Interactive Nature-Inspired Heuristics for Automatic Facial Composite Generation

Tuğba Akbal
Dept. of Computer Engineering
Istanbul Technical University
Maslak, Istanbul, 34469, Turkey
tuba_akbal@yahoo.com

Gül Nildem Demir
Dept. of Computer Engineering
Istanbul Technical University
Maslak, Istanbul, 34469, Turkey
nildemdemir@hotmail.com

Alp Emre Kanlıkılıçer
Dept. of Computer Engineering
Istanbul Technical University
Maslak, Istanbul, 34469, Turkey
emreknlk@hotmail.com

Merve Can Kuş
Dept. of Computer Engineering
Istanbul Technical University
Maslak, Istanbul, 34469, Turkey
mervecankus@gmail.com

Fatma Hülya Ulu
Dept. of Computer Engineering
Istanbul Technical University
Maslak, Istanbul, 34469, Turkey
hulya.ulu@gmail.com

ABSTRACT

The aim of this project is to experiment with different interactive nature-inspired heuristics to automatically generate a target picture through determining the vector of parameter values of an active appearance model (AAM). The AAM forms a statistical model of the human face from an input database of face images. Using this model, it is possible to generate faces which may not even exist in the original database. The nature-inspired heuristics used in this study are genetic algorithms, evolutionary strategies, particle swarm optimization and differential evolution. In the interactive versions of these heuristics, users get involved in the algorithm in either the fitness evaluation or the selection stages.

Categories and Subject Descriptors

I.2.8-Problem Solving, Control Methods and Search

General Terms

Design, Algorithms, Human factor

Keys

Nature-inspired heuristics, interactive nature-inspired heuristics, computerized facial composite generation, active appearance model.

1. INTRODUCTION

In case of a crime, the police usually asks the witness to describe the suspect's face. Presently there are three different methods of generating such facial composites.

In the first method, a sketch artist draws the face of the suspect as the witness verbally describes it.

The second method is based on using a computer based system such as E-FIT [1] and PROfit [2]. These systems require the selection of individual facial features (eyes, nose, mouth etc.) from a large electronic database. The witness first chooses these facial features and then specifies their positions on the face with the help of a trained operator.

The last method is based on using a computer based face generation tool such as Evo-FIT [3] and Eigen-FIT [4]. These tools use nature-inspired heuristics such as evolutionary algorithms (EA) [5] to generate different facial composites. At each stage of the algorithm the user is asked to assign scores or ranks to each face based on the similarity to the target face. As a result of this scoring/ranking process, faces which are more similar to the target face are generated in each iteration.

The first two methods strongly depend on the current psychological and emotional state of the witness. The witness is not only required to recall the face but also he/she needs to give an accurate description of each of the facial features. These methods also require the manual modification of the individual facial features to make the face look more like the target. Moreover, a skilled operator who understands the witness well plays a big part in generating a face using these methods. The requirements above are also psychologically challenging. Usually people are more likely to recall the face as a whole than to recall facial features separately. The existing implementations of the third method do not include the problems mentioned above. They are face recognition systems and do not work as recall-based. Instead of individual facial features, eigenfaces are used.

In this study, we used an approach inspired from the two successful face generation tools Evo-FIT and Eigen-FIT. We used the parameter vector of length n of the active appearance model (AAM) [11] to generate the target face through a user-friendly interface. We handled this problem as an optimization problem and used several interactive nature-inspired heuristics to obtain the AAM parameter vector representing the face most similar to the target. The nature-inspired heuristics used in this study are two versions of genetic algorithms (GA) [5], evolutionary strategies (ES) [18], particle swarm optimization (PSO) [19] and differential evolution (DE) [17]. We built a database of fifty-one pictures of professors, assistants, students of our faculty and administrative staff, and created the AAM from the pictures in this database. After the implementation of each of the heuristics, we conducted tests to evaluate their performance. The tests are designed to evaluate the algorithms from two different perspectives: Their performance in the non-interactive mode and the performance of their interactive versions using the interface. In the second set of tests, the suitability of the interactive version of each approach will be examined. Human interaction is involved in the fitness evaluations or the selection stages. This is a subjective process since the user can assign different fitness values to the same faces at different evaluations. Moreover, evaluating many faces may cause the user to get exhausted and become careless. In such cases, the interactive versions of the algorithms can have worse performance than their non-interactive counterparts.

In Section 2, the AAM will be presented in detail. Section 3 will discuss related work, especially Evo-FIT [3] and Eigen-FIT [4]. Brief descriptions of the implementation of the interactive nature-inspired heuristics will be given in Section 4. Section 5 will explain methods to test the system. Finally, Section 6 will discuss the conclusions and future work.

2. THE ACTIVE APPEARANCE MODEL

The AAM [11] forms a statistical model of the human face from an input database of face images. Using this model, it is possible to generate faces which may not even exist in the original database. Therefore, AAM is considered as a powerful generative model which is able to represent different types of objects. AAM works according to the following principle: A face image is marked with n landmark points. The content of the marked face is analyzed based on Principal Component Analysis (PCA) of both face texture and face shape. Assume that the n landmark points $\vec{x}$ and the intensities on these landmarks $\vec{g}$ are as given below.

$$x = [x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_n]$$
$$g = [g_1, g_2, \ldots, g_n]$$

These are reduced to a more compact form through PCA such that

$$x = \bar{x} + \Phi_s b_s$$
$$g = \bar{g} + \Phi_g b_g$$

To remove the correlation between shape and texture model parameters, a third PCA is applied on the combined model parameters such that

$$x = \bar{x} + \Phi_s W_s^{-1} Q_s c$$
$$g = \bar{g} + \Phi_g Q_g c$$

where $b = \begin{bmatrix} W_s b_s \\ b_g \end{bmatrix}$ and $b = \begin{bmatrix} Q_s \\ Q_g \end{bmatrix} c$.

3. RELATED WORK

Composite face generation programs can be summed up in two categories: Programs based on the selection of facial features from a database such as E-FIT [1] and PROfit [2][6] and programs that generate faces automatically using a nature-inspired heuristic, like Evo-FIT [3][7] and Eigen-FIT [4][8][9]. Since our approach falls into the latter category, we will only explain Evo-FIT and Eigen-FIT.

3.1 EVO-FIT

Evo-FIT uses an evolutionary algorithm approach. Initial faces are generated through a PCA shape and texture model as a whole. The user selects from a larger face set a small number of faces which look most like the target. An evolutionary algorithm generates a new face population based on the selection of the user through crossover and mutation. Evo-FIT can also import hairstyles from a PROfit database, but hair is used as an external parameter and is not optimized by the EA. The selected hairstyle is applied to all faces during the EA run. With its interface to a photo editing software, Evo-FIT provides the option to alter the face images at run-time (e.g. move eyes closer, change shape of eye, etc). Good results comparable to those of E-FIT are reported [7].

3.2 Eigen-FIT

Eigen-FIT uses the AAM and an elitist GA to form faces. Three versions of Eigen-FIT are tested in [8] and [9]. The first one uses breeding between the elite individual and others in the population. At each generation all offspring are rated between 1 and 10. The second one does not include a rating; best faces are selected from among the offspring. The last version uses breeding only between the best individual and another individual from the population. One offspring is generated at each generation. The offspring and the best individual are shown to the user to choose the best one. Eigen-FIT also allows external feature modification during the EA runs. Reports on Eigen-FIT have good results [8], [9].

4. THE FACE GENERATION SYSTEM

We used five different nature-inspired heuristics (NIH) to produce the AAM parameter vector of the target face. These are interactive steady-state genetic algorithms (ISSGA) [13], interactive generational genetic algorithms (IGGA) [13], interactive particle swarm optimization (IPSO) [14], interactive evolutionary strategies (IES) [15] and interactive differential evolution (IDE) [16].

The AAM software AAM-API [12] is used in the generation of AAM parameter vector of faces. AAM-API can either be used as a traditional API by linking in an AAM library, or it can be used as a precompiled command line program [12]. In our system we used it as a command line program.

The flow of the Interactive NIH-based (INIH) face generation system can be described as follows: The user selects the algorithm as ISSGA or IGGA or IPSO or IES or IDE to be used in the face generation. The initial population is generated randomly and displayed with the help of the AAM-API. At each stage of the algorithms the user selects/ranks the presented faces based on their similarity to the target face. This input is used in the evolutionary process of the algorithms to generate new AAM parameter vectors. Face images corresponding to the AAM parameter vectors are displayed on the screen. If the user is satisfied with at least one of the faces, he/she stops the run of the algorithm and selects that face. The selected picture is shown as the result. Otherwise the algorithm continues to run.

4.1 User Interface

The user interface is developed in the C++ language. When the program starts, the algorithm selection screen appears. The user has five algorithm options listed with radio buttons: interactive evolutionary strategies (IES), interactive differential evolution (IDE), interactive particle swarm optimization (IPSO), interactive generational genetic algorithm (IGGA) and interactive steady state genetic algorithm (ISSGA). The user selects the algorithm and clicks the "Start" button to continue.

4.1.1 IGGA

On the screen, initial faces are shown to the user. Under each image there is a combo-box with number elements 1 to population size N. In each iteration, the user assigns fitness scores to each face based on their similarity to the target face. The face with score N is the face which looks most like the target. Same scores can be given to different faces. After the fitness assignment, the user clicks the "Next Step" button to process the images with IGGA [13]. When the iteration ends, new images are shown to the user. If at any iteration the user is satisfied with one of the pictures he/she should give the highest score to that picture. After clicking the "Stop" button only the face with the highest score is shown.

4.1.2 ISSGA

In the initial screen three faces are shown to the user. Under each image there is a check-box. At each iteration, the user selects two images from among the three presented on the screen, the ones which look most like the target face. These two selected images replace the parent individuals in the population. After the selection, the user clicks the "Next Step" button to process the pictures with the ISSGA [13]. If at any iteration the user is satisfied with the given images he/she should click the "Stop" button. After clicking the "Stop" button, all the face images created from the final population are displayed on the screen. User selects the image which looks most like the target and hits the "Finish" button to complete the process. The selected face is shown on the screen.

4.1.3 IPSO

In the initial screen four faces, representing the four initial particles are shown to the user. Under each image there is a check-box. At each iteration, the user is shown the image represented by each of the current particles together with their overall best images and is asked to select the better one which now becomes the overall best image for that particle. At end of the iteration, the user is shown four images which are the overall best of each particle and is asked to select the global best. After the selection, the user clicks the "Next Step" button to process the pictures with the IPSO [14] algorithm. If at any iteration the user is satisfied with the given images he/she should click the "Stop" button. After clicking the "Stop" button, the image represented by the current global best particle is displayed and the user is asked to push the "Finish" button to complete the process.

4.1.4 IDE and IES

For both of the algorithms, in the initial screen sixteen faces are shown to the user. Under each image there is a check-box. At each iteration, the user selects four images, the ones which look most like the target face. These four selected images are used to generate further populations. After the selection, the user clicks the "Next Step" button to process the pictures with the IES [15] or IDE [16] algorithm. If at any iteration the user is satisfied with the given images he/she should click the "Stop" button. After clicking the "Stop" button, the last four images are shown to the user. The user selects the image which looks most like the target and hits the "Finish" button to complete the process. The selected face is shown on the screen.

4.2 Implementation

We took fifty-one pictures to build our database and annotated the face images with AAM-API to compose the AAM.

In our problem, chromosomes represent faces and faces are defined as a set of AAM parameters. These parameters are real numbers in the range [-0.3,0.3]. The number of AAM parameters used in the model is 17. Therefore each chromosome has n=17 genes represented as real numbers. The initial population for all algorithms is generated randomly according to a Gaussian distribution with mean 0 and standard deviation 0.1. All other parameter settings for each algorithm are explained in detail in [13], [14], [15] and [16].

5. EXPERIMENTS

Experiments consist of two groups to evaluate the algorithm from two different perspectives: Analyzing the performance of the algorithm based on solution quality and analyzing the algorithm based on its performance in the interactive mode. For the first group of tests, a fitness function needs to be defined and the algorithms should be run in a non-interactive mode. For the second group of tests, all algorithms should be executed interactively through the interface.

5.1 Performance based on Solution Quality

The tests conducted to analyze the performance of the algorithms based on solution quality have two major targets: Identification and tuning of algorithm parameters (AAM parameter range, mutation rate, crossover rate etc.) and discussion of the algorithm performance based on generated final solutions. Algorithms have to be run in a non-interactive mode without the interference of the user to conduct this type of tests. The interference of the user makes it harder to find out the algorithm parameters. The reason is that the user has to run the algorithm many times, but after some runs he/she gets bored and tired. Therefore, it is better to run the algorithm automatically. As mentioned previously, in the interactive mode, the algorithms get the necessary fitness values from the user as input. In the non-interactive versions, the fitness values must be evaluated automatically. In our system, fitness values of faces are evaluated as follows: A target face image is specified. The pixelwise distance of the target image from the image represented by the AAM parameter vector is calculated. The generated faces with smaller distances to the target face are more fit. Several runs should be completed to experimentally determine the required parameters for each approach.

After the determination of the algorithm parameters, the algorithm results can be evaluated. Several runs with the determined parameter settings should be performed to evaluate how well the solutions generated by the algorithm in the non-interactive mode are. Solutions should be evaluated based on the best found solution and the number of fitness evaluations required to achieve the best solution.

5.2 Analyzing the Performance of the INIHs

In this case, the interactive versions of the algorithms are run. To asses the performance of each approach, several tests are designed. Target images are selected based on some factors:

- whether the image is in the database or not
- whether the user knows the person represented by the image or not

In this part of the experiments, after the target images are determined, N users are asked to run each algorithm for each target image. For each algorithm and each image, the AAM parameter vectors produced by the N users are averaged to give a mean face image. Then these mean face images are shown to test subjects other than the test users who produced the images. These subjects are asked to name the person in the image. Algorithm performances will be evaluated based on the recognition rates of the produced images.

Another factor which determines the usability of an approach in the interactive mode depends on the amount and the nature of work the user has to do. For example it is easier for a person to choose a subset of images from a larger set than scoring or ranking each picture presented to him/her. Also the convergence properties of the algorithms play an important role in the interactive mode. As the number of iterations the user has to make -thus the number of images to be evaluated- increases, the performance of the user will deteriorate because he/she will get tired and bored and will not pay attention properly. The algorithms will also be evaluated based on the number of iterations required and the total number of images viewed by the user.

5.3 Current Status and Results

Currently the whole system has been built and made to work successfully. The parameters are currently specified intuitively and based on settings recommended in the related literature without conducting in-depth experiments.

The tests on tuning the parameters automatically and performance analysis based on solution quality (non-interactive mode tests) are not completed yet. The interactive mode tests are completed on a small experimental set which consists of 4 target face images, 8 test subjects to generate the images and 10 test subjects to recognize and name the generated faces. Details and discussion of the results are given in [13], [14], [15], [16].

6. CONCLUSION AND FUTURE WORK

In this work, we have completed the target system successfully. Initial results are encouraging, but more experiments are needed. The test results are not sufficient to make any solid generalizations and comparisons between the different methods. However, they are enough to show that this is a feasible system and that the interactive versions of the nature-inspired heuristics we have chosen to implement are suitable for the problem.

This work presents a preliminary study, therefore it aims to give an overall performance analysis and an idea about the applicability of the project. More work has to be done to make the project compete with similar projects in literature. Possible future enhancements:

- Increasing the number of face images in the face database to improve the AAM
- Making face properties like size, shape, placement editable during run-time and updating the corresponding AAM vector respectively
- Building the AAM models of individual facial features separately and thus allowing freezing of a facial feature during the process while the other features are allowed to change
- Adding hair style, moustache, beard and also accessories to help during the generation and also the recognition stages

7. ACKNOWLEDGEMENTS

This work has been developed as the result of a senior year graduation project by the authors under the supervision of Dr.A.Şima Etaner- Uyar (etaner@cs.itu.edu.tr) and Binnur Kurt (kurt@cs.itu.edu.tr) in the Computer Engineering Department of Istanbul Technical University, Turkey (http://www.cs.itu.edu.tr/).

8. REFERENCES

[1] "E-FIT: Electronic Facial Identification Software for Police Departments", New England Press Inc., 2004, http://www.efitforwindows.com

[2] "PROfit: The Professional Photo-fit System of Choice", ABM United Kingdom Ltd., http://www.abm-uk.com/uk/products/profit.asp

[3] Frowd C.D., Hancock P.J.B., "EvoFIT: Facial Composite System for Identifying Suspects to Crime", Department of Psychology, Sterling University, http://www.evofit.co.uk

[4] "EigenFIT: Facial Composite Technology", VisionMetric Ltd., http://www.eigenfit.com

[5] Eiben A.E., Smith J.E., Introduction to Evolutionary Computing, Springer, 2003.

[6] "PROfit: A Photofit System using Highly Advanced Facial Composition Tools", ABM United Kingdom Ltd., http://www.abm-uk.com/uk/pdfs/profit.pdf

[7] Frowd C.D., Hancock P.J.B., Carson D., "EvoFIT: A Holistic, Evolutionary Facial Imaging Technique for Creating Composites", *ACM TAP*, Vol. 1 (1), pp. 1-21, 2004.

[8] Gibson S. J., Pallares-Bejarano A., Solomon C. J., "Synthesis of Photographic Quality Facial Composites using Evolutionary Algorithms", in Proceedings of the British Machine Vision Conference, Editors R. Harvey and J.A. Bangham, pp221-230, 2003.

[9] Solomon C. J., Gibson S. J., Pallares-Bejarano A., "EigenFit - The Generation of Photographic Quality Facial Composites", The Journal Of Forensic Science, 2005.

[10] Eiben A. E., Schoenauer M., "Evolutionary Computing", *Information Processing Letters*, 82(1): 1-6, 2002

[11] Iain Matthews, Simon Baker, "Active Appearance Models Revisited," International Journal of Computer Vision, Vol. 60, No. 2, pp. 135-164, November 2004 .

[12] "AAM-API", http://www2.imm.dtu.dk/~aam/aamapi/

[13] Demir G. N., "Interactive Genetic Algorithms for Facial Composite Generation", 2006 Genetic and Evolutionary Computation Conference, Undergraduate Student Workshop (GECCO-UGWS 2006), July 2006.

[14] Akbal T., Ulu F. H., "Interactive Particle Swarm Optimization Algorithm for Automatic Facial Composite Generation", 2006 Genetic and Evolutionary Computation Conference, Undergraduate Student Workshop (GECCO-UGWS 2006), July 2006.

[15] Kus M. C., "Interactive Evolutionary Strategies for Automatic Facial Composite Generation", 2006 Genetic and Evolutionary Computation Conference, Undergraduate Student Workshop (GECCO-UGWS 2006), July 2006.

[16] Kanlikilicer A. E., "Interactive Differential Evolution for Facial Composite Generation", 2006 Genetic and Evolutionary Computation Conference, Undergraduate Student Workshop (GECCO-UGWS 2006), July 2006.

[17] Storn R., Price K., "Differential Evolution - A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces", Technical Report TR-95-012, International Computer Science Institute, Berkeley, CA, 1995.

[18] Beyer H.G:, Schwefel H.P., "Evolution strategies *A comprehensive introduction",* Natural Comp. 1: pp. 3–52, 2002.

[19] Eberhart R. C., Kennedy J., Shi Y., Swarm Intelligence, 1st Ed., Morgan Kaufmann, 2001.