used a small face database which is not enough to evaluate the performance of the facial feature localization.

Segundo et al. [26] proposed a 3D facial landmark detection based on the analysis of y-projections and x-projections of the topographic depth information. They used a combination of region/edge detection algorithms and a Hough transform based shape detection method to localize the main face area first and then detect facial features. They reported a nose detection rate of 99.95% on Face Recognition Grand Challenge(FRGC) v2 database [24]. However, using methods to detect face area first may result in extra chance of mistakes and they did not report the accuracy of their face detection. Furthermore, it is helpless if the face detection is the purpose of the nose tip localization.

Another problem of above approaches is that their results are not compared with the ground truth data. To the best of our knowledge, most of the methods do not use benchmark datasets to evaluate their results. Romero et al. [25] presented the first work on benchmark datasets based on FRGC database. They manually marked landmarks of eleven facial features. With those marked feature locations, an over 90% nose tip identification is reported. Overall, the nose tip localization is the key solution to the face detection problem. The current approaches either do not achieve satisfied results or do not run the test on enough face data.

1.1.2. approaches of 3D face registration/alignment

A face recognition system should have the ability to handle the influence of pose variations. Therefore, a face registration/alignment step is required. Mian

et al. [21] used a Principle Component Analysis (PCA) based algorithm to correct the pose variations. Three principle components are used as the *x*, *y* and *z*-coordinates of the point cloud of a face. However the noise (for example hair), surface loss and distortion of a face will affect the performance of this method. A widely used solution is ICP-based face alignment. Faltemier et al. [10] proposed a method for curvature and shape index based nose tip detection to localize the position of the nose tip and then align the whole input image to a template using the ICP algorithm. Kakadiaris et al. [16] implemented a multistage alignment method including three algorithmic steps: Spin-images based alignment, ICP-based alignment and Simulated Annealing on Z-Buffers alignment. However, both of these approaches used the whole face area during their alignments. The expression variations could affect the results of alignment by using the whole area of the input images. Other ICP-based approaches [19] [29] attempted to solve the expression problems by only using the less malleable face area such as areas around nose and eyes. Although using the least affected areas is theoretically robust to expression variations, it is based on an assumption that the segmentation of those expression-invariant regions is accurate and 100% correct, which is normally difficult to obtain. Another problem of current face registration approaches is it is difficult to evaluate the experimental results. Most of the current approaches barely mention the evaluation of the face registration results. Some approaches use the further face recognition results to represent the performance of the face registration. However, it is only the between-class performance and is also affected by many other factor involved in the face recognition phase.

1.1.3. Overview of our approach

In the paper, we propose a novel Multi Shell 3D Shape Descriptor to represent the shape of the 3D shape/surface. Using a binary neural network technique - knn AURA to train, store and retrieve the patterns of the shape descriptor of the nose tip, the location of the nose tip can be precisely identified. The main face area is then detected by using the location of the nose tip. As for the face registration, unlike other approaches, we align face along *ox*, *oy* and *oz*-axis separately. Firstly, we roughly correct the poses of all faces by using PCA. Then we minimize the misalignment about *oy* and *oz*-axis by using the symmetry of the face. On the basis of these achievements, the expression-invariant region of a face can be extracted. Finally, ICP algorithm is applied to align the expression-invariant regions of all faces.

This paper is organized as follows. In section 2, The methodology for nose tip identification based on Multi Shell 3D Shape Descriptor and kNN AURA is presented. Section 3 proposes the integrated face registration method. Section 4 shows the experimental results. Section 5 makes the conclusion of this paper.

2. Nose tip localization and face detection

2.1. Multi shell descriptor

3D facial features can be considered as small groups of points and pieces of 3D surface. There are many methods to describe a 3D shape or surface. In 1984, Grimson and Lozano-Perez [12] first discussed how local measurements of 3D position and surface normals recorded by a set of tactile sensors may by used to identify and locate objects. They mentioned that angles relative to the surface normal is an efficient local constraint. Compared with curvature-based shape descriptors, Stein and Medioni [27] proposed a method using a splash structure to describe a surface. At a given location P they compute the surface normal n. Then a circular slice around n with the geodesic radius r is computed. A surface normal n' can be determined at every point on this circle. θ angles between the n and all n' are obtained. By using splashes, a 3D surface can be described. They also stated that the computation of curvature requires a higher order derivative than the tangent. For a curvature based scheme, the signal to noise ratio is lower than for a tangent(or surface normal) based scheme. In 1997, Chua and Jarvis [7] introduced the Point Signature method to describe a 3D shape. They used a sphere to crop a 3D shape at a point P. Then a number of contour points are produced. The surface normal and normal plane also can be calculated at

the point P. Distances d from the contour of points to the normal plane are computed starting from a certain position along a clockwise direction. d and the angle θ of the clockwise rotation together can be used to describe a 3D surface within a sphere. Rather than only use the contour points cropped by a sphere, Xu et al. [30] computed the distances d of all points to the normal plane at the center point P within a sphere. Then the central and second statistical moments - mean and the deviations of these d are computed. A 3D surface patch cropped by a sphere is described using these two moments. Inspired by the above approaches, in this paper, the moments of the local shape characteristics - angles related to the surface normal are used to describe a 3D surface. We provide a novel method to describe the convex or concave degree of 3D local shape within a given sphere but related to a number of shells.

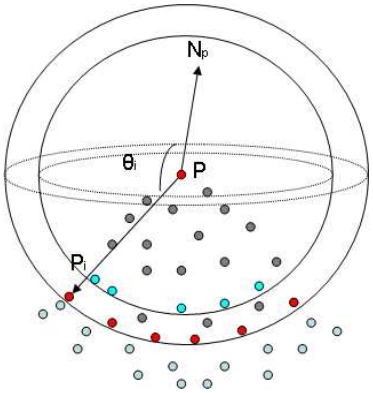

Fig. 1. *P and its neighbouring point within two spheres.*

For a point P in a 3D point cloud, itself and its neighbouring points P_i together forms a 3D surface as shown in figure 1. By finding all the points P_i with the length of edge $P_i - P$ approximately equal to the radius r, point P and those P_i create a $1-$ring mesh. Then the angle between $P_i - P$ and the vertex normal N_p can be calculated by using the following equation:

$$\theta = \arccos\left(\frac{(P_i - P)\cdot N_p}{|P_i - P||N_p|}\right) \qquad (1)$$

where N_p is the vertex normal of point P, θ is the angle between the vertex normal N_p and the edge $P_i - P$, r is the radius of a sphere. θ is between $0^o \sim 180^o$

After the θ of all farthest neighbouring points are calculated, each point P has one of the farthest neighbouring point set $PF(P) = \{P_1, P_2, ..., P_n\}$ and one angle set $\theta(P) = \{\theta_1, \theta_2, ..., \theta_n\}$($n$ is the number of farthest neighbouring point). By calculating the mean θ, we can find out how convex or concave the mesh surface is. For instance, if the mean θ_i of all those farthest neighbouring points is greater than 90^o, this surface within a sphere of radius r can be considered as a convex surface. When the mean of θ_i is less than 90^o,the surface will be a concave one.

However, mean θ above is not enough to describe the subtlety of 3D curvature. Therefore, another statistical attribute: standard deviation of θ is used to provide extra information of the shape. Moreover, the various situation of the cloth and hair sometimes may cause unexpected points to have similar mean value and STD of θ to an expected local facial feature. Similar results can be seen in other works [30] [25] using descriptors representing shapes within a single sphere. Inspired by M. Ankerst's work [2] which first mentioned the multi-shell model, we introduce more shells to calculate mean and deviation of angles. Those two kinds of attributes are used with more shells as shown in figure 2 to create a Multi Shell Surface Angle Moments Descriptor(MSSAMD).

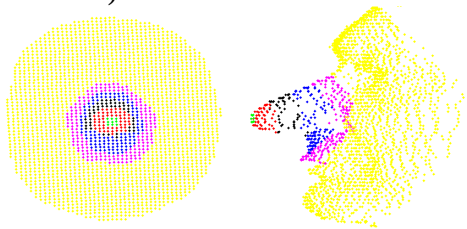

Fig. 2. *The 3D surface is separated by several shells around a point.*

The θ_i represents the angle between N_p and PP_i as shown as in figure 1. Each 'shell' has its standard deviation and mean value of the angles of the points located in its range. Therefore, a 3D surface is described by this MSSAMD including two vectors: $[std_1, std_2, ..., std_n]$ and $[mean_1, mean_2, ..., mean_n]$. The third eigenvector of the covariance matrix as the direction of the normal on point P. Given point $p(x,y,z)$ as the center of a sphere and its neighbouring points $p_i(x_i, y_i, z_i)$ inside the sphere, the covari-

ance matrix of point p is:

$$C = \frac{1}{n}\sum_{i=1}^{n}(p_i - m)(p_i - m)^T \quad (2)$$

$$CV = DV \quad (3)$$

where m is the mean vector of all points, V is the matrix of eigenvectors and D is the matrix of eigenvalues.

2.2. k-Nearest Neighbour AURA Algorithm

To localize the nose tip, we have to create a standard model of a nose tip. The most similar shape within a face to the standard model is the most likely position of the nose tip. A face point-cloud may contains more than thousand of points and a face database usually consists of thousands faces. Thus, a high effective pattern storage and pattern retrieval method is required. In this paper, we use a binary neutral network technique (kNN AURA algorithm) to measure the similarity between the query shape and the standard feature model.

Advanced Uncertain Reasoning Architecture (AURA) is a set of methods based on binary neural networks in the form of correlation matrix memories (CMMs) for high performance pattern matching [3]. Correlation Matrix Memories (CMMs) are a form of static associative memories. Kohonen [17]first introduced the idea of correlation matrix memories in 1972 and made the pioneering contribution together with Anderson [1]. CMMs learn and store the associations between input patterns P and outputs O, which have to be transformed to a binary vector. The input and output patterns are involved in the training of an initially empty binary matrix M. During training, the values within M are only changed to '1' where both input and output vectors are set according to the Hebbian learning introduced in 1949 [13]. The training of M is presented as the following equation.

$$M = \bigvee P^T O \quad (4)$$

Where P is the input pattern (a row vector of binary elements); O: output pattern; M: Correlation Matrix memory; $\bigvee$ is logical OR. Figure 3 shows an example of CMM training process.

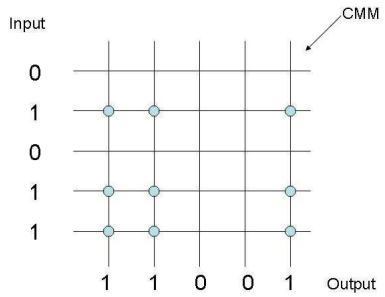

Fig. 3. *Example of training a CMM. When both of the bit of the input and output vectors are '1', a connection of corresponding position in matrix will be set.*

After training, the recall operation returns a summed integer output vector V, then can be thresholded to be a binary vector. If I is the input vector for recall operation, then (following equation):

$$V = MI^T \quad (5)$$

In order to apply AURA technique, input patterns have to be quantized and converted into binary values. The simplest way to transform decimal values into binary values is to divide the possible range of the decimal value of an attribute into several parts called bins, then a binary bit is set to '1' on the basis of which bin the actual value belongs to. CMM necessitates both input and output vectors. The training process stores the binary attributes value into a column of the matrix. Therefore, the output vector is designed as the sequence number of the faces in the training group. As shown as in Figure 4, the training process is to store the nose tips one by one until all the training faces have been saved in the matrix.

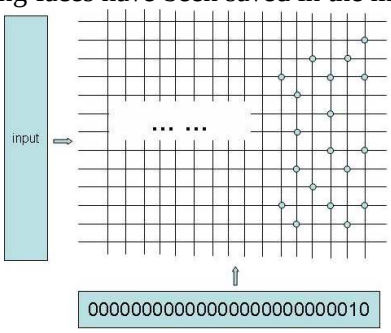

Fig. 4. *Store the each image into a column one by one.*

In the recall or query phase, the query pattern is measured and then feature attributes are generated. However, a difficulty of the quantization method is

the boundary effect. Since there are clear boundaries between bins, a decimal value will only belong to one bin. Thus, the distance between two values within the same bin may be greater than the distance of two values in two neighbouring bins. In this paper, we use a Integer Pyramid technique proposed by Hodge et al. [15] to compensate for that situation. In recall procedure, a weight vector replaces the single bits set in the query vector, each with a 'triangular kernel' of integer values arranged so that the maximum value of a kernel is located where the set bit was, and adjacent zero bits are replaced with smaller integers, decreasing uniformly. This vector of integers then forms the input to the CMM, with the response V calculated in the same way as before. This use of kernels gives a maximum value in V for the stored vector that has been most closely corresponding to the query vector. Vectors that do not match exactly will have a reduced but non-zero response to each query bit. This gives a more gradual decrease in response for non-matching vector than in the original CMM application. Knowing what the maximum response should be, we convert the reduction in response to a vector of 'distances' of the query from the stored vectors. With the triangular kernel described, the distance approximates the quantized City Block Distance. An example of this use of kernels is shown in figure 5.

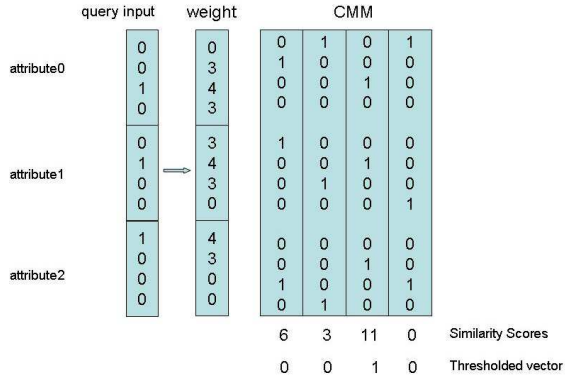

Fig. 5. *An example of CMM recall with kernel weighted inputs.*

The weight vector is later improved using a parabolic kernel [15] to approximate the quantized squared Euclidean distance. For one stored vector, the distance is:

$$d_E^2 = \sum_{\forall f}(x_f - x'_f)^2 \tag{6}$$

where d_E^2 is the squared Euclidean distance, x_f is the query attribute value and x'_f is the stored value for attribute f.

To calculate this distance using a CMM, the parabolic kernel weight values are calculated as in the equation below. For the attribute f and bin k, with the original set bin in bin t:

$$W_{f,k} = (\frac{n^*}{2})^2 - (t-k)^2\alpha_f \tag{7}$$

$$\alpha_f = \frac{n^{*2}}{n_f^2}$$

where n^* is the maximum number of bins for any attribute and n_f is the number of bins for the attribute f. a_f is to ensure the spread of the kernel for all attributes within the CMM input vector. Figure 6 shows the parabolic shape weight values.

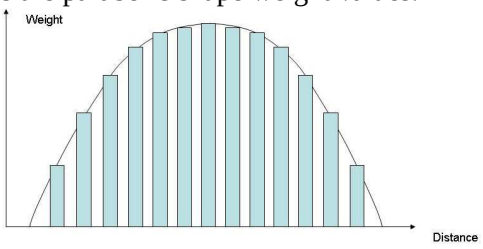

Fig. 6. *The weight values of the CMMs are set to be analogous to parabolic shape which describe the distance from the central bin.*

By using the parabolic kernel technique, the output V contains scores ranked by Euclidean distance. These can be used as a similarity score vector for each query, so $V = \{v_1, v_2, ..., v_p\}$ is the similarity with each of the training nose tips. $max(V)$ tells the level of similarity that a query pattern has to at least one nose tip of the training group.

2.3. *Nose tip localization hierarchical methodology*

In order to create a MSSAMD descriptor suitable for nose tip detection, the maximum radius of the farthest sphere is defined as $25mm$ simply because it is the approximately range from a nose tip to its edges. We simply used $5mm$ to be the width of a shell to make sure there are enough points existing in every shell area. As a result, there are totally five shells. By using the manually marked nose tips in the training dataset, the attributes in MSSAMD of different

shapes of nose tips are converted into binary vectors then stored into the CMM. After the training process, the attributes of the MSSAMD of all points in the target faces are also calculated and encoded with kNN AURA weights.

We define the three following steps to reduce the number of candidate points for the nose tip in a particular image:

Step one: For a point P_i, the attributes of MCSAMD or MSSAMD are matched with the features stored in the AURA. By using a kNN AURA matching algorithm, a similarity score vector V is generated. V contains the similarity scores to all features from different subjects stored in AURA. The highest similarity score $S = max(V)$ is chosen as the final similarity score for this point P_i. Then by simply defining a threshold T_{nose}, any candidate with a similarity score below T_{nose} is deleted from the candidates list. This step can significantly narrow down the range of candidate points.

Step two: There are usually some other points left in the candidate list such as those in the hair, clothes or chin areas that cannot be eliminated in step one. However, most of those exceptional points are scattered and the points around the actual nose tip always get a relatively high similarity score. Therefore, we can locate the correct nose tip cluster by calculating the number of the candidates within a certain range. The cluster with the highest density of candidate points is chosen as the nose tip candidate cluster.

Step three: After the nose tip cluster is selected, the candidate with the highest similarity score inside this cluster is considered as the final choice.

2.4. Face localization

After the nose tip has been identified and localized. As the nose tip is at the center of the face, the main face area can be extracted from the original image. In this paper, 100*mm* is selected as the radius of this sphere to crop face so as to keep as much detail as possible. An example is shown in figure 7.

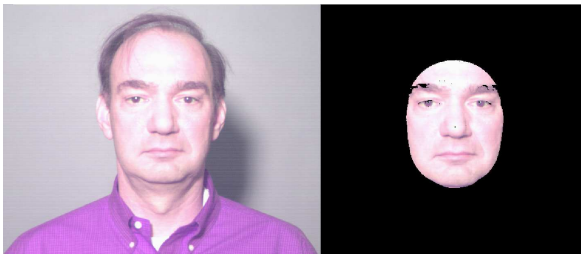

Fig. 7. *Left figure is the original face; right side is the cropped face using a sphere* $r = 100mm$*; the center of the sphere is at the nose tip.*

3. Face alignment

3.1. Face pose correction based on Principle Component Analysis

A 3D face appears as a 3D shape that has the most convex point at its center - the nose tip. The other parts of the face are very close to a cropped piece of barrel surface as shown as in figure 8. The length of c is shorter than the length of a and b and the length of a is longer than the length of b. That fact has been illuminated by A. Mian et al [21] and L. Zhang et al. [31].

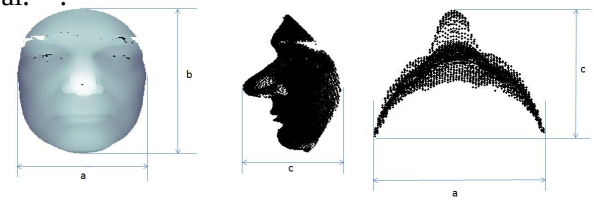

Fig. 8. *a,b and c are the width, height and depth of the 3D face surface.*

Thereupon, according to the distribution information of points such as a,b and c, the top three largest principle components can be used as x, y and z coordinates axes. Then poses of all faces theoretically can be aligned into a consistent coordinate system. Firstly, let $p_i(x_i, y_i, z_i)$ $1 \leqslant i \leqslant n$ represent a point within a face surface S, which has n points. Taking m as the mean vector of all p_i. Then the covariance matrix C can be given by:

$$C = \frac{1}{n}\sum_{i=1}^{n}(p_i - m)(p_i - m)^T \tag{8}$$

By performing PCA on the covariance matrix C, a matrix V of eigenvectors and a diagonal matrix D

of eigenvalues are given by:

$$CV = DV \tag{9}$$

Then three eigenvalues $\lambda_1 \geqslant \lambda_2 \geqslant \lambda_3$ and three corresponding eigenvectors v_1, v_2 and v_3 can be computed. Due to the particular shape of the cropped face, the smallest distribution of the point cloud of a face is along the normal direction of the face surface. Consequently, the eigenvector v_3 represents the normal direction and the v_1 and v_2 are the vertical and horizontal dimension directions. By means of PCA, the matrix V is also a rotation matrix to convert the coordinates of S to be its principal axes:

$$S_{new} = V(S - m) \tag{10}$$

After PCA pose correction, most faces are at a good front view position. However, some faces are not correctly aligned due to the asymmetric shape produced by different hair styles. In some cases, surface loss at some positions will cause misalignment. Additionally, distortion of the face also will affect the accuracy of the face alignment.

3.2. *Face alignment based on the symmetry of the human face*

A human face can be considered as a symmetric surface along the OYZ plane as shown in figure 9. Inspired by [31] and [22], face alignment based on the Iterative Closest Point(ICP) algorithm can be optimized by utilizing the symmetry of the face. The iterative closest point algorithm algorithm (ICP) is widely used for geometric alignment of 3D models. ICP is a method to fit a target cloud of points to another cloud of points which constitute a model image. The whole idea of ICP is to minimize the sum of square error between target points and the model points, then estimate an appropriate transformation to align the target points to the model points. Besel et al. [4] proposed the first ICP algorithm and proved that the ICP algorithm always converges monotonically to the nearest local minimum of a mean-square distance metric. The smallest distances between each point in the target image and the points of model image are calculated to form a rotation matrix. This procedure is repeated until the squared error distance of the points of the target image to their closest points in the model image falls below a preset threshold.

If there is a target face: $F = (X_t, Y_t, Z_t)$, we can define a mirror face as the model face M:

$$M = F_{mirror} = (-1 \cdot X_t, Y_t, Z_t) \tag{11}$$

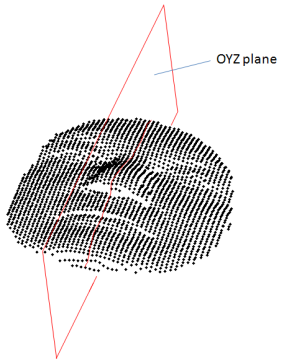

Fig. 9. *Human face is a symmetric surface about OYZ plane.*

By applying the ICP algorithm, the target face can rotate to fit the model face if the mirror face is used as the model face. The rotation matrix and the transformation matrix can be calculated and obtained. According to the fundamentals of computer graphics [11], every 3D rotation is a composition of three rotations about the x-axis, y-axis and z-axis:

$$R = R_y(\theta) \cdot R_x(\alpha) \cdot R_z(\beta) \tag{12}$$

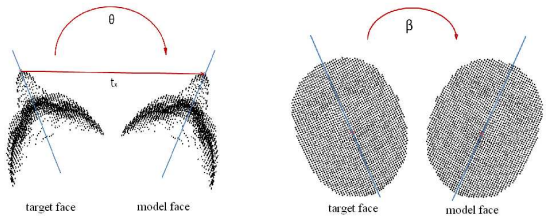

Fig. 10. *Rotations along y-axis(left figure) and z-axis(right figure) from the target face to model face(mirror face).*

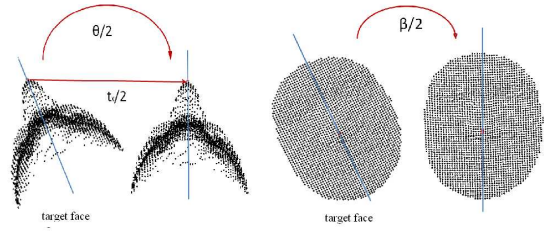

Fig. 11. *The target face is aligned to a perfect front view position according to the θ and β generated by applying ICP to rotate target face to model face (mirror face).*

Since the model face is the mirror face of the target face along the *oyz* plane, the rotation angle α along the *x*-axis is equal to zero and there are two rotations left as shown in figure 10. If the target face is rotated by angle $\frac{\theta}{2}$ along the *y*-axis and angel $\frac{\beta}{2}$ along the *z*-axis, the aligned face is just at a front view pose as shown in figure 11. After applying the ICP algorithm between the target model and the mirror model, a rotation matrix R and a transformation matrix T can be calculated. Given the rotation matrix R, we can calculate the three angles α, θ and β respectively. As we already know that the model face is the *x* mirror of the target face, the rotation along *x*-axis is almost equal to zero. The composite rotation is mainly formed by rotations about the *y*-axis and *z*-axis. If there is a rotation defined as follows:

$$\alpha_{new} = 0 \tag{13a}$$

$$\theta_{new} = \frac{\theta}{2} \tag{13b}$$

$$\beta_{new} = \frac{\beta}{2} \tag{13c}$$

The transformation matrix $T = [t_x, t_y, t_z]$ can be calculated by applying ICP algorithm. Then the new transformation matrix can be created as:

$$T_{new} = [\frac{t_x}{2}, 0, 0] \tag{14}$$

Then we can apply the rotation according to the new rotation matrix R_{new} and the transformation matrix T_{new}. The target face is aligned to a new position by applying the rotation:

$$F_{new} = R_{new} \cdot F + T_{new} \tag{15}$$

Even when the automatic localized position of the nose tip has a certain distance to the real nose tip that is exactly on the symmetry plane, the error distance along *x*-axis of the nose tip to the real position is neutralized because of the calculation of $\frac{t_x}{2}$ as shown as in figure 12. Thus, another effect of this rotation is that the error distance of the automatically localized nose tip position along *x*-axis is further reduced towards zero.

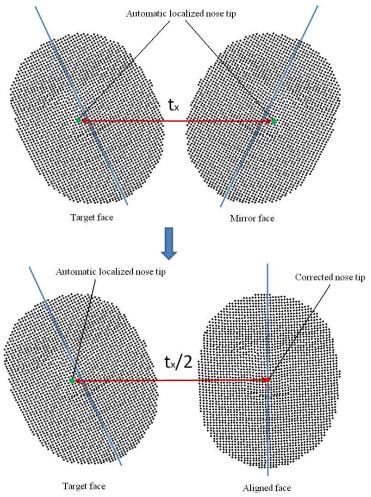

Fig. 12. *The position of the nose tip is further corrected by implementing* $[\frac{t_x}{2}, 0, 0]$ *as the transformation matrix.*

Facial expression variations could generate some asymmetric shapes, which will affect the mirror face alignment. However, most facial expressions occur in the area near the mouth and the facial region around the nose tip is the area least affected by expression variations. Consequently we can use a sphere around the nose tip to crop a piece of the face surface as a relatively expression-invariant and symmetric area. Additionally hair also may affect the symmetry of this area. Thus we choose $45mm$ as the radius of this sphere to avoid the effect of hair and keep the symmetry of this area.

Finally, implementing the face alignment using the symmetry of human face has two outcomes: 1. Error distance of the localized nose tip position along the *x*-axis is reduced towards zero. 2. Face misalignments along the *y*-axis and *z*-axis are minimized.

3.3. ICP face alignment using expression-invariant regions

After face alignment based on the symmetry of the human face, the misalignment along the *x*-axis is still not aligned and there is still an error in the automatic localized position of the nose tip along the *y*-axis and *z*-axis. On the other hand, human faces share relatively similar facial features and structure. So it is possible to align a face to another face by adjusting its rotation to a standard position. If the slight imprecision of the alignment caused by

the variations of facial expression is temporarily ignored, the faces from the same individual share a common shape. Thus, when those faces are fitted to a standard face template which is from another individual, their alignments will appear very close to be the same. Every facial feature is aligned to almost the same position. That result also can be used to further improve the accuracy of the nose tip detection. Since faces belonging to the same person share more elements in common than faces from different individuals, the facial features, especially the nose tip, if they are from the same people, will be corrected to similar positions. In order to reduce the number of misalignments caused by expressions, it is required that the parts of the face insensitive to expressions are used in the alignment. In face alignment based on the symmetry of the face, the misalignments along y and z-axis have been minimized. As a result, we can define a region shown in figure 13 Only points near the nose tip and above the eyes (within a sphere $r = 70mm$) are used in the ICP alignment just because the nose, eyes and the forehead regions are the least affected by expressions in 3D shapes. The expression-invariant region cropped in the standard face template is slightly (radius=75mm) larger than the corresponding region of the target face to avoid unexpected incorrect results.

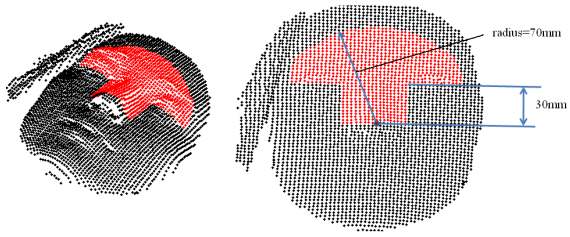

Fig. 13. *When apply the ICP algorithm, only the points within the red region of the target model are used.*

Such a region which is on the upper face could be affected by hair noise as shown as in figure 14. Hair style variations may cause asymmetric shapes. Fortunately, we have aligned the face according to the symmetry of the face. The shape of a face especially in the expression-invariant region should be a symmetric shape. So, the z value of a certain point should equal its corresponding point on the mirror side. Consequently, the hair can be detected by finding the much larger z values (by defining a threshold) compared to the corresponding points of the mirror side of the face. Then those points are removed before applying the ICP algorithm in case those points affect the alignment.

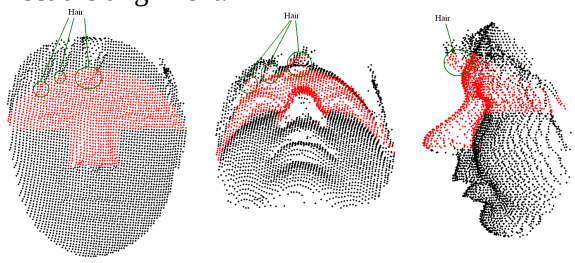

Fig. 14. *The hair could damage the symmetry of the shape in expression-invariant region. The hair noise also could affect the results of ICP-based alignment.*

Unlike other face alignment approaches [19 29 10] based on the ICP algorithm, which used the whole composite rotation matrix to rotate the target face, we only use the information about rotation along the x-axis to align the target face. Given a composite rotation matrix generated by the ICP algorithm, we can obtain the rotation angles α, θ and β along the x, y and z-axis. Since we have minimized the misalignments on the y-axis and z-axis in the face alignment based on the symmetry of the face, here we only need the α along the x-axis to align the target face. Then the rotation matrix R can be calculated by using $R = R_x(\alpha)$. And the transformation matrix T can be computed as: $T = [0, y_{template}, 0]$, where $y_{template}$ is the y value of the nose tip of the standard face template.

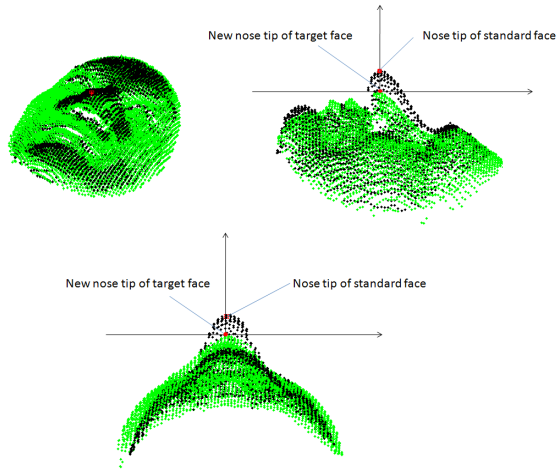

Fig. 15. *Nose tip re-localization. Green face is the target face and the black face is the standard face template. Using the y value of the nose tip position of standard face template and the original x value to locate the new nose tip position.*

Then we can implement the composite rotation by using equation 15. Furthermore, after applying ICP alignment, the nose tip of the target face is re-localized by using the nose tip of the standard face template. The new position of the nose tip uses the *y* value of the nose tip position of the standard face template plus its own *x* value of the nose tip to find the closest *z* value within the target face. Figure 15 demonstrates an example of how to implement nose tip re-localization. This process can further improve the accuracy of the nose localization, especially the nose position accuracy between faces belonging to the same individual simply because those face share a similar shape. After this ICP-based alignment using the expression-invariant region, all faces are precisely aligned into a perfect front view position even along all of the *x*, *y* and *z*-axis. Defining the re-localized nose tip as the zero point of the coordinate system, all faces are shifted into the same coordinate system.

4. Experiments results

4.1. Results of 3D nose tip localization

In this paper, the FRGC dataset is chosen as the experimental database. The face images of the FRGC database are segmented into training and validation partitions. The training set contains 3D scans, and controlled and uncontrolled still images from 943 subject sessions. The validation partition is designed as target subsets, there are 4,007 subject sessions of 466 subjects. Each subject session has a 3D scan file containing 3D points and a 2D still image file representing texture information. The resolution of faces in the FRGC dataset is 640×480. In order to reduce the cost of computation, we resize the 2D and 3D file to 160×120. The resized 3D files are smoothed to delete the spikes and to fill in the unexpected holes by using a similar technique to that proposed by Mian et al. [20]. Firstly, we remove spikes from the face surface by locating outlier points. Any point whose distance is greater than a certain threshold d from any of its neighbouring points will be considered as a spike point. d is defined using $d = \mu + 0.6\sigma$, where μ is the mean distance between neighbouring points and σ is the standard deviation. The holes caused by the removal of spike points can be compensated and filled by using cubic interpolation. 40 3D faces are selected from the Spring2003 subset as the training set. Those 40 faces are from 40 individuals including different races, genders and numbers of points. 4007 faces from the Fall2003 subset and the Spring2004 subset are used as target groups. Since there are 139 faces with very poor 2D-3D corresponding, 3868 faces having good 2D-3D corresponding are selected to more precisely evaluate the performance.

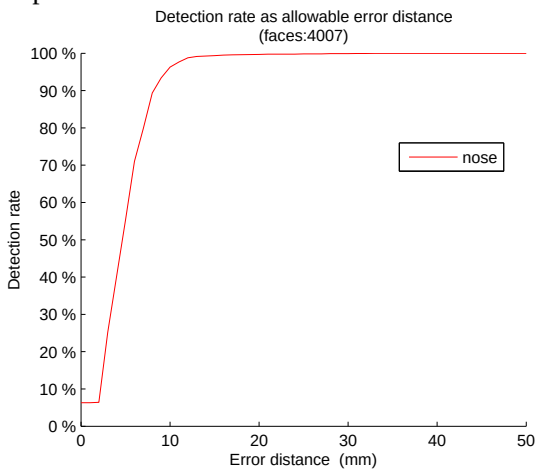

Fig. 16. *Error distance curves for the nose tip identification of all faces.*

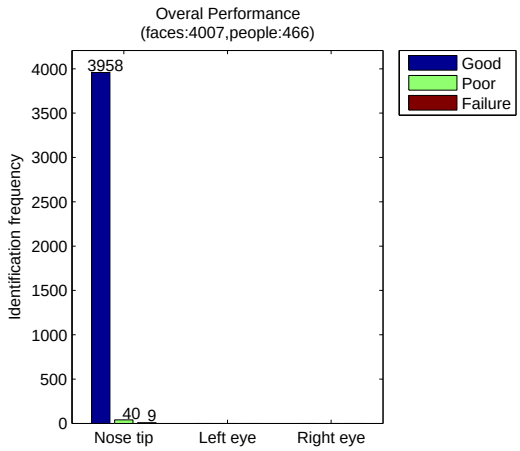

Fig. 17. *Histogram of the identification frequency for the nose tip identification of all faces.(Good:*$0-12mm$*;poor:*$12-24mm$*;failure:*$>24mm$*)*

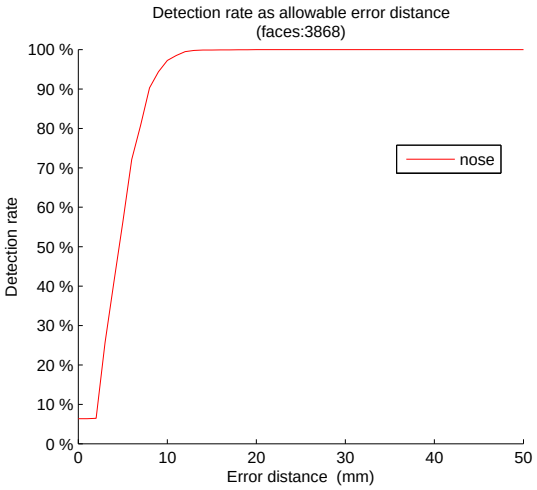

Fig. 18. *Error distance curves for the nose tip identification on faces with good 2D-3D corresponding.*

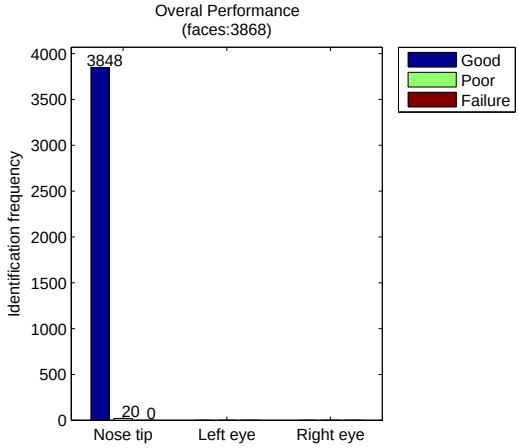

Fig. 19. *Histogram of the identification frequency for the nose tip identification on faces with good 2D-3D corresponding.*

The face detection depends on how accurate the nose tip localization is, so we can use the performance of the nose tip localization to represent the accuracy of the face detection. Using the position of the nose tip in the ground truth data as the standard position, the error distance between the localized nose tip and the ground truth data can be calculated. Figure 16 shows the cumulative error distance curve of all faces in the FRGC v2 database. Figure 17 shows the histogram of the identification frequency of all faces in the FRGC v2 database. We have to emphasize that even a large error distance does not always mean that the nose tip localization of this face is fail due to the mismatching between 2D and 3D channels in some face images. By manually checking all positions of the localized nose tip in 3D channel, our nose tip localization method only fails in two faces which actually have no nose tip at all. If we use all faces (4950 faces) in three subsets of the FRGC database as the experiment dataset, 99.96% nose tips can be correctly localized. When the ground truth data are used to evaluate 3868 faces having good 2D-3D corresponding, 100% nose tips are correctly localized. Figure 18 and figure 19 shows the error distance curve and the histogram of the identification frequency of those faces.

Unlike some techniques making use of the texture information in the 2D face detection, this approach is a pure 3D shape analysis which is naturally invariant to illumination variations. It is also an orientation-invariant method. The range of rotation around z-axis can be 0^0-360^0. In order to compare with approaches using all faces in FRGC database including v1 and v2 datasets, the nose tip localization is also implemented on FRGC v1 database, the nose tip detection rate is 100% on 943 faces. Thus the nose tip detection of whole FRGC database is 99.96%(2 failures out of 4950). Compared with results using other state-of-the-art techniques, our approach achieved the highest detection rate of the nose tip localization, shown in table 1.

Table 1. *Details in comparison with state-of-the-art techniques.*

Manually check results	
FRGC v2 (4007 faces)	Identification rate
Our approach	99.95%
Segundo [26]	99.95%
Faltemier [10]	98.20%

Manually check results	
FRGC v1&v2 (4950 faces)	Identification rate
Our approach	99.96%
Mian [20]	98.3%

Compared with ground truth data	
FRGC v2	Identification rate
Our approach (3868 faces)	100%
Pears [23] (3680 faces)	99.92%

4.2. Evaluations of 3D face alignment

Using the results of the nose tip localization, the main face area can be cropped. Then we apply the PCA pose correction on the FRGC v2 database. About 10% of the 4007 faces appear to have a certain misalignment. By applying the integrated face alignment approach, no observable misalignment is found during the manual check. However, it is not easy to compare the performance of our face alignment method with other state-of-the-art techniques. In this section, we try to evaluate the within-class and between-class differences of all faces in the FRGC v2 database by comparing different face alignment approaches. We separate the FRGC v2 face database into two categories: neutral faces (2182 faces) and non-neutral faces (1825 faces) to test the performance of correcting face pose and the ability to handle the expression variations. We classify the current state-of-the-art techniques into four types and then use the following methods to simulate those four face alignment techniques.

1. PCA-based face alignment using the whole face area which is introduced in section 3.1 (a similar method is used in [21]).
2. Face alignment using the ICP algorithm to fit the whole target face to a standard face template (similar methods are used in [10 16]).
3. Face alignment using the ICP algorithm to fit a sphere (r=45mm) area around the nose tip of the target face to a standard face template (a similar method is used in [29]).
4. Face alignment using the ICP algorithm to fit the expression-invariant area of the target face to a standard face template (a similar method is used in [19]).

Table 2. *Comparison the MSE between faces belonging to the same individual by using different face alignment approaches in (within class performance).*

Methods	MSE(Neutral)	MSE(Non-neutral)
1	0.5033*mm*	0.5793*mm*
2	0.2594*mm*	0.3327*mm*
3	0.3186*mm*	0.4358*mm*
4	0.2729*mm*	0.3084*mm*
Our method	0.1940*mm*	0.2550*mm*

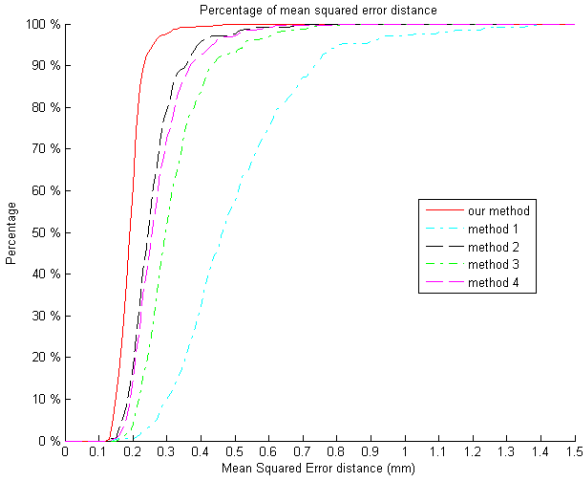

Fig. 20. *Cumulative percentages of the within-class Mean Squared Error Distance of neutral faces.*

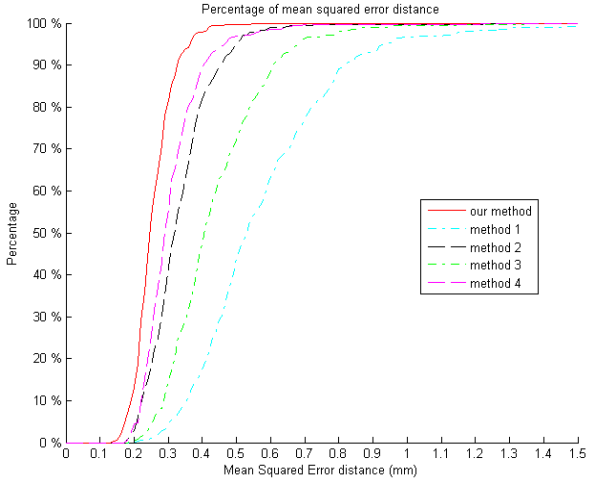

Fig. 21. *Cumulative percentages of the within-class Mean Squared Error Distance of non-neutral faces.*

Since the expression-invariant regions of faces belonging to the same people share similar shapes, we can use the differences of the expression-invariant region between faces of the same individual to represent how good the face alignment is. It is also an indicator of the within-class difference. We can calculate the mean squared error distance (MSE) between the corresponding points within the expression-invariant region of faces belonging to the same individual. If a subject has *n* face images, we will calculate the MSE of every possible face-face combination. Then we compute the mean values of the error distances between the corresponding points (the closest points) of these face-face combinations. Table 2 shows the within-class MSE values of differ-

ent face alignment methods. Our method achieves the smallest within-class MSE values both in neutral faces and non-neutral faces. The cumulative percentages of the within-class MSE of neutral faces and non-neutral faces using different face alignment methods are shown in figure 20 and figure 21.

Table 3. *Comparison of rank-one identification rates (between-class performance).*

Methods	firs vs neutral	first vs non-neutral
1	27.71%	19.99%
2	63.60%	44.97%
3	47.42%	30.43%
4	53.83%	46.60%
Our approach	96.31%	85.29%

The MSE evaluation given above tests the within-class differences of these approaches. On the other hand, we can use the results of the identification experiment based on the results of different alignment approaches to compare the between-class distinguishing ability. In the FRGC v2 database there are 465 subjects. We select the first face images of each subject as the gallery dataset. The remaining face images are separated into two datasets: neutral faces and non-neutral faces. We define two rank-one identification experiments: "first face vs neutral face" and "first face vs non-neutral face". In the "first face vs neutral face" experiment, 1761 neutral faces consist of the test dataset and the gallery dataset includes all of the first face image (465 faces) of each individual in FRGC v2 dataset. Each face in the test dataset is matched to every face in the gallery dataset. If the match with rank-one similarity is a match between two faces belonging to the same person, this match is considered as a correct match, otherwise it is an incorrect one. So there are 1761×465 matches. In the "first face vs non-neutral faces" experiment there are 1781×465 matches. To generate the similarity score of a match, we use the mean squared error distance method to measure the similarity between expression-invariant regions of two faces. The mean squared error distance method is also used in the ICP-based face recognition approach [10 21]. Table 3 shows the results of these two experiments. We find that our approach outperforms the other methods both in "neutral faces vs neutral faces" and "non-neutral faces vs non-neutral faces" experiments.

5. Conclusion

This paper presented an automatic 3D face detection and registration approach. We use kNN AURA algorithm to identify and localize a key facial features - the nose tip to detect the main face area. A 99.96% identification rate of the nose tip localization in a large dataset(FRGC v2) with expression variations demonstrated the robustness and effectiveness of this method. Excepts two noseless faces, 100% nose tips are correctly identified. After then, we proposed an integrated ICP-based approach to align faces even with expression variations. We firstly use PCA to roughly correct some server misalignment, then align face by using the symmetry of the face minimizes the possibility of misalignments along the y and z-axis and reduce the error distance of the automatically localized nose tip position along x-axis to zero. The expression-invariant region can be extracted. Finally, a face alignment based on ICP algorithm using the expression-invariant region produces the rotation angle α along the x-axis. In order to evaluation the result of face registration, we propose a method to measure the within-class and between-class performance in face registration/alignment. In the comparison with state-of-the-art face alignment techniques based on FRGC v2 dataset, our approach achieves the best performance. Our approach is a full automatic method both in face detection and registration without manually amending the results during the process. It builds a good foundation even for the further face recognition phase.

6. Acknowledgments

This work is supported by the Science&Technology Research Fund of Henan Provincial Educational Department(No.13A520033).

References

1. J. A. Anderson. A simple neural network generating an interative memory. *Mathematical Biosciences*, vol

14:197–220, 1972.
2. M. Ankerst, G. Kastenmuller, H. Kiegel, and T. Seidl. 3d shape histograms for similarity search and classification in spatial databases. *SSD'99*, pages 207–226, 1999.
3. J. Austin. Distributed associative memories for high speed symbolic reasoning. *IJCAI'95 Working Notes of Workshop on Connectionist-Symbolic Integration: From Unified to Hybrid Approaches*, pages 87–93.
4. P. J. Besl and N. D. Mckay. A method for registration of 3-d shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no.2:239–256, 1992.
5. V. Bevilacqua, P. Casorio, and G. Mastronardi. Extending hough transform to a points' cloud for 3d-face nose-tip detection. *Lecture Notes in Computer Science*, vol 5227/2008:1200–1209, 2008.
6. K. Bowyer, K. Chang, and P. Flynn. A survey of approaches and chanllenges in 3d and multi-model 3d+2d face recognition. *In: CVIU*, 101:1–15, 2006.
7. C. S. Chua and R. Jarvis. Point signature: A new representation for 3d object recognition. *Internat. J. Computer Vision*, 25 (1):63–85, 1997.
8. A. Colombo, C. Cusano, and R. Schettini. 3d face detection using curvature analysis. *Pattern Recognition*, vol. 39, number 3:444–455, 2006.
9. C. Conde, A. Serrano, L. Rodriguez-Aragon, and E. Cabello. 3d facial normalization with spin images and influence of range data calculation over face verification. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition,*, vol 3, issue 20-26, 2005.
10. T. Faltemier, K. W. Bowyer, and P. J. Flynn. A region ensemble for 3d face recognition. *IEEE Trans. Inf. Forensics Security,*, vol. 3, no. 1:62–73, 2008.
11. J. D. Foley and A. V. Dam. *Fundamentals of Interactive Computer Graphics*. Addison-Wesley Systems Programming Series, 1983.
12. W. Grimson and T. Lozano-Perez. Model-based recognition and localization from tactile data. *IEEE International Conf. on Robotics, Atlanta, GA*, 1984.
13. D. O. Hebb. The organization of behavior. 1949.
14. C. Hesher, A. Srivastava, and G. Erlebacher. A novel technique for face recognition using range imaging. *Proc. IEEE Int. Symposium on Signal Processing and Its Applications*, 2003.
15. V. J. Hodge and J. Austin. A binary neural k-nearest neighbour technique. *Knowledge and Information Systems*, vol 8, number 3:276–291, 2005.
16. I. Kakadiaris, G. Passalis, G. Toderici, N. Murtuza, and T. Theoharis. Three-dimensional face recognition in the presence of facial expression: An annotated deformable model approach. *IEEE Trans. Pattern Anal. Mach. Intel.*, vol 29, no.4:671–680, 2007.
17. T. Kohonen. Correlation matrix memories. *IEEE Transactions on Computers*, vol 21:353–359, 1972.
18. Y. Lee, K. Park, J. Shim, and T. Yi. 3d face recognition using statistical multiple features for the local depth information. *In:Proc. 16th internat. Conf. Vision Interf.*
19. X. Lu, A. K. Jain, and D. Colbry. Matching 2.5d face scans to 3d models. *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no.1:31–43, 2006.
20. A. Mian, M. Bennamoun, and R. Owens. Automatic 3d face detection, normalization and recognition. *Interantional Symposium on: 3D Data Processing Visualization and Transmission*, vol 0:735–742, 2006.
21. A. Mian, M. Bennanmoun, and R. Owens. An efficient multimodal 2d-3d hybrid approach to automatic face recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 11:1927–1943, 2007.
22. G. Pan, Y. Wang, Y. Qi, and Z. Wu. Finding symmetry plane of 3d face shape. *Proc. of 18th International Conference on Pattern Recognition (ICPR'06)*, vol 3:1143–1146, 2006.
23. N. Pears, T. Heseltine, and M. Romero. From 3d point clouds to pose-normalised depth maps. *International Journal of Computer Vision*, Volume 89, Numbers 2-3,:152–176, 2010.
24. P. J. Phillips, W. T. Scruggs, A. J. O'Toole, P. J. Flynn, K. W. Bowyer, C. L. Schott, and M. Sharpe. Frvt 2006 and ice 2006 large-scale experimental results. *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32 no. 5:831–846, 2010.
25. M. Romero and N. Pears. 3d facial landmark localisation by matching simple desciptors. *2nd IEEE Int. Conf. Biometrics: Theory, Applications and Systems*, 2008.
26. M. P. Segundo, C. Queirolo, O. R. Bellon, and L. Silva. Automatic 3d facial segmentation and landmark detection. *Image Analysis and Processing, ICIAP*, pages 431–436, 2007.
27. F. Stein and G. Medioni. Structural hashing: Efficient three dimensional object recognition. *Proceedings of Computer Vision and Pattern Recognition*, pages 244 – 250, 1991.
28. Y. Wang, J. Liu, and X. Tang. Robust 3d face recognition by local shape difference boosting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:1858–1870, 2010.
29. C. Xu, S. Z. Li, T. Tan, and L. Quan. Automatic 3d face recognition from depth and intensity gabor features. *Pattern Recognition*, 42(9):1895–1905, 2009.
30. C. Xu, T. Tan, Y. Wang, and L. Quan. Combining local features for robust nose location in 3d facial data. *Pattern Recognition Letters*, vol 27, issue 13:1487–1494, 2006.
31. L. Zhang, A. Razdan, G. Farin, J. Femiani, M. Bae, and C. Lockwood. 3d face authentication and recognition based on bilateral symmetry analysis. *The Visual Computer*, 22:43–55, 2006.