PCA facial expression recognition

Article *in* Proceedings of SPIE - The International Society for Optical Engineering · November 2013

DOI: 10.1117/12.2051196

Ali Ganoun
University of Tripoli

PCA FACIAL EXPRESSION RECOGNITION

Inas H. El-Hori, Zahraa K. El-Momen, Ali Ganoun
University of Tripoli, Electrical and Electronic Engineering Department, Tripoli, Libya

ABSTRACT

This paper explores and compares techniques for automatically recognizing facial actions in sequences of images. The comparative study of Facial Expression Recognition (FER) techniques namely Principal Component's analysis (PCA) and PCA with Gabor filters (GF) is done. The objective of this research is to show that PCA with Gabor filters is superior to the first technique in terms of recognition rate. To test and evaluates their performance, experiments are performed using real database by both techniques. The universally accepted five principal emotions to be recognized are: Happy, Sad, Disgust and Angry along with Neutral. The recognition rates are obtained on all the facial expressions.

Keywords: PCA, Gabor Filter GF, Facial Expression Recognition (FER).

1. INTRODUCTION

Facial Expressions can play an important role wherever humans interact with machines. Automatic recognition of facial expressions may act as a component of natural human machine interfaces. It has been used in various real life applications such as security systems, and interactive computer simulations/designs. The most useful applications contain crowd surveillance, video content indexing, personal identification (ex. driver's license), entrance security, etc. We believe recognition of human facial expression by computer is a key to develop such technology.

Facial Expression is one of the most powerful, nature, and immediate means for human beings to communicate their emotions and intentions. Due to its potential applications, automatic FER has attracted much attention since two decades. Though much progress has been made, recognizing facial expression with a high accuracy remains to be difficult due to the complexity and variety of facial expressions.

Michael et. al. [1] proposed a method for automatically classifying facial images based on labeled elastic graph matching, a 2D Gabor wavelet representation, and linear discriminant analysis. Results of tests with three image sets are presented for the classification of sex, "race", and expression. A visual interpretation of the discriminant vectors is provided. Turk et. al. [2] seeks to implement a system capable of efficient, simple, and accurate face recognition in a constrained environment. Classification is performed using a linear combination of characteristic features (eigenfaces).

2. METHODOLOGY

In the language of information theory, we want to extract the relevant information in a face image, encode it as efficiently as possible, and compare one face encoding with a database of models encoded similarly. A simple approach to extracting the information contained in an image of a face is to somehow capture the variation in a collection of face images, independent of any judgment of features, and use this information to encode and compare individual face images.

A practical FER system is shown in Fig. 1; the Recognition process begins by first acquiring the image using a camera. The image needs to be preprocess (Image Preprocessing) such that environmental and other variations in different images are minimized and detecting the face.

Then the features must be extracted (Feature Extraction) from image using GF technique which uses local characteristics of the face to derive features from the face image, then applying PCA (Image Processing) which is a common technique for finding patterns in data of high dimension. Next is (classification) by making calculations of distances then extracting the resulted expressions (Neutral, Happy, Anger, Sad, and Disgust).

Sixth International Conference on Machine Vision (ICMV 2013), edited by Antanas Verikas,
Branislav Vuksanovic, Jianhong Zhou, Proc. of SPIE Vol. 9067, 906712 · © 2013
SPIE · CCC code: 0277-786X/13/$18 · doi: 10.1117/12.2051196

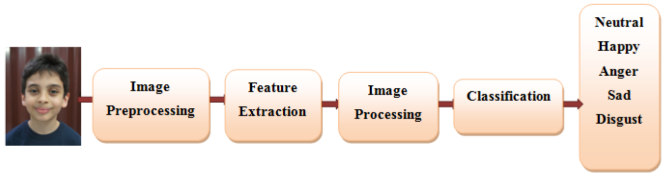


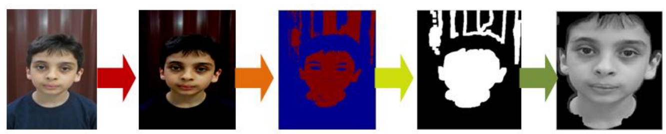
Figure 1. Facial expression recognition system overview.

2.1 Image preprocessing

This process begins by first acquiring the image, second the acquired image needs to be adjusted by using lighting compensation, third detect the skin color, next is removing noises from the image, and finally localize skin color boundaries as shown in Fig 2.

Environmental and other variations in different images are minimized. Usually, the image preprocessing step comprises of operations like image scaling, image brightness and contrast adjustment and other image enhancement operations.

a



b

Figure 2. a) Flowchart of the image preprocessing. b) Image corresponding for each stage at preprocessing.

2.2 Feature Extraction (Gabor Filters GF)

GF have proven to be a powerful tool for facial feature extraction, this method proposed to use the whole face image for extraction of Gabor-based features. A set of 40 filters are employed to derive a high-dimensional feature vector.

A. *GF bank construction*

In the spatial domain, a 2D Gabor filter is a Gaussian kernel function modulated by a complex plane wave [3]:

$$\varphi_{\pi\,(f,\theta,\gamma,\eta)} = \frac{f^2}{\pi\gamma\eta}\, e^{-\left(\frac{f^2}{\gamma^2}x_t^2+\frac{f^2}{\eta^2}y_t^2\right)} e^{j2\pi f x_t} \qquad (1)$$

Where; x, y = pixel coordinates, f = frequency of the complex sinusoid, θ = orientation of the wavelet, γ = spatial width of the wavelet along the sinusoidal plane wave and η = wavelet's spatial width perpendicular to the wave.

To obtain an appropriate feature vector from an image, a bank of 40 filters (5 scales and 8 orientations) is constructed using the following parameters for optimal performance with 128x128 pixel face images [3][4]:

$$\varphi_{g,h}(x,y) = \varphi_{\pi(f_g,\theta_h,\gamma,\eta)} \qquad (2)$$

Where; $f_g = f_{max}/(\sqrt{2})^g$, $\theta_h = \frac{h}{8}\pi$, $\gamma = \eta = \sqrt{2}$, $f_{max} = 0.25$

B. *Feature extraction using GF*

The Gabor feature representation of a grey-scale face image I(x, y)$\in R^{a\times b}$, where; a and b are image dimensions (in pixels). Face image I(x, y) with the GF $\varphi_{g,h}$(x, y) of scale g and orientation h:

$$O_{g,h}(x,y) = I(x,y) * \varphi_{g,h}(x,y) \qquad (3)$$

The Gabor feature vector x is obtained by down sampling the magnitudes of $O_{g,h}(x,y)$ for all scales and orientations by a factor ρ, turning them into row vectors $O_{g,h}^{\rho}$ and finally concatenating them into the feature vector x [3][4]:

$$x = (O_{0,0}^T\ O_{0,1}^T \ldots\ldots\ldots\ldots O_{4,7}^T)^T \in R^N \qquad (4)$$

Where; N = 40ab/ρ.

2.3 Image Processing (PCA)

PCA reduces a complex data set to a lower dimension and builds a face space which better describes the face images; the basic vectors of this face space are called the principal components.The face images are decomposed into a small set of characteristic feature images called “Eigenfaces” (which contain the common features in a face) which are extracted from the original training set of images PCA. The Eigenface approach involves the following initialization operations:

1. Mean face of the filtered images is computed and subtracted from all the images.

$$A = [V1 \; V2 \; V3 \; \dots \; VW] \qquad (5)$$

2. The Eigenfaces (eigenvectors) from the training set are calculated and only K images that correspond to the highest Eigen values define the face space.

$$\mathrm{C}u_i = \lambda_i u_i \qquad (6)$$

Where: C = covariance matrix, λ_i= eigenvalues.

$$\text{Eigenface:} \quad u_i = [u_1 \;\; u_2 \;\; u_3 \dots u_k] \qquad (7)$$

3. By projecting the face images onto the face space, the corresponding distribution in K-dimensional weight space (principal components) for each individual image is found. [2][5][6].

2.4 Classification

1. The Euclidian distance of a projected test image from all the projected train database images are calculated and the minimum value is chosen in order to find out the train image which is most similar to the test image, the test image is assumed to fall in the same class that the closest train image belongs to [2].

$$\epsilon_k^2 = \|(X - X_k)\| \qquad (8)$$

Where: X= the projection of a test image, X_k = train database with face class k (Happy Sad ... etc).

2. In order to determine the intensity of a particular expression, its Euclidian distance from the mean of the projected neutral images is calculated. The more the distance - according to the assumption - the far it is from the neutral expression (Fig. 3). As a result, it can be recognized as a stronger expression. [2].

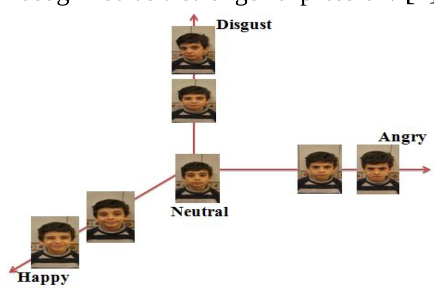


Figure 3. The strength of the expressions.

3. RESULTS AND COMAPRISON

3.1 Database Preparation

A database of real time facial expression images was collected. In our implementation, there are 4 sequences of databases; sequence (1, 2, 3 and 4), each sequence represent a person and have a set of test and train images. Each sequence has been rated on five basic facial expressions. All images are resized to a uniform dimension of 280×180. Table 1; shows the number of Test, Train and size of images per sequence, Fig. 4, contains images from our Database of five different expressions for sequence (4). The testes were done on both techniques; PCA with GF and PCA only. We got different recognition rates for all five principal emotions of all 4 different sequences.

Table 1. Database of test sequences.

Name	# of train images	# of test images	Image size (pixels)
Sequence (1)	**50**	**31**	**600 x 800**
Sequence (2)	**50**	**25**	**480 x 640**
Sequence (3)	**59**	**34**	**640 x 480**
Sequence (4)	**43**	**30**	**1872 x 3344**


Figure 4. Example of the five expressions of sequence (4) from the Database

3.2 Results Obtained on Database using PCA with GF

Table 2 shows the Recognition Rates for every single Facial Expression of test images of all sequences, Happy has the highest recognition rate because of the lip corners are pulled obliquely, next best rate is Anger that could be caused by the eye brows lowered and drawn together. The worst rate has the expression Sad and that couldn't be recognized because of the similarity to the Disgust expression which made the confusion to recognize.

The calculated Euclidean Distance from Neutral ranges between 33 & 82, Fig. 5 shows a plot of the calculated distance from Neutral of sequence (4); every Test image has a different Euclidean Distance, the more the distance the far it is from the Neutral expression, the highest distance from Neutral expression is test image (6) which results a strong Anger expression with distance 82. The smallest distance is test image (2) which results a facial expression of Neutral with distance 33.

Table 2. Comparison of Recognition Rates using PCA with GF.

Expression	Sequence(1)	Sequence(2)	Sequence(3)	Sequence(4)	Average
Happy	**90.9%**	**100%**	**100%**	**60%**	**87.7%**
Sad	**0%**	**75%**	**100%**	**40%**	**53.7%**
Anger	**85.7%**	**60%**	**83.3%**	**75%**	**76%**
Disgust	**66.6%**	**100%**	**66.6%**	**50%**	**70.8%**
Neutral	**100%**	**60%**	**42.8%**	**50%**	**63%**

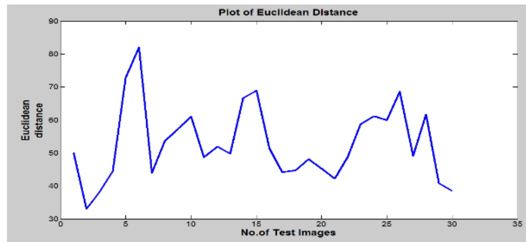


Figure 5. Plot of Euclidean distance of sequence (4) database.

3.3 Results Obtained on Real Database using PCA

Table 3 shows the recognition rates of all sequences using PCA method. Overall the rates are low; Neutral has the highest rate, which could be caused by the face features are relaxed, that caused better recognition rate.

The calculated Euclidean distance from Neutral ranges between 2892 & 8676, the distance from Neutral which is very supreme in compare with the distances obtained using PCA with GF. Fig. 6 shows the plot of Euclidean distance of sequence (4); the highest distance from Neutral expression is test image (6) which results a strong Anger facial expression with distance 8676. The smallest distance is test image (21) which results a facial expression of Neutral with distance 2892.

Table 3. Recognition Rates using PCA.

Expression	Sequence (1)	Sequence (2)	Sequence (3)	Sequence (4)	Average
Happy	**63.6%**	**60%**	**55%**	**50%**	**45.9%**
Sad	**0%**	**50%**	**0%**	**60%**	**27.5%**
Anger	**57%**	**60%**	**83%**	**25%**	**56%**
Disgust	**77.7%**	**100%**	**33%**	**33%**	**61%**
Neutral	**100%**	**80%**	**57%**	**25%**	**65.5%**

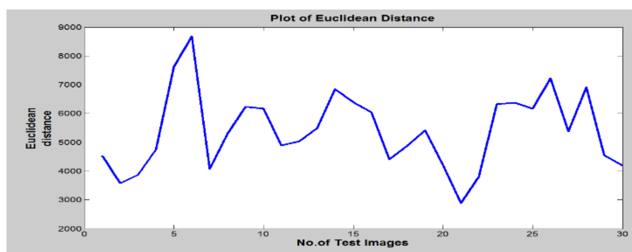


Figure 6. Plot of Euclidean distance calculated of Test Images of sequence (4) database.

3.4 Comparison of PCA & PCA with GF

By comparing PCA with GF with PCA we obtained better results for the first as shown in Fig. 7, this Bar Chart which represents the recognition rates for every sequence, using PCA with GF has the higher rate. PCA with GF improved in sequences (1) & (2) to exceed 80%, in comparison with the result obtained with sequence (4) which has the lowest rate and that was caused by the back ground which wasn't homogeneous, but still using PCA with GF method improved it from 36.6% to 53.3%. In the other hand sequence (3) had a major improvement after using PCA with GF from 47% to 79%. Table 4 representing the execution time for both methods PCA and PCA with GF. As we can see the time using PCA ranges between 286 to 387 seconds, and the time using PCA with GF ranges between 347 to 451 seconds, which is higher than the first method and that was caused by creating the filter bank and filtering every single image.

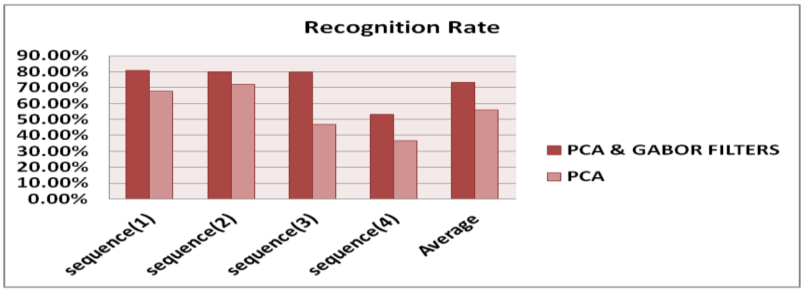


Figure 7. Comparison of FER techniques.

Table 4. Comparison in time execution.

	Time using PCA (second)	Time using PCA with Gabor filters(second)	Time difference (second)
Sequence(1)	**308**	**380**	**72**
Sequence(2)	**286**	**347**	**61**
Sequence(3)	**351**	**430**	**79**
Sequence(4)	**387**	**451**	**64**

4. CONCLUSIONS

In this research a comparative study of PCA and PCA with Gabor filter was done. Four different sequences were used as a test sequences, two males and two females.

The experimental results indicate that the method is effective for PCA with GF (average recognition rate 73.35%), outperforms under different facial expression background, and illumination condition. It is a good recognition performance in compare with using PCA only (average recognition rate 55.85%). There are still some different recognition rates between the 4 sequences that were tested, the environmental background and face hiding objects (glasses scarf … etc) were the causes for the disparate recognition rates.

4.1Future Work

We observed that there are still some setbacks in our research were efficiency needs to be considered. As we know that our methodology had higher expression detection rates but the algorithm still needs to be more optimized so as to reduce the computational time and sometimes it is inevitable to trade of between accuracy and speed. The final step required in future is that gathering a very large number of data is really important in this field of research and it can be always possible to use more. Gathering more emotional words and generating more facial expressions can give more accurate results and drive us in different conclusions. Generating more facial expressions for each emotion is also very important as there is not only one way to represent an emotion. Find more variations of the emotional facial expressions can be a further step in this research.

REFERENCES

[1] M Lyons, J. Budynek and S. Akamatsu, "Automatic Classification of Single Facial Images", IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 21, No. 12, (1999).

[2] Matthew Turk and Alex Pentland, "Eigenfaces for Recognition"The Media Lab. Massachusetts Institute of Technology,Vol.3, No.1,pp. 71-86, (1991).

[3] Vitomir ŠTRUC, Nikola PAVEŠI´C, "The Complete Gabor-Fisher Classifier for Robust Face Recognition", EURASIP Advances in Signal Processing, Vol. 2010, doi:10.1155/2010/847680, (2010).

[4] Vitomir ŠTRUC, Nikola PAVEŠI´C, "Gabor-Based Kernel Partial-Least-Squares Discrimination Features for Face Recognition", Informatica (Vilnius), Vol.20, No.1, pp. 115–138, (2009).

[5] Gregory Shakhnarovich, Baback Moghaddam, "Face Recognition in Subspaces", MITSUBISHI ELECTRIC RESEARCH LABORATORIES, TR2004-041, (2004).

[6] Aron Yu, Catherine Elder, Jeff Yeh, Norman Pai, "Facial Recognition using Eigenfaces: Obtaining Eigenfaces", The Connexions Project, Version 1.6, Connexions module: m33183, (2009).