Interactive Differential Evolution for Facial Composite Generation

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

ABSTRACT
The purpose of this study is to generate facial composites by interactive differential evolution (DE). Faces are modeled using the Active Appearance Model (AAM). DE is similar to an evolutionary algorithm and yields superior results in optimization problems. In this study, an interactive version of DE is implemented in such a way that candidate faces generated by the AAM is presented to the user and the user selects a subset of these faces based on their resemblance to the target face. Experimental results show that DE is successful in converging to a face rapidly and it is robust to fluctuations in the user's decisions.

Categories and Subject Descriptors
I.2.8-Problem Solving, Control Methods and Search

General Terms
Design, Algorithms, Human Factors

Keys
Differential Evolution, Interactive Differential Evolution, computerized facial composite generation, Active appearance model.

1. INTRODUCTION
In criminal investigations, witnesses are asked to describe criminals' faces. There are some methods developed for better generation of faces. Our approach for face generation is based on evolutionary algorithms. We implemented an Interactive Differential Algorithm (IDE) which takes in input from the user to generate a likeness to the target face through several iterations of a sequence of steps. These are selection, mutation and recombination steps. New faces are generated at the end of each iteration by evaluating the parameters estimated based on user choices.
The following sections of this paper cover the following topics; Section 2 describes the IDE-based Face Generation System. Algorithm design and implementation issues are also explained in this section. In section 3 experiments and test results are presented. Conclusions and future work possibilities are the topic of section 4.

2. DIFFERENTIAL EVOLUTION ALGORITHM

2.1 Non-Interactive Differential Evolution
Differential Evolution is first introduced by Storn and Price in 1996. It is a population-based optimization algorithm for solving problems that cannot be solved analytically [1]. The block diagram of DE is shown below.

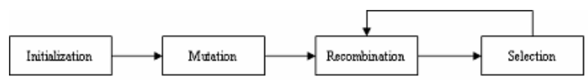

Figure 1. Work flow of DE

At initialization, bounds on the parameters are determined and for each individual, which represents a face in our program, parameters are randomly generated.

$$x_{i,G} = \left[x_{1,i,G}, x_{2,i,G}, \ldots, x_{D,i,G} \right], \quad i = 1, 2, \ldots, N$$

At the mutation step, three parameter vectors x_{r1}, x_{r2} and x_{r3} are selected for each parameter vector in i. in order. Note that all vectors in this step are distinct from each other. Mutation continues with adding the weighted difference of two of the vectors to the third.

$$V_{i,G+1} = X_{r_1,G} + F\left(X_{r_2,G} - X_{r_3,G} \right)$$

F is a mutation factor and it is selected between 0 and 2. Result vector v_i is called a donor vector.
At the recombination step, new individuals are created by combining the donor vector which is created in the mutation step with the old individual's vector. Combination takes place according to the following rule.

$$u_{j,i,G+1} = \begin{cases} v_{j,i,G+1} & \textit{if } \text{rand}_{j,i} \le CR \text{ or } j = I_{\text{rand}} \\ x_{j,i,G} & \textit{if } \text{rand}_{j,i} > CR \text{ and } j \ne I_{\text{rand}} \end{cases}$$

where v_j is the donor vector for the j^{th} parameter, *d* is the number of total parameters in an individual. *L* is a random number between 0 and *D*, and x_i is the old individual from the previous recombination. $\langle n \rangle_d$ represents the *n* modulo *d*.
After recombination, selection is done by using a fitness function *f*. New individuals for the next generation are selected as those with the best fitness function values. Mutation, recombination and selection steps continue until desired approximations are achieved.

2.2 Interactive Differential Evolution

Interactive differential evolution is based on differential evolution in which the fitness function f is removed and evaluations are performed by a user. In the selection step, the user selects the best images and these images are used to create solution candidates for the next generations through mutation.

2.3 Implementation

We built a database of pictures consisting of 51 photos of our university personnel. The photos are annotated and eigenfaces are extracted from the pictures through a principal component analysis. Using the AAM model [2] we obtained, faces are generated using 17 parameters and thus the vector length is also 17. During initialization, for each parameter in the vector, the IDE algorithm produces normally distributed random numbers with average 0 and standard deviation 0.2. The boundaries for each parameter are set as [-0.3,0.3]. Any values that are outside these boundaries are mirrored. These values are chosen based on the model parameters of the average face obtained from the model. Iterations are continued until the user is satisfied with the result and decides to stop. At each iteration, the user is presented with 16 new images. The user is required to select 4 of the images that resemble the target image. These 4 images are used to create the new group of faces for the next iteration. The user interaction is achieved through a user interface developed for this study [3].

3. EXPERIMENTS

We tested the program on four faces. Two of these faces, which are also in the database, belong to people who are well known by the testers. The third face belongs to a person well known to the testers but is not in the database and the final face, which also is not in the database, belongs to a person who is not known by the testers and is seen very briefly.

Eight testers are used. For each face, the IDE Algorithm is executed by these 8 testers. So for each face there are 8 generated output images. After the testing, the vectors of these 8 generated images are averaged to obtain the mean face. 10 subjects, who know the four people used as the target faces well, are shown the resulting mean images and are asked to name the persons. The resulting recognition rates are given below.

First Face: %100 recognition between figure 1a and figure 1b. Mean image is generated using 6.3 average iterations.

Second Face: %60 recognition between figure 2a and figure 2b. Mean image is generated using 4.5 average iterations.

Third Face: %40 recognition between figure 3a and figure 3b. Mean image is generated using 4.9 average iterations.

Fourth Face: %20 recognition between figure 4a and figure 4b. Mean image is generated using 4.8 average iterations.

Target Face **Mean of Test Faces**

Figure 1-a

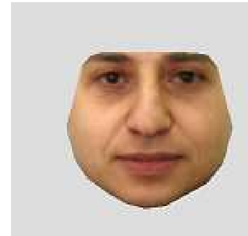

Figure 1-b

Figure 2-a

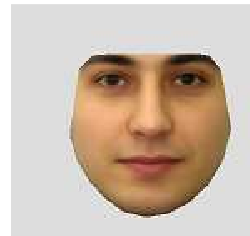

Figure 2-b

Figure 3-a

Figure 3-b

Figure 4-a

Figure 4-b

The test results are not sufficient for determining the exact behavior of the algorithm. We currently do not have a database which is large enough for obtaining realistic results. Also we should do more testing to obtain more realistic mean image of test results.

4. CONCLUSION AND FUTURE WORK

In this project, the IDE implementation is successfully completed. This preliminary results show that the proposed solution works fine even for our small face image database. We plan to test the proposed algorithm on a larger database. Additional tests are necessary to compare the proposed solution with the similar projects in the literature. We are planning to adapt the algorithm and enhance the GUI such that the witness can edit the individual face components during the run. We also intend to create component based face models, so that the user can decide that a component is close enough to the actual face and can keep it fixed in remaining iterations. In order to use the proposed method in real forensic applications we are planning to add hair styles and several accessory options to the model.

5. ACKNOWLEDGMENTS

This work is being developed as a senior year graduation project under the supervision of Dr. A. Şima Uyar and Binnur Kurt by 8th term student Alp Emre Kanlıkılıçer in the Computer Engineering Department of Istanbul Technical University, Turkey.

6. REFERENCES

[1] Fleetwood K., "An Introduction to Differential Evolution",http://www.maths.uq.edu.au/MASCOS/Multi-Agent04/Fleetwood.pdf

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

[3] Akbal T., Demir G. N., Kanlikilicer E., Kus M. C., Ulu F. H., “Interactive Nature-Inspired Heuristics for Automatic Facial Composite Generation”, Genetic and Evolutionary Computaton Conference Undergraduate Student Workshop, 2006.