The University of Calgary
Department of Electrical and Computer Engineering

ENCM 509 - Fundamentals of Biometric Systems Design

Laboratory Experiment # 6

3D face modeling and recognition

1 Introduction

The purpose of this laboratory exercise is to investigate biometric data modeling (synthetic biometrics) and its possible uses.

Collecting images and testing the facial expression algorithm is a time-consuming problem: it is a lengthy and privacy-issue related procedure. In addition, it is hard or impossible to imitate real-life noises or distortions such as backgrounds, facial accessories and head rotation. In this case, so-called analysis-through-synthesis can be applied, where the testing data can be created using a model of the face.

The Biometric Technologies Laboratory at the UofC has a package for 3D facial modeling called FaceGen from Singular Inversions. Version 3.3 of this package is used in this exercise and is available on drive `N:\ENCM\509\SingularInversions\FaceGenModeller3.3`.

One of the functions of this package, called PhotoFit, can create a 3D model of a head given one still frontal image, or, better yet, two more profile images. You can use your pictures taken prior to the lab.

2 3D face model

The term "face synthesis" indicates a class of problems aimed at synthesizing a model of facial appearance. The generation of faces and facial expressions is the manipulation of topological primitives with special rules. Physical modeling is based on descriptions of the facial muscles, skin, and their relationships by mathematical equations. Given a muscle model and emotions, the latter can be encoded or incorporated to produce a synthetic face. It allows for creating a databases of facial images to be used for testing the facial recognition algorithms and systems.

Analysis of facial appearance information as well as behavioral information such as facial expressions can contribute to facial biometric analysis. Another example is speech-to-lips synchronization, lip reading etc. These techniques utilize topological and physical models in the interpretation of facial information and the design of synthetic facial expressions.

Systems for face and facial expression synthesis provide the possibility to design faces in 2D and 3D, to model race, gender and adult age group, control shapes, coloring, lighting and face rotation etc. In this laboratory exercise the software package called FaceGen is used. The 3D model used in the package is a combination of 3D mesh model of a human head, and a texture model of the skin. A face appearance can be morphed with respect to various parameters: expression morphing (smile (closed and open), anger, fear, disgust, sad, surprise), age morphing (shape and texture modifications), race and gender morphing, asymmetry etc.

3 Exercise

3.1 Creating your face model

Open FaceGenModeller 3.3 and run the application. The application displays some average 3D face. The next instruction shows how to create a 3D model of your face.

3.2 Exercise 1: 3D model creation

- Choose the most right panel "PhotoFit" and click on the bottom key "Next".
- In the new window, choose "Load" for Frontal Image, and browse to the directory where your face is stored. Choose one neutral facial expression, and possibly front and/or rotated images.
- Proceed to the next prompted step "Start Now" (you may also check mark the option "Preserve facial hair in detailed picture" if not marked yet). It may take several minutes to create the model.
- You can change the background color (for example, to green) in "View" and "Background Color".
- To model hair, you can choose "Texture overlay", the panel under the picture, and in the new window move "Short black hair" from "Available overlays" to "Current overlays" using the button `>>>`.
- Save the image with extension `.fg` (to be able to open it again in FaceGen) using "File" then "Save", as well as with extension `.bmp` using "File" then "Save as" (to be able to use for further steps such as face recognition).

Thus, you save the primary face model (using green background) with neutral facial expression.

3.3 Exercise 2: 3D model manipulation

Explore some option in the FaceGen menu for your neutral facial expression (you can load the saved file `.fg` of your 3D face model from Exercise 1):

- Choose panel "Generate", - you can change age (20 – 60 years old), gender, race etc.
- Choose panel "View", - you can change background color and light options (Light Source) etc.
- Choose panel "Camera", - you can change zoom and distance ratio
- Choose panel "Shape", - you can change some of the facial shapes expressions
- Choose panel "Texture", - you can add some facial accessories (though the colors may not be quire realistic)
- Choose panel "Genetic", - you can generate 8 random faces similar to yours
- Choose panel "Morph", - you can change some of the facial expressions.
- In addition, you are able to rotate the 3D view using the mouse while pressing the left button and holding it on the face image.

Save the following results:

- Among these options, use some, for example, "Shape" and "Morph" to change your neutral facial expression as close as possible to your chosen expression (for example, "Mouth open"). This will be a "Synthetic expression". Save 3 different face expressions; they shall correspond to any of the 3 expressions of your face taken prior to the lab. We will compare the results of face recognition for both real and synthetic face later.
- Use the rotation and rotate face $15^0, 30^0, 45^0$ (approximately) degree and save images as `.fg` and `.bmp` files., - we will use them to verify if the algorithm is still be able to recognize the expression on the rotated face (head rotation is the real problem in the automatic face recognition).
- Use the options from "View" to change background to some pattern; save images as `.fg` and `.bmp` files.
- Use the options from "View" to change lighting, for example, light source coming from the left or right or top; save images as `.fg` and `.bmp` files.

3.4 Exercise 3: Face recognition for the real face

Use your own images and/or your partner's to create a small database of your images with and without facial expression. Remember, image size must be normalized prior to processing. You can also perform image filtering of intensity normalization of your choice.

In this exercise, we will use the file FaceRec.m on N: drive $\backslash ENCM \backslash 509 \backslash lab6$.

The input data shall be either bmp or jpg image. The number of images, M, is set up in the code (it is equal to 9, you can change it to smaller number and use the remaining face for testing; this number can be large for larger databases).

It is important to have the images normalized: their sizes must be equal (in number of pixels), otherwise, the program will report errors. You can use any graphical editor, for example, Paint, to do so.

The program finds the mean face and perform calculations of differences between the mean image and every image, finding the covariance matrix and so-called eigenvectors.

After the "training", the recognition can be done. A new image (could be of one of the persons in the database but rotated or morphed) must have the same size as the training images. The default path of your input image is lab directory on N drive (you can change it by changing line 30). Then, the eigenvector of the new image is calculated, and a difference (Euclidean distance) between that and the training images. The last step is analyzing this difference and making a decision about a match (which database facial image matches the new one). This is $1 : N$ comparison, or identification. It is not automated in this lab, and your exercise is to create the automated identification procedure. For that, you need to define some thresholds for the matching procedure.

The face (template) from the database is a match if it is the closest one to the input face (live template), based on the Euclidean distance. This is related to two thresholds.

One of them has to distinguish between a known face and unknown face: the input face belongs to a class (of the known face), if the Euclidean distance is below an establish threshold. Then, the face image is considered to be a known face. The second threshold shall divide faces from non-face objects. If the distance is above the first threshold, but below the second threshold, the image can be determined as an unknown face. If the input image is above these thresholds, the image is not a face.

The two thresholds must be found. Statistics can be used to determine the first and the second threshold. To find first threshold, all the maximum and minimum Euclidean distances when the input face belongs to the database, must be gathered. Given 9 (synthetic) faces in the database, FaceRec.m must be run 9 times, using the 9 faces in the dataset as the input face and record the maximum and minimum Euclidean distance. Then, by calculating the mean and standard deviation of all the values, the first threshold can be set at the maximum mean. To find the second threshold, use unknown (any synthetic) faces as the input faces and gather all the max

and min Euclidean distance. Choose the maximum mean as the second threshold.

Note that for large databases, a ranking model of matching can be applied, and the top few minimal distances are ranked as possible candidates.

In this exercise,

- Extend the script (m-file), that would help you to define the thresholds and provide the decision output.
- Perform facial recognition and find the true and false match rates and rejection rates.
- Draw conclusions and make suggestions on the procedure improvement.

3.5 Exercise 4: 3D modeling for testing the facial recognition algorithm on synthetic faces

In real video-recording or frame recoding, the person can arbitrary move the head, and the facial recognition algorithm must be robust enough to recognize facial features for certain degree of such a "distortion". Thus, 3D face modeling can be used to generate the test sets for testing or verification of the recognition algorithms.

Use your 3D face model, created in Exercise 1.

The sample (synthetic) images named 1.bmp .... 9.bmp are located in the lab directory, and the number of images, M, is set up in the code as 9. Choose your own number of images.

- Use FaceGen 3.3. to generate synthetic faces rotated by $15^0, 30^0, 45^0$ (approximately) and/or more different degrees. Perform face recognition and record the results.
- Repeat the same actions for the face model with morphing (to create various expression).
- Repeat the same actions for the face model with aging (now, at 30,40,..,60 years old etc.) and record the calculated differences.
- Repeat the same actions for the neutral face using 4 various backgrounds, and record the calculated differences.
- Repeat the same actions for the neutral face using 4 different lighting conditions, and record the calculated action units.
- Evaluate the true and false match and reject rates.
- Compare the algorithm performance for both real and synthetic images. Draw conclusions and make suggestions on the procedure improvement.

4 Laboratory Report

In this report, include:

- Your procedure description, illustrations, conclusions (20%).
- Answers to questions, with illustrations where appropriate, to Exercise 3 (30%).
- Answers to questions, with illustrations where appropriate, to Exercise 4 (30%).
- Script of the modified code for the identification procedure, with thresholds determined (20%).

5 Acknowledgments

We wish to thank research students, especially S. Samoil and A. Shier, who helped collect images for this lab, and contributed to the projects implemented in the Biometric Technologies Laboratory. The FaceGen Software is the product of Singular Inversion Inc., that is under the educational license in the ECE department.

Svetlana Yanushkevich

November 2, 2015