# 3D Angle Searching System with PSO for Face Recognition

Ching-Tang Hsieh[1,a], Chia-Shing Hu[2,b] and Meng-Shian Shih[3,c]

[1]Department of Electrical Engineering, Tamkang University, New Taipei City, 25137, Taiwan, R.O.C

[2]Department of Electrical Engineering, Tamkang University, New Taipei City, 25137, Taiwan, R.O.C

[3]Department of Electrical Engineering, Tamkang University, New Taipei City, 25137, Taiwan, R.O.C

[a]hsieh@ee.tku.edu.tw, [b]894350106@s94.tku.edu.tw, [c]792350075@s92.tku.edu.tw

**Keywords:** PSO, FACE RECOGNITION.

**Abstract.** Conventional 2D face recognition methods often struggle when a subject's head is turned even slightly to the side. In this study, a face recognition system based on 3D head modeling that is able to tolerate facial rotation angles was constructed by leveraging the Open source graphic library (OpenGL) framework. To minimize the extensive angle searching time that often occurs in conventional 3D modeling, Particle Swarm Optimization (PSO) was used to determine the correct facial angle in 3D. This reduced the angle computation time to 6 seconds, which is significantly faster than other methods. Experimental results showed that successful ID recognition can be achieved with a high recognition rate of 90%.

## 1. Introduction

Facial recognition has become a popular yet challenging area in the discipline of pattern recognition. Human faces contain various important characteristics, which can be used for personal identification. Facial recognition can be used for a wide variety of purposes such as target tracking [1], feature identification[2] and post image processing[3]. Conventional studies on facial recognition have mostly focused on 2D image processing, which struggles when faced with facial images taken from an angle. Through the use of 3D modeling with Particle Swarm Optimizations (PSO), face recognition can now offer better tolerance for rotation angles, making its application more robust and more efficient. In addition, by leveraging the Open source graphic library (OpenGL) framework, 3D models of human heads can be stored in a library for rapid loading and computation.

## 2. Structure of the system

The facial recognition process begins with the input of three images of the subject into the system: one image taken from the front, one from the side, and the last one from an arbitrary angle. These images are then used by FaceGen[4], a 3-D face creation software, to generate a 3D head model of the subject. Manual adjustment is then performed. The overlay function is used to place the original 2D sample image over the 3D head model, with the appropriate transparency level set. The size of the head model is then compared with the 2D sample image and adjusted accordingly. Key reference points used during comparison include the location of the eyes, point of the nose, lips and outline of the face.

After importing the adjusted 2D image into OpenGL, an image thresholding method is applied to the image together with the existing 2D images in the graphic library to obtain binary images. Error calculation is performed to find points of difference between these images. The image in OpenGL with the least amount of error is said to have the correct angle of rotation that matches the 2D sample image. The facial rotation angles in the XYZ dimensions are obtained from this image.

By rotating the FaceGen 3D head model with the facial rotation angles, copies of 2D images are produced. Finally, images from FaceGen, 2D images from OpenGL and the adjusted 2D image are transformed into binary images. Error calculation is once again performed on all the images. The image with the lowest error energy is selected and used for ID confirmation. The full process is shown in Figure 1.

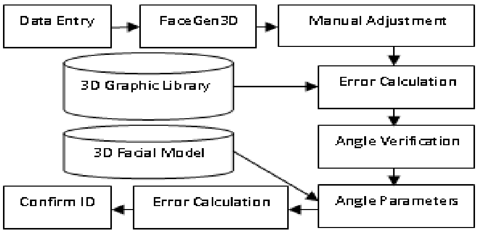


Figure. 1 System Overview

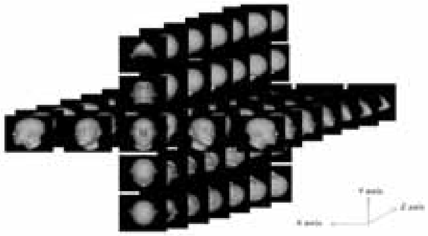

Figure. 2 OpenGL image database for 3D head models

## 2.1 Process for 3D Head Modeling

Two images are required before a 3D head model is constructed. One of the images needs to be taken from the front, with features including the eyes, the nose and the mouth clearly shown, for a total of 11 mark points. The other image needs to be taken from the side, with the facial curvature tracked by a total of 9 mark points. After internal computation in FaceGen, a head model with approximately 65,000 vertices is made.

Open Graphic Library (OpenGL) was chosen for our system due to its computation power. As a 2D/3D graphics API, developers can use their own data, as well as data from other sources to generate 3D graphical images. OpenGL supports file formats for popular programs such as AutoCAD and 3DS, making the file conversion fast and flexible. In our system, when loading 3D models onto the computer, the loading time in OpenGL was 0.5 second, significantly faster than using MATLAB or C++ frameworks, which had loading time of 33 seconds and 3 seconds respectively.

After loading the 3D head model, 2D images are generated for each rotation angle along the XYZ direction, ranging from -90 to 90 degree, with the image at [0,0,0] representing the subject facing the front of the screen. A total of 5,929,741 images at a resolution of 200x200 pixels are generated for each 3D head model, requiring 700GB of hard disk space to store the data. Figure 2 illustrates a snap shot of the image database for 3D head models.

## 2.2 Error Calculation

Error calculation is performed on image particles to determine the error energy. Images of two subjects taken from identical angles are judged to have the minimum amount of error energy, whereas images of the same two subjects taken from different angles contain greater amounts of error energy. Thus, in order to calculate the error energy, it is important to obtain the correct rotation angles for the images. Error energy is defined by the following equation:

$$\text{Energy} = \sum_{xy} | I_{input}(x,y) - I_{particle}(x,y)| \quad (1)$$

## 2.3 Particle Swarm Optimization

The concept of Particle Swarm Optimization (PSO) was first proposed by J. Kennedy and R. Eberhart in the 1995 IEEE conference [5]. As an optimization method based on population dynamics, PSO simulates social behaviors. In a social population, it is believed that individual behavior is not only affected by the experience and mind set of the individual, but also by the society to which the individual belongs. In PSO, each particle has its own searching speed within its own searching area, and its searching strategy depends on past searching experience of the particle, as well as the behavior of the group. Every individual within the searching group exchanges information with each other in a similar fashion to the way birds search for their food source, as shown in Figure 3.

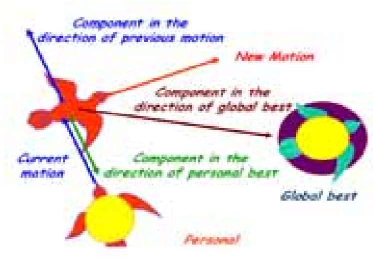


Figure. 3 PSO method illustrated in food searching strategies for Birds

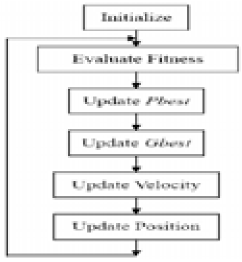


Figure. 4 PSO calculation Process

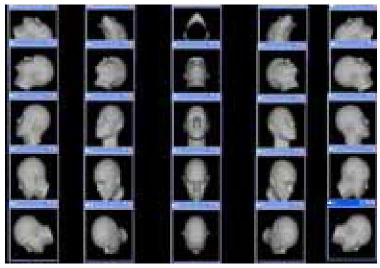

Figure. 5 Particle distribution of the X-Y plane

The orange bird on the left corner represents the current location of the bird. The yellow circle at the bottom indicates where the bird previously had its best food searching experience ($P_{best,}$ , Personal Best). The purple circle on the right represents the best food location for the entire group of birds ($G_{best}$, Global Best). This bird is currently moving in towards the upper left along the blue arrow. The green arrow- pointing to the lower right hand corner shows where the bird should move according to its $P_{best}$ value, and the brown arrow towards the right hand side shows where the bird should move according to the $G_{best}$ value. Taking all the parameters into consideration, this bird is now heading in the upper right hand corner along the red arrow. The calculation process requires every particle to remember its previous searching values in order to learn from the past experience. The value for $G_{best}$ provides a reference for the particle to gradually adjust its position towards the best location based on its $P_{best}$ value and the $G_{best}$ value.

The calculation process in PSO involves initialization, evaluations, updating $P_{best}$, updating $G_{best}$, updating searching speed (velocity), and updating position, as shown in Figure 4.

In the initialization process, depending on the stability of the system, the number of particles in the group needs to be specified. For a relatively stable system with minimum gaussian distribution occurring at the peak and valley region, only a moderate number of particles is needed. Instead of assigning a value of 0 to each particle, a random value, *Rand*, is generated and used as a system variable. *Rand* ranges from 0 to 1, and is used to improve the convergence of the system.

Some systems aim to search for a maximum value, whereas others search for a minimum value, therefore in the process of evaluating fitness, choosing the most suitable values for the variables depends on the optimization equations in the system. In our system, the fitness function is defined as below, where Energy is the error energy.

$$\text{Fitness} = \min(\text{Energy}) \tag{2}$$

The $P_{best}$ value is updated according to the past calculated data for each particle, and $G_{best}$ is determined by summarizing and evaluating all the $P_{best}$ values from all the particles. The current $G_{best}$ is compared with the previous $G_{best}$ and updated accordingly. When every particle finishes calculating its $P_{best}$ and $G_{best}$, the following equation is used to adjust the next movement for the particles.

$$V_i\,(t+1) = w \cdot V_i(t) + C_1 \cdot rand(P_{best} - V_i(t)) + C_2 \cdot rand(G_{best} - V_i(t)) \tag{3}$$

Where $V_i$ (t+1) is the velocity for the next search, and $V_i$ (t) is the velocity from the last search. The inertia weight, w, determines the speed for the particle to move. To improve stability as the system approaches the optimal global solution, a method (equation 4) to determine the value of w was developed based on previous work[6],[7] and used in our system. As the number of iteration increases, the value of the inertia weight w is decreased, as shown in equation below.

$$w = w_{max} - (w_{max} - w_{min}) \cdot iter / iter_{max} \tag{4}$$

In equation (3), *iter* is the number of current iteration and $iter_{max}$ is the maximum number of iterations. $C_1$ represents the cognitive ability for each particle to retain data for its previous search results, and $C_2$ represents the social communication ability in between particles. A larger value of $C_1$ indicates this particle would favor its own personal choice in the next search, making frequent stops on its path towards $G_{best}$, and thus increasing the total number of iterations necessary. The advantage would be that it reduces the likelihood of stopping at a local optimal solution. However, for simple systems with low complexity, this behavior could be a waste of computational effort. A larger value of $C_2$, on the other hand, indicates that the particle favors the global preference in its next search, which shortens its path to $G_{best}$, since the number of iteration is reduced and its stopping locations are minimized. For a system with less complexity, a large $C_2$ would be acceptable. Based on studies performed in [7], it is recommended that the sum of $C_1$ and $C_2$ should be greater than 4.

Once the value for $V_i$ (t+1) is calculated, the following equation is used to determine the next location for the particles.

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (5)$$

### 3. Face angle matching process using PSO

Referring to figure 1, PSO is used for both the error calculation and angle verification process in our system. A total of 15 steps are involved in the optimization process using PSO as shown in Figures 6 and 7. Head images with varying facial rotation models are created in 'FaceGen'and are then loaded into the PSO system and converted into 'particles, which are uniformly distributed throughout the XYZ co-ordinate system.To execute age synthesis, the following three steps are implemented: image normalization (face alignment), topography evaluation (landmark analysis), and age synthesis, as shown in Figure 1.

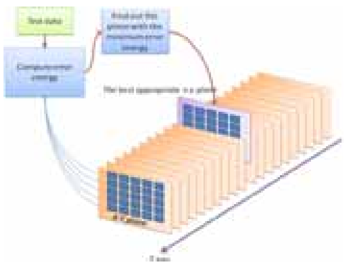

Figure. 6 PSO system flow chat

Figure. 7 Determination of the Optimal X-Yplane

Figure. 8 Estimation of facial rotation angle

In total, 180 planar slices through the Z axis are used, representing all of the possible facial rotational angles from -90° to 90°. Each plane is populated with 25 particles for a total of 4500 particles (Figure 5). PSO algorithms then optimize the error energy equations for each particle. If the error energy in that iteration is lower than the previous $P_{best}$ for that particle, then $P_{best}$ is updated. Likewise, if any of the $P_{best}$ values are lower than the previous $G_{best}$ values, then $G_{best}$ is updated accordingly for that iteration and if the $G_{best}$ value for the iteration is lower than the previous $G_{best}$, then the overall $G_{best}$ is updated. The velocity and subsequent movement of the particles can then be calculated. After ten iterations of this process, all particles except $G_{best}$ are discarded, and five further rounds of iteration are taken for 25 new particles immediately surrounding $G_{best}$.

After these steps have been completed for every X-Y plane, the plane with the lowest error energy (the lowest $G_{best}$ value) is found and the corresponding optimal facial rotational angle can be determined.

### 4. Results and discussions

A total of 30 subjects were recruited for the experiment, each with pictures taken from different angles with different poses. The average time to estimate the facial rotation angle from a test image was approximately 6 seconds. Success rate for recognition was above 90% after performing the test 10 times for each subject. Figure 8 shows the results for estimating facial rotation angles during the experiment.

Table. 1 The effect of C1 and C2 on iteration and success rate

| Test # | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Weight | 0.6 | 0.6 | 0.6 | 0.6 | 0.6 | 0.6 | 0.6 | 0.6 | 0.6 | 0.6 |
| C1 | 0.2 | 0.4 | 0.6 | 0.8 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 |
| C2 | 0.4 | 0.4 | 0.4 | 0.4 | 0.45 | 0.5 | 0.6 | 0.8 | 0.9 | 1.0 |
| Iteration | 23 | 26 | 25 | 26 | 22 | 20 | 19 | 18 | 17 | 16 |
| Success Rate % | 98 | 98 | 98 | 97 | 97 | 95 | 88 | 81 | 64 | 31 |

Table. 2 Comparison of results from several face recognition methods

| | Hierarchical Search[5] | Genetic Algorithm | Proposed method(PSO) |
|---|---|---|---|
| Avg. converge iterations | 140 | 95 | 25 |

The values for C1, C2, which alter the preference for particles towards their own personal best ($P_{best}$) or the group best ($G_{best}$) respectively, were varied to test their effect on the system. Changes in C1 did not have a significant effect on the system. The number of iterations increased slightly leading to a minor delay in the final calculation. This shows that if each particle is 'self centred' and relies primarily on its own personal experience, it takes longer for it to approach the global solution. On the other hand, when $C_2$ was increased, the global solution was reached more quickly, showing the value of collaboration between particles. Another finding was that when the speed of particle movement between iterations ($V_i$) exceeded a certain level, for every iteration that followed, particles were unable to approach the true global solution, because the particles moved too quickly and the success rate dropped (Table 1).

This situation occurred when the C2 value was set higher than C1. Therefore, for the purposes of accurate identification of individuals based on facial images, the results show that a PSO approach that favors the individual experience of particles is more effective. This is achieved in our system by setting $C_2$ to be less than or equal to $C_1$.

When PSO was compared with implementations using conventional methods, including Hierarchical Search [3] and Genetic Algorithm, the results showed that PSO offers a faster and more reliable solution for face recognition.

## 5. Conclusions

In this paper, a 3D face recognition system based on PSO is proposed. Experiments have shown that the system is able to quickly and reliably achieve successful face recognition for the purpose of ID confirmation. The system successfully determined the facial rotation angle in an average of 6 seconds with a high recognition rate of 90%.

References


[1] Jin Young Kim, Dae Young Ko, Seung You Na. : Implementation and enhancement of GMM face recognition systems using flatness measure, 13th IEEE, Robot and Human Interactive Communication, (2004), p.247 – 251.

[2] Baochang Zhang, Shiguang Shan, Xilin Chen, Wen Gao. : Histogram of Gabor Phase Patterns (HGPP): A Novel Object Representation Approach for Face Recognition, Image Processing, IEEE Trans. **V**ol. 16 (2007)**,** p. 57-68.

[3] Yanwei, Pang, Yuan Yuan, Xuelong Li, "Gabor-Based Region Covariance Matrices for Face Recognition," IEEE Trans. on circuits and systems for video technology, **V**ol. 18 (2008)**.**

[4] http://www.facegen.com, © Singular Inversions Inc.(2008).

[5] Eberhart, Russell, Kennedy, James. : Particle swarm optimization, IEEE International Conference on Neural Networks - Conference Proceedings 4, (1995)**,** p. 1942-1948.

[6] S. Cui and D.S. Weile. : Application of a Parallel Particle Swarm Optimization Scheme to the Design of Electromagnetic Absorbers, IEEE Transactions on Antennas and Propagation, Vol. 53 (2005), p. 3616-3624.

[7] J.G. Vlachogiannis and K.Y. Lee. : Determining Generator Contributions to Transmission System Using Parallel Vector Evaluated Particle Swarm Optimization, IEEE Transactions on Power Systems, Vol. 20, No. 4, (2005), p.1765-1774.