



Utilization of 3-D Models for Enhancing Facial Recognition over Various Angles

By Jovan Toroman

Abstract- Algorithms for facial recognition PCA (Principal component analysis) encounter a dramatic performance decline when analyzing photos taken from various angles. As part of this research, a 3-D model was utilized for the sake of improving abovementioned performance. This model enables us to convert an image of somebody's face, which was taken from a specific angle to another face. Model was tested using biometric database created on Belgrade University. Picture rotation according to model was performed before comparing with the specimen pictures from the database. Results of this study have shown accuracy in regards to biometric verification, where identification percentage has been drastically improved.

Keywords: *biometrics, face recognition, 3D graphics, eigen face.*

GJCST-D Classification: *I.5.2*



Strictly as per the compliance and regulations of:



Utilization of 3-D Models for Enhancing Facial Recognition over Various Angles

Jovan Toroman

Abstract- Algorithms for facial recognition PCA (Principal component analysis) encounter a dramatic performance decline when analyzing photos taken from various angles. As part of this research, a 3-D model was utilized for the sake of improving abovementioned performance. This model enables us to convert an image of somebody's face, which was taken from a specific angle to another face. Model was tested using biometric database created on Belgrade University. Picture rotation according to model was performed before comparing with the specimen pictures from the database. Results of this study have shown accuracy in regards to biometric verification, where identification percentage has been drastically improved.

Keywords: *biometrics, face recognition, 3D graphics, eigen face.*

I. INTRODUCTION

Face is a specific characteristic of every human being. When interacting with other people, we direct attention to their face. With the help of facial features, we recognize our acquaintances and distinguish them from unknown people. The human brain is pretty good at recognizing faces and is able to make the right decision even in difficult conditions such as poor lighting, aging, wearing glasses, or changing haircuts. Automatically recognizing a person with a computer is an area that is still in development, with numerous challenges that remain to be solved. In laboratory conditions, without external disturbances, modern systems for facial identification have good performance. However, problems such as changes in lighting, viewing angle, and occlusion have a significant impact on the performance of these biometric systems.

Since cameras are becoming cheaper and more affordable, biometric detection and face detection systems have numerous applications in different areas. They are used to identify faces in photos within social networks, detect fraud in voting at elections, control access to different systems, as well as in surveillance systems. In this paper the problem of changing the position of the head on the picture of the face will be considered. This problem is particularly obvious when using comparison algorithms with two-dimensional facial images. In the case where the position of the face in the image in the base differs in relation to the position in the image obtained by the sensor, the algorithm

becomes less accurate. The proposed solution presented in this paper is based on a three-dimensional model. By using a semi-profile image, the program will generate an en face of an image, which will be compared with the real en face images using the Eigenface method.

II. DETECTION AND FACIAL RECOGNITION

If it is necessary to create surveillance system that would monitor the camera, for example, the arrival of employees to work, it is necessary to solve two closely related problems: face detection and facial recognition. Face detection involves finding the part of the image on which the face is located. The most extensive method for detection is Viola-Jones [1], which works by principle of the difference in the width of the rectangle with the pixels in different shades of gray. This method returns the rectangular area on which the face is located and works best for an end face of an image.

Face recognition includes more than one task:

Identification: For the given image, you need to find the person in the database which the image belongs to, or conclude that the person is not at the base.

Verification: For a given pair <image, person> to confirm whether this image belongs to a person or not.

Segmentation: Divide the number of images into groups by person.

All these tasks are essentially reduced to one: discover if two photographs belong to the same person.

III. FACIAL RECOGNITION ALGORITHMS

The first recognition algorithms were feature-based, and their principle of operation was to detect all the key elements of the face, such as eyes, eyebrows, bumps, nose and others, in order to measure them on the basis of their dimensions, angles observation, and distances between each other and calculate a set of values that characterize a given face. Usually, about 40 values were considered and, depending on the specific algorithm that used them, i.e. its characteristics and performance, the performed identification would have a certain degree of accuracy. Work [2] belongs to this category. However, this approach has proven to be insufficiently precise, due to the difficulty of accurately determining the characteristics and neglecting of the information contained in the texture of the face itself.

The second groups of algorithms that were later created, and who still hold the primacy in this area, are based on the observation of an entity as a whole. The essence of the way they work is reduced to the existence of a template in the database, and to the comparison of the person who is the subject of identification with the given templates. To date, some algorithms based on this approach have been developed, some of which are very popular and often used.

The first algorithm from this group is the so-called Eigenface algorithm [3]. This algorithm is based on the Principal Component Analysis method. An analysis of the main components is a statistical method for reducing a number of variables that are observed to a smaller number of new variables that we call the main components. The idea is to find characteristic vectors (eigenfaces) of covariance matrix of the sets of pictures of the face we observe, treating each image as a vector in a multi-dimensional space. These characteristic facial vectors can be seen as a set of characteristics that represent the variation between different facial images in the base. Every face in the database is stored as a vector, a corresponding linear combination of these characteristic facial vectors. For comparison of vectors obtained from different faces we can use different methods, such as Euclidean or Mahalanobis distance.



Figure 1: Eigen faces

The implementation of the eigenface algorithm is available in the OpenCV library (Open Source Computer Vision) [4]. OpenCV is an open source library that contains functions for implementing computer vision activities. It was written in programming languages C and C++, and has support for launching on Windows, Linux and Mac OS X operating systems. OpenCV was used to test the effect of using the proposed model on the accuracy of recognition.

IV. EXISTING WORK IN THIS FIELD

As for the use of three-dimensional models in the field of face recognition, the most famous work is that of Blanz and Vetter, who used a laser 3D scanner to create a database with 200 people [5]. Then they manually defined the characteristic points for the eyes, nose and mouth. After that, the recognition was done based on the interpolation of the 3D model from the base. Other work was mostly based on face recognition with 3D cameras. In this way, very good results are obtained, but the disadvantage of this approach is that it requires expensive equipment and cannot be applied to ordinary images. Some of the papers in this field are [6] and [7].

V. SOFTWARE DESCRIPTION

The image rotation program is written in the C++ programming language. The OpenGL API allows the use of hardware acceleration when plotting the model through direct communication with the graphics card.

A free head model was downloaded from the Internet [8]. The model is represented by a series of points in the space, a series of triangles and a series of texture coordinates. Each point in the space is arranged by the triplet (x, y, z), where x, y and z are the coordinates of the point along the corresponding axes. The triangles consist of three points in space. A set of texture coordinates is of the same size as a series of points, with each point corresponding to one coordinate of the texture. The texture coordinates are arranged in pairs (u, v) where u and v are textures along x and y-axes, respectively.

This model is placed on the picture by placing the eye and nose points on the model and the face from the image. This is achieved by scaling and translating the model. Also, it is important to rotate the model so that the angle of the model is equal to the angle under which the face of the image stands. Rotation is performed using the transformation matrix, as shown in Figure 2. If we represent a point as a matrix whose rows are values x, y and z, by multiplying this matrix with the transformation matrix we obtain the coordinates of the point.

$$\begin{aligned}
 M_x(\theta) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix} && \text{Rotacija oko X ose} \\
 M_y(\theta) &= \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} && \text{Rotacija oko Y ose} \\
 M_z(\theta) &= \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} && \text{Rotacija oko Z ose}
 \end{aligned}$$

Figure 2: Transformation matrices for rotation

Then the picture is "glued" to the model. This will be done by changing the texture coordinates of the model so that each pair (u, v) takes the values of the transformed point (x', y', z') so that $u = x$ and $v = y$.

When gluing an image to a model, the problem of hidden surfaces is created. This problem is easily shown on an example of a cube (Figure 3): the blue, red and green sides are visible and can be filled with texture. The other three sides are concealed. So in the face model as well, there are surfaces that are hidden and they are fulfilled using the principle of the mirror image. In order to see which surfaces are hidden, we will check the order of the triangle points: if they are counterclockwise, then it's a hidden surface. The program memorizes for each triangle whether it is hidden or not during gluing of the texture. Also, for each point of the model we will determine the point that is symmetric to it in relation to the x axis. The symmetric point is chosen so that the sum $(sx + x) + (sy - y) + (sz - z)$ is minimal.

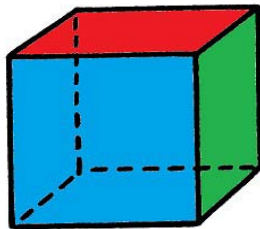


Figure 3: Hidden surfaces problem

When the image is glued to the model, the model rotates back into the flat position using transformations for the negative value of the angle for which we rotated during installation. Then using the OpenGL function, we plot the model on the screen. For each triangle, we will pass three points in the space and the corresponding texture coordinates, whereby the coordinates of the texture of the symmetric point will be passed to the hidden triangles. The next step is to copy the graphics card buffer into memory and memory to the hard disk in bitmap format. The entire transformation process is shown in Figure 4.

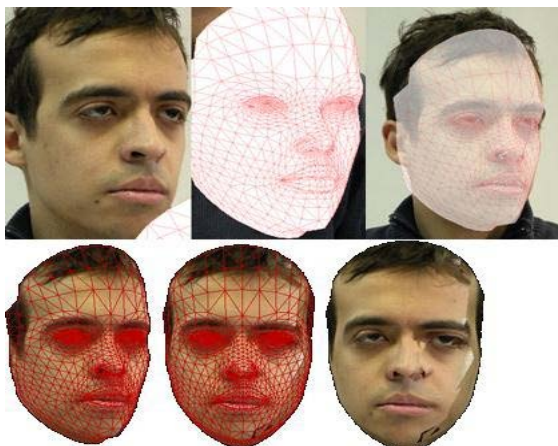


Figure 4: Transformation process

In the processed images, the border between the areas that were hidden and other surfaces are visible. This limit would not be visible if the light came frontally, or if the lighting was normalized.



Figure 5: Original "en face" picture

The resulting images will be further aligned using the funnel [9], which will improve the results and allow more valid comparison of the results.

VI. RESULTS

All tests were conducted on a database with 34 persons from the Faculty of Organizational Sciences [10]. The database contains images of persons under different angles in similar lighting conditions, photographed twice in six months. For the creation of the database, the en face images from the first period were used. Then, three tests with images from the second period were performed: en face images, semi-profile images and program-generated en face images by processing semi-profile images (from now on referred to as "returned"). The accuracy of the verification process can be seen using ROC (Receiver Operating Characteristics) curve in Figure 6.

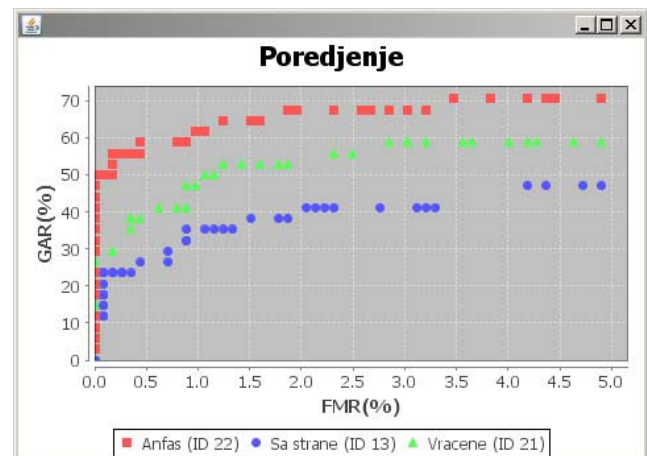


Figure 6: Verification - relation of GAR and FMR

The identification process gave the following results: En face images showed the best performance, which was expected. Twenty-two persons were successfully identified (65%). Semi-profile images showed a significant decline in the performance of the algorithm - only 13 people were identified (38%). Returned images gave better results than semi-profile images, but worse than en face images. As many as 21 people (62%) were identified, which is only one person less than the en face image.

VII. CONCLUSION

Rotation of images using a three-dimensional model in the initial phase of the study showed good results. Research conducted on a biometric database collected at the Faculty of Organizational Sciences has shown that this approach contributes to the precision of the biometric system. These results encourage us to continue improving the proposed solution.

For now, the process of attaching the model on the picture is done manually, which means that it is necessary for a person to be present in order for the system to work. In the following versions of the program, the program should independently determine the position and angle of the head so that the system becomes automated. This automation would enable more intensive testing on larger databases.

The model of the face to which texture is mapped is rigid and unchanging, which would be appropriate if all people had the same face shape, which is obviously not the case. It should be possible to modify the model each time a person is mapped in order for the model to fit as good as possible to the person to be mapped. Change parameters could be used to recognize the face in addition to or instead of a PCA algorithm.

Also, it would be possible to extend the application's performance in terms of solving the problem of lightening normalization.

REFERENCES RÉFÉRENCES REFERENCIAS

1. P. Viola, M. Jones "Rapid object detection using a boosted cascade of simple features ", Computer Vision and Pattern Recognition, pp. 511–518, 2001.
2. R. Brunelli, T. Poggio "Face recognition: Features versus templates", IEEE Trans. Pattern Anal. Mach. Intell. 15(10), pp. 1042–1052, 1993.
3. M. Turk, A. Pentland, "Eigenfaces for recognition" Journal of Cognitive Neuroscience 3(1), pp. 71–86, 1991.
4. G. Bradski, A. Kaehler "Learning Opencv, 1st Edition (First ed.)", O'Reilly Media, Inc., 2008.
5. V. Blanz, T. Vetter, "Face recognition based on fitting a 3D morphable model," Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.25, no.9, pp.1063,1074, Sept. 2003.
6. A. Bronstein, M. Bronstein, R. Kimmel, "Three-Dimensional Face Recognition", International Journal of Computer Vision, vol. 34, 2005.
7. Kakadiaris, G. Passalis, G. Toderici, M. Murtuza, Y. Lu, N. Karampatziakis, T. Theoharis "Three-dimensional face recognition in the presence of facial expressions: An annotated deformable model approach", IEEE Trans. Pattern Anal. Mach. Intell. 29(4), pp. 640–649, 2007.
8. Female head by Bitmap world <http://www.turbosquid.com/3d-models/polygonal-female-head-3ds/265937>, last visited on 10th of June, 2013.
9. G. B. Huang, V. Jain, E. Learned-Miller: Unsupervised Joint Alignment of Complex Images, 2007.
10. Milenković, V. Pantović, D. Starčević, M. Minović, "A multimodal biometrics system implemented using open source technology," Telecommunications Forum (TELFOR), 2011 19th pp.1352 - 1355, Nov. 2011.