International Conference on Internet Studies,
July 18-19, 2015, Tokyo, Japan

3D FACE MODEL CONSTRUCTION BASED ON KINECT FOR FACE RECOGNITION

Ching-Tang Hsieh,
Tamkang University, Taiwan
hsieh@ee.tku.edu.tw

Yi Huang,
Tamkang University, Taiwan
save2623@hotmail.com

Ting-Wen Chen
Tamkang University, Taiwan
alucardoom@hotmail.com

Li-Ming Chen
Tamkang University, Taiwan
ms0071799@hotmail.com

ABSTRACT

We propose a simpler and faster method to recognize face. First, we use Kinect to detect frontal face and get depth image information with face, then we portrayed face in OpenGL to construct a three-dimensional face model based on the depth information. The face model also retains texture information of the original face images, and to create a complete change depth of face. It has a good result of repairing the distortion in side face. We can get a set face images with different angles by the method proposed, In recognition part, we use PCA(Principal Component Analysis) to reduce the dimensions, and classified with SVM(Support Vector Machine). The experiments show that the side face recognition can have good results.

Keyword: Face Recognition, PCA, SVM, Kinect

1. INTRODUCTION

The research of face recognition began in the 1960s, it refers to use computer vision technology to determine identity through facial feature analysis and comparison of information is very popular in recent years, the research, which is widely used in access control, identity authentication and crime investigation.

However, face recognition is much difficult in biometrics, the main reason is the structure of the human face are very similar, and the shape of the structure containing the facial features are also very similar, this feature is advantageous for face detection and location, but it is disadvantageous in distinguish individual human [1].

Among the many problems encountered by face recognition, side face is a very difficult and challenging study. When converted to the side of the face, the feature will reducing, it is the reason to have low accuracy in recognition. And the image on the monitor, there does not deliberately face the camera when people are naturally moving. Therefore, it is difficult to identify people which only using a frontal face image. In case to build the database can identify successfully, it takes considerable
In order to save time on collecting data, we presents a method, to capture a frontal face images with color and depth information by Kinect, then using OpenGL to quickly construct 3D face model, and rotation 3D face model's angle to simulate the various angles of the face, the result is saved as 2D color face image, reducing dimension by PCA and using SVM to classification [2].

In this paper, we can simulate multiple different angles of the human face by a single face image, and 3D face model retain the integrity of the face information, experimental results show that there is a good result for face recognition.

2. RELATED WORK

3D face reconstruction methods can be categorized into the following ways: laser scanning, Stripe generator, Stereoscopy, Video sequence and Image reconstruction [6]. Among these methods image reconstruction is the most simple, low cost and low consumption time. Image reconstruction usually needs one front and one side face to construct 3D face model [7] and even only uses one front face to reach the goal.

Liu et al [7] sets up a feature set database including three face parts which are extracted with ASM and then combines to a basic 3D face model. Due to these face feature parts are consecutive in a 3D face model, this method will cause discontinuity among these face blocks after 3D face model deformed.

Heo et al [5] use a single 2D facial image and propose a method of feature points extraction and analysis from non-frontal face image based on EM (Expectation-maximization) algorithm. And use a basic 3D face model to create 3D facial images. However, non-frontal face feature information is somewhat insufficient. The feature points obscured must be used to guess the future approach to estimate.

Breuer et al [3] proposed a method of automation 3D face reconstruction from 2D facial image or video based on SVM. This method uses SVM to detect the position of the face and gets the information. They calculate whether influence of angle on face, and use estimate method to calculate angle if it is. Selecting characteristics information is which part of the face by using Gaussian distribution. Then use a 3D face model-based and combine with the characteristics information to reconstruct 3D face model. But its methods on the same 2D face model, according to the different angle of the original face, there may result in reconstructing completely different model.

Gong et al [4] proposed a method of 3D face reconstruction based on TDF (Ternary Deformation Framework). It uses the relationship between feature points of eyes, nose and mouth to calculate the pose of the 2D face image by linear regression. This method simulated 3D face pose have better results. But it due to the feature points too less, and no feature points of the contour of face. So the reconstruction of 3D face image is difference between the areas of the contour of face.

From the above information, we can know that 3D face reconstruction of image-based has three main steps: (1) feature point extraction (2) model selection (3) 3D model deformation. The following will propose method in this paper.

On the face recognition Omar Faruqe and Al Mehedi Hasan [8] use PCA capture features combined with SVM classifier for face recognition. Using the face database ORL, containing a total of 400 images of 40 people. In the experiment using the LSVM (Linear SVM), PSVM (Polynomial SVM), and RBFSVM (Radial Basis Function SVM) and compare the effect of differences in experimental results show that combining PCA and SVM applied to have very good effect on the face recognition.

Li et al [9] proposed for face recognition using Kinect in different variations, this paper proposes a method of preprocessing, regardless of their initial frontal face of appearance, shape or texture, using three-dimensional point cloud obtain a new face, the algorithm also contains hole-filling and smooth noise. The method comprising the steps of preprocessing include: (1) the nose detection; (2) 3D face cut and posture correction; (3) symmetric fill; (4) smoothing. Use public database Curtin Faces, the repository has a total of 4784 images of 52 people, training variations include posture (Pose), light (Illumination), facial expressions and face obscured (sunglasses) change, it has a good performance on the face recognition, but it takes a long time algorithm of their training database containing information on a variety of changes in each person's, and to collect a large number of training data will spend too much time cost.

3. PROPOSED METHOD

We proposes a method to recognize the face, using Kinect to capture color and depth with a single frontal face, building 3D face models by OpenGL, then employee the PCA and SVM to classification. The flow chart of the proposed method is shown in Figure 1.

3.1 Kinect depth of color and perspective correction

In order to extract the information for constructing 3D face, we use Kinect to capture color and depth images, however color camera and infrared camera are not in the same position on Kinect, resulting differences view in color and depth image. Then we use OpenNi function to fix the position color camera and depth image to ensure that the relative point coordinates consistent color and depth images, it shown in Figure 2.

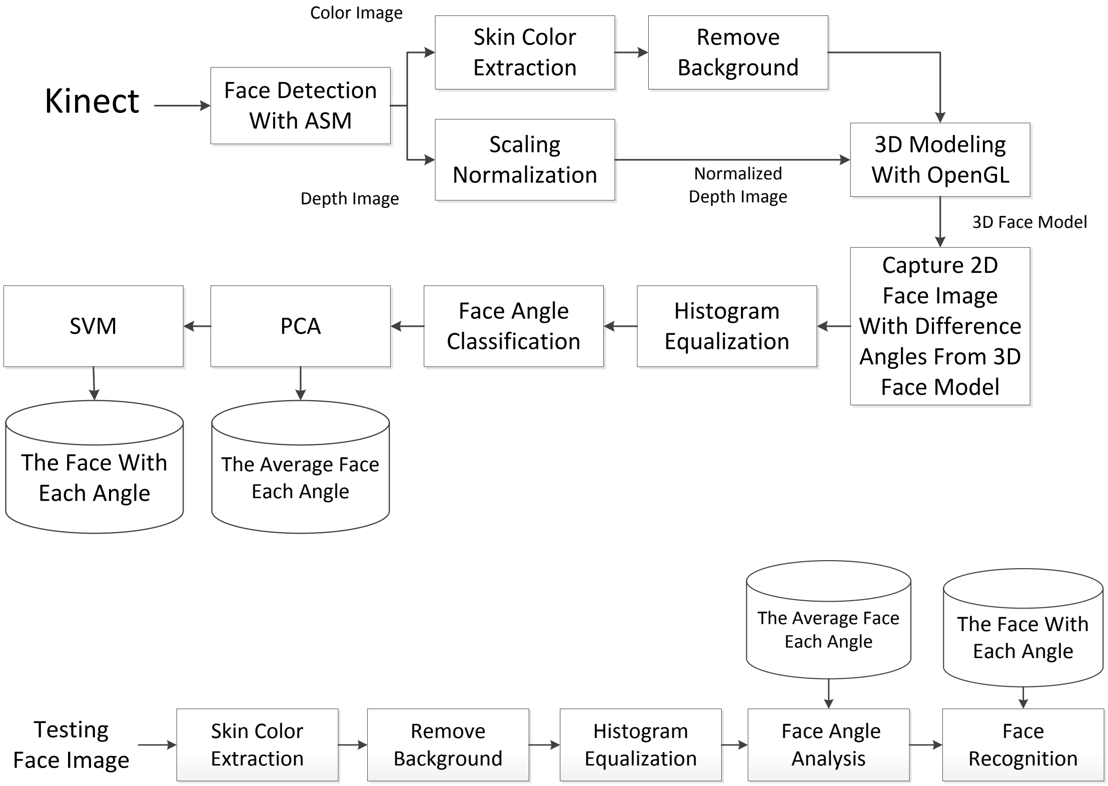
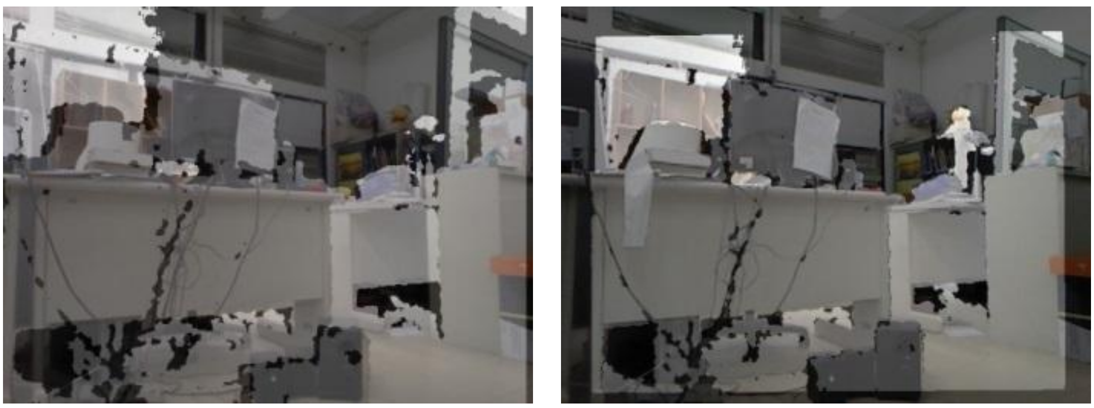


Figure 1. Flow chart of the proposed method



(a)Without aligning color and depth image (b) Aligning color and depth image

Figure 2. Aligning color and depth image

3.2 Capture facial images

We utilize ASM to extract facial components, and find the minimum depth of the images in this range as the distance between the target and the Kinect camera. This information will be pre-processing and we can create a better 3D face model. Extracting Face Information

3.3 Extracting Face Information

First, face detection is performed with ASM and these locations of face feature are extracted and also correspond to depth image. Thus, we get the face skeleton both in color and depth image. Then, we distinguish skin color with YCbCr color space and keep the face component in both images. In this step primarily to previously obtained color image segmentation of the face and the background, we will before the RGB values of the color image processing and converted into data types stored.

In order to differentiate a person's face and background, we used to detect YCbCr color space. We can find face there will be a non-skin section, for example: the eyebrows, eyes, nose, etc., we have to keep the whole face, so we eliminate the background block with mathematical morphology, and then employ OpenGL to create three-dimensional face.

3.4 Depth than the pixel length correction

In Kinect depth image, and regardless of how far from or near infrared camera, the actual length of a pixel is represented by 1mm, but our face but have before RGB color image sizes, the greater the distance between the lens more approachable face image, whereas the smaller the farther away. If not calibrate depth pixel represents the length, followed by creating a 3D face due to length ratio is wrong, and cause deformities distortion. Therefore, we must be in accordance with the distance between the camera face to adjust the depth of the actual length of the pixel at different distances represent.

In the three-dimensional space, coordinate X, Y, Z with a pixel should be the same scale. To correct the pixel length ratio, first of all we must know that in the image at different depth position, X, Y-axis presents a pixel of the actual length.
By the following experiment: Take a 30cm length object were shoot in front of Kinect distance 50cm, 100cm, 150cm to capture images, and measured the object pixel values in 640x480 images, we can get the results shown in Figure 3.

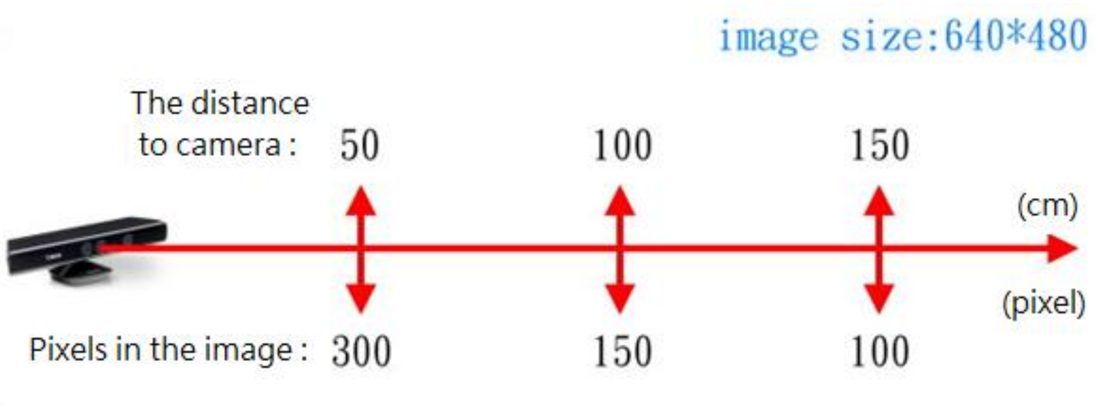
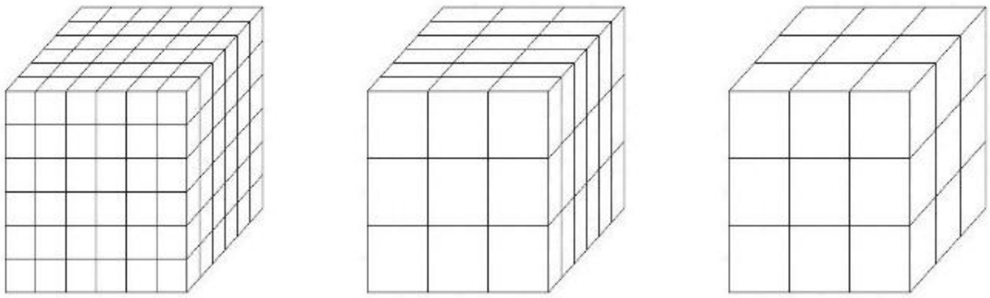


Figure 3. The distance from camera to the target and the relationship with the number of image pixels

Figure 3 shows the distance from camera to the target, the 50cm distance around 300 pixel; and 150 pixel distance 100cm;100 pixel away from 150cm, the farther away depth the fewer pixel in the image showing, the number of pixel decreased linearly.

Thus, we can learn, at a distance of 50cm under, pixels X, Y axis (RGB images) and Z axis (depth image) of the actual length ratio is 1:1:1, at this distance we do not need to adjust the pixel length ratio; when the distance is 100cm, the pixel length ratio becomes 2:2:1 (X:Y:Z), if we want to keep the ratio of the length to the same, we must use two depths in a pixel to represent pixel 100cm depth under; similarly, the length ratio became 3:3:1 when distance is 150cm, so we have to put three pixel in a depth pixel. Through Figure 4 we give an example to explain the depth pixel length ratio.



(a)X:Y:Z=1:1:1 (b) X:Y:Z=2:2:1 (c) X:Y:Z=1:1:1

Figure 4. An example of the depth of pixel length

In the three-dimensional space, pixel in X, Y, Z axis were 1:1:1, if we want to describe a cube 6x6x6, Figure 4(a). If a pixel length ratio becomes 2:2:1, and the cube becomes Figure 4(b). If we want to keep the same ratio, and that the Z-axis (depth) must to use 2 pixel represent a lattice cell, shown in Figure 4(c). Through the depth pixel normalize, we can determine at each location in Kinect.

Before correcting pixel depth length scales, first we normalize the depth values (1).

$$D_n(i,j) = D_o(\mathrm{i,j}) - D_{min} \quad (1)$$

Which D_n is new depth value after normalization, Do is original depth value, and D_{min} is minimum depth length. From the previous experiment, we know the depth value is converted to length by (2), and take the Kinect depth unit as millimeter.

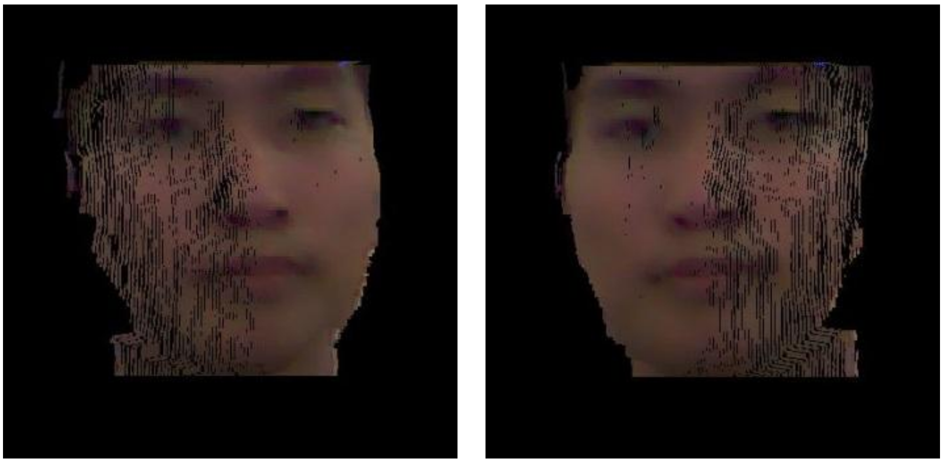
$$\mathrm{f}(D_l) = (D_l - 500) * \frac{1}{500} + 1 \quad (2)$$

Where D_l is distance between Kinect and target. Then we normalize the depth from (1) to make the depth pixel value after correction (3).

$$D_{out} = \frac{D_n}{f(D_l)} \quad (3)$$

3.5 OpenGL 3D face model create

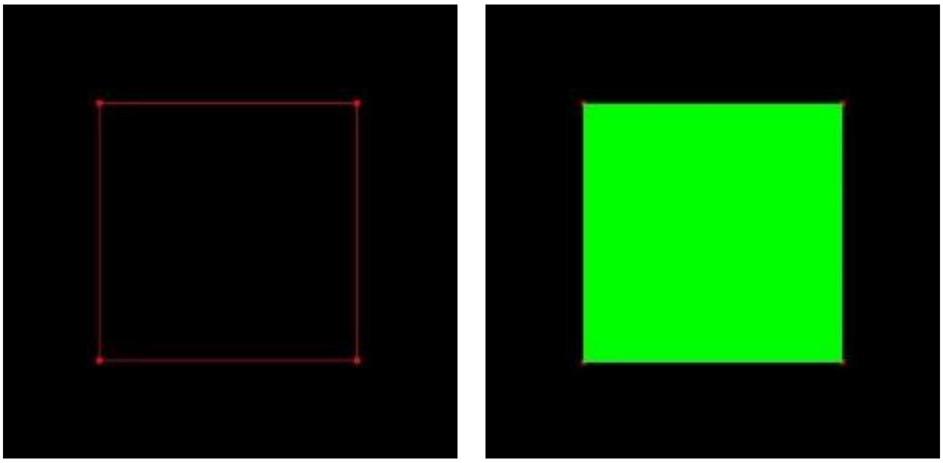
Next we will use OpenGL, and display a 3D face model. If we use the "one image pixel" as "one pixel" on 3D face , between that pixel will have some discontinuity gaps, it will generate the wrinkle face when 3D face model rotation , shown as Figure 5.



(a) Gap is generated when the left turn (b) Gap is generated when the right turn

Figure 5. When rotating 3D face the gap is generated

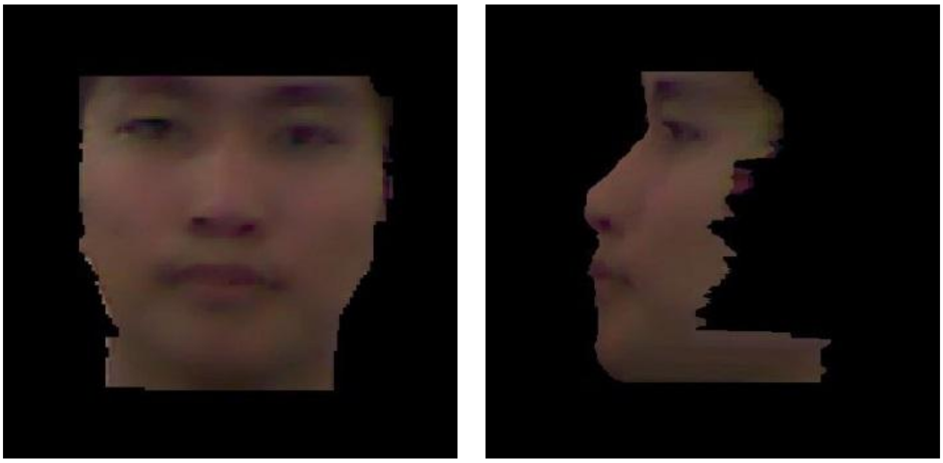
Therefore, we do not directly use a point as a single pixel to form our 3D face model, but the use of four points form a square area to make it to the plane as a pixel, Figure 6(a) shown four vertices constitutes the square area fill in color, Figure. 6(b).Thus, when the rotating 3D model will not generate wrinkle-like gap.



(a) 4 point constituted a square (b) fill color in square block

Figure 6. 4 point constituted a square as 3D face model

3D Face model we generated shown in Figure 7. After taking 2D face image with each angle from 3D face model, we use histogram equalization to reduce the effects of light.



(a) Frontal face (b) Side face

Figure 7. Proposed 3D face model

3.6 Identification section

When identification, we first determine the angle of test image, we use PCA to calculate mean face separately for each angle, there will produce an average face, then input test image and obtaining difference (4).

$$\Delta T_i = |T - M_i| \tag{4}$$

Where *T* is test image, i is the number of faces, *Mi* is the i-th mean face, Δ*Ti* is differential image. Then we compare every point between mean face and test face, taking the nearest angle to vote. Thereby determine the face angle, The mean face angles as shown in Figure 8

(a) 0° (b) 30° (c) 60° (d) 90°

Figure 8. The average face of each angle

4. EXPERIMENT

The experiment environment is indoor and depth image is captured with Microsoft Kinect. In this chapter, we show the result of 3D model we created and turn into 2D images for face recognition, analysis and discussion of the results.

4.1 Database

The database is divided into two categories, namely "3D models turn into a 2D image" and "ordinary 2D color image", the following detailed description of these two types of database, and presents some of the information database images.

Frontal face captured by Kinect color and depth images, and create 3D models, finally capture 2D face images each angle from the 3D models. Database contains 8 people, 4 types face angle, and 2 categories with face expression and face expressionless, each classification takes 20 images, a total of 1280. The part of the image data shown in Figure 9.

Figure 9. 2D face image from 3D model

4.2 Face Recognition of 3D face model turn to 2Dface image

We will turn 3D model into 2D images for training, using PCA to reduce the feature dimension of the image, and classification by SVM, then take color image to recognize. The recognition rate as shown in Table 1.

- Training data: 3D models turn to 2D image, including 8 people, face with expression have 10 images, and face with expressionless change have 10 image, a total of 160 images.
- Test data: Normal face images, including 8 people, face with expression have 10 images, and face with expressionless change have 10 image, a total of 160 images.

Table 1. Accuracy of 3D face model turn to 2Dface image

Face angle	Correct number of recognition	Accuracy rate	Average system time(s)
0°	158	98.75%	0.855
30°	140	87.50%	0.81
60°	127	79.38%	0.755
90°	118	73.75%	0.662

4.3 The training face recognition

Next, we use another set of face image to be trained, and the same test data to recognize, the recognition rate as shown in Table 2 **錯誤! 找不到參照來源。**.

- Training data: Normal face images, including 8 people, face with expression have 10 images, and face with expressionless change have 10 images, a total of 160 images.
- Test data: Normal face images, including 8 people, face with expression have 10 images, and face with expressionless change have 10 images, a total of 160 images.

Table 2. Accuracy of normal face image

Face angle	Correct number of recognition	Accuracy rate
0°	159	99.38%
30°	142	88.75%
60°	129	80.63%
90°	123	76.88%

4.4 Expression face recognition

To test the difference with expression and expressionless, we made two experiments, the training data contain only "expressionless face" and "face with expression," and identify with the same test dataset, the recognition rate as shown in Table 3 **錯誤! 找不到參照來源。** and Table 4.

Table 3. The recognition rate of face with expression

Face angle	Correct number of recognition	Accuracy rate
0°	135	84.38%
30°	127	79.38%
60°	119	74.38%
90°	103	64.38%

Table 4. The recognition rate of face with expressionless

Face angle	Correct number of recognition	Accuracy rate
0°	153	95.63%
30°	125	78.13%

60°	114	71.25%
90°	95	59.38%

4.5 Identification face angle

We use the difference value between mean face and test face to identify the test face's angle, first we compare the pixel value between mean face and test face, and vote the nearest face angle, the highest number of votes image was the angle of the target image.

- Mean face: 3D models turn to 2D image, including 8 people, face with expression have 10 images, and face with expressionless change have 10 image, divided 4 types of angles, for each angle have 160 images, in order to obtain an mean face.
- Test data: Normal face images, including 8 people, face with expression 10 images, and face with expressionless change 10 image, divided 4 types of angles, for each angle have 160 images, total of 640 images.

Table 5. Recognition rate at each angle

Correct number	Accuracy rate	Cost time(s)	Cost time(one face)
574	89.68%	32.032	0.05

Experiments show has an angle of mean face can be used for identification, but each image are to be compare with the mean face angle, so the more angles classification, the longer the time-consuming, and if angle change is too small, prone to mistake easily , detailed are shown in Table 5 and Table 6.

Table 6. Recognition rate at each angle

Face angle	Correct number of recognition	Accuracy rate
0°	160	100%
30°	141	88.13%
60°	113	70.63%
90°	160	100%

The face angle recognition error at 30 degree and 60 degree, because the analysis of 30 degree and 60 degree were similar.

5. CONCLUSION

This paper presents a face recognition method with capture single front color and depth faces, after pre-treatment to regularize image length and separate color channel, we created a 3D face model and obtained variety angles face, and save it as 2D image then use of SVM PCA dimensionality reduction and classification, the results showed that there is a good face recognition accuracy for the results. We also proposed variation facial expressions and the number of select feature vectors experiments on the recognition rate for face recognition. The proposed method can save a lot of time to collect data, and has a faster speed in 3D modeling.

Acknowledgments: This work was supported by the National Science Council under grant number MOST 103-2632-E-032 -001-MY3 and MOST 103-2410-H-032 -052.

6. REFERENCES

[1] J. Goldstein, L. D. Harmon, and A. B. Lesk, "Identification of Human Faces," Proc. IEEE, May 1971, Vol. 59, No. 5, 748-760.

[2] M. A. Turk and A. P. Pentland, "Face Recognition Using Eigenfaces," Proc. IEEE, 1991, 586-591.

[3] P. Breuer, V. Blanz, K. I. Kim, W. Kienzle and B. Scholkopf, "Automatic 3D Face Reconstruction from Single Image or Video" 8th IEEE International Conference on Automatic Face and Gesture Recognition, 2008.

[4] X. Gong, , G. Y. Wang and, L. L. Xiong, "Single 2D Image-based 3D Face Reconstruction and Its Application in Pose Estimation" Fundamenta informaticae, Volume 94 Issue 2, pp.179-195, 2009.

[5] J. Heo, , S. W. Park, and, M. Savvides,"3D Face Reconstruction from a Single 2D Face Image," Computer Vision and Pattern Recognition Workshops, 2008.

[6] W. S. Lee and N. M. Thalmann, "Fast head modeling for animation," Journal Imageand Vision Computing, vol. 18, no. 4, pp.355-364 , 2000.

[7] Y. H. Liu, , Y. Q. Su, , Z. J. Yuan and, Y. F. Zhu, "Image Based Active Model Adaptation Method for Face Reconstruction and Sketch Generation," LECT NOTESCOMPUT SC 3942, pp. 928-933, 2006.

[8] Faruqe, Md Omar, and M. Al Mehedi Hasan. "Face recognition using PCA and SVM," Anti-counterfeiting, Security, and Identification in Communication, 2009. ASID 2009. 3rd International Conference on. IEEE, 2009.

[9] Li, Billy YL, et al. "Using kinect for face recognition under varying poses, expressions, illumination and disguise," Applications of Computer Vision (WACV), 2013 IEEE Workshop on. IEEE, 2013.