Interactive Particle Swarm Optimization Algorithm for Automatic Facial Composite Generation

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

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

ABSTRACT
Particle Swarm Optimization is a type of nature-inspired heuristic which is based on bird flocking and fish schooling in nature. In interactive particle swarm optimization, user evaluations are used to determine the fitness of particles and the end of iterations. In this study, Interactive Particle Swarm Optimization approach is used as a solution to face generation problem. And "success" of the approach is reported based on the "percentage of images correctly recognized".

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

General Terms
Design, Algorithms, Human factor

Keys
Particle Swarm Optimization, Interactive Particle Swarm Optimization, computerized facial composite generation, Active appearance model.

1. INTRODUCTION
Particle Swarm Optimization (PSO) is a new algorithm (1995) for finding optimal solutions to numerical and qualitative problems. PSO is developed by a social-psychologist (James Kennedy) and an electrical engineer (Russell Eberhart)[1] inspired by the behavior of fish schooling and bird flocking. A scenario is that a group of birds try to find food. They do not know the actual place of the food but in each try they learn how far they are from the food location. The method used by the birds is to follow the bird in the group which is nearest to the food (best position). In PSO, the birds are represented as particles and these particles are updated with respect to the best value in each iteration to reach the end [2].

The most important point in an optimization problem is how to define the cost function. The cost function represents the objectives of the expected solution by a mathematical function. Usually, the aim of optimization is to find this function's minima or maxima [3].

The relationship among the objects can be more complex, so it can be hard to find a definition for the cost function. Therefore, in such cases it is preferred to use an interactive approach. In the interactive version of PSO (Interactive Particle Swarm Optimization), the aim is to enhance the quality of the optimization method by using visual skills, knowledge and strategic sense of a human user [3]

In this study, Interactive Particle Swarm Optimization (IPSO) is used as a solution to the Face Generation Problem detailed in [4].

Genetic and Evolutionary Computation Conference GECCO'06,
July 8–12, 2006, Seattle, WA, USA.

2. IPSO for Facial Composite Generation
2.1 Particle Swarm Optimization
The PSO algorithm initially has a population of possible solutions called *particles* (the swarm) which are randomly generated. A particle has an n-dimensional vector called its *position* represented by *x* where *n* is the number of object variables. A particle has a *fitness function* f(x) which is based on the objective function. A particle also has an n-dimensional vector called *velocity* represented by *v* which shows the possible move of the particle. The best solution accomplished by each particle contributes in determining the possible moves of the particle and *Particle_best* is this best solution represented by *p* and *Global_best* is the best solution of the whole population so far represented by g. t_{max} is the maximum number of iterations and t is the current iteration [2].

```
The PSO algorithm [2]
procedure PSO; {
  Initialize particle's position x and velocity v;
  For t=1 to t_max {
    For j =1 to N {
       Calculate f(x_j) fitness value;
       if f(x_j) > f(p_j) then p_j = x_j;
    }
    Choose the global best: g = argmaxf(pj);
     For j =1 to N {
       Update x_j particle position;
       Calculate v_j particle velocity;
     }
  }
}
```

In each iteration every particle's position and velocity is updated by the formulas below:

Updating the position of j-th particle at t-th iteration:

$$x_j(t+1) = x_j(t) + v_j(t)$$

Updating the velocity of j-th particle at t-th iteration:

$$v_j(t+1) = w(t)v_j(t+1) + c_1u_1(p_j(t)- x_j(t)) + c_2u_2(g_j(t)- x_j(t))$$

where c_1, c_2 are learning factors, u_1, u_2 are uniformly distributed random numbers [0,1) and w is an inertia parameter formulated below:

$$w(t) = 0.4 + (t_{max} - t)/(2^* t_{max})$$

The inertia weight w is used to control the effects of former values of velocity on the current velocity [2].

2.2 Interactive Particle Swarm Optimization

Instead of a fitness function, a user is asked to evaluate the solutions. As given in the pseudocode below, for each newly generated particle, its *pbest* and its current value is evaluated to determine the new *pbest* by the user. At the end of each iteration, the new *pbest* particles are evaluated by the user to determine the new *gbest*. In this method, assuming there are N particles in the population, at each iteration, the user views and evaluates (3N) solution candidates (particles).

<u>The IPSO algorithm [2]</u>

```
procedure IPSO; {
  Initialize particle's position x and velocity v;
  For t=1 to t_max {
    For j =1 to N {
       Show x_j and p_j solutions to user;
       If x_j is better than p_j (byuser) then p_j = x_j;
    }
    User selects the best p_i: g = p_i i € [1,N];
     For j =1 to N {
       Update x_j particle position;
       Calculate v_j particle velocity;
     }
  }
}
```

2.3 The Interactive Algorithm Flow

To allow the users to interact with the PSO, a user interface is developed as detailed in [4]. Assuming that the population consists of N particles corresponding to N faces, from the N initial faces, the user selects the most similar picture to the target one. The PSO algorithm generates N new particles based on the best one selected from the initial population. In the following iterations, for each particle, the user is shown the best image the particle has achieved so far and the new image represented by the particle. The user has to choose the better one which now becomes the new best picture for that particle. As the final step of each iteration, the users are asked to select the best one from among the best pictures for each particle. The user repeats these operations until she is satisfied with the resemblance of the generated image with the target.

Table 1. Parameter Settings for IPSO

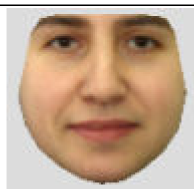
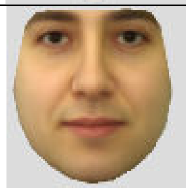
constant	description	value
pmax	length of the interval of position values, since position values will be symmetrical around 0 and will change between [-0.3,0.3] for our problem	0.6
v_max	length of the interval of the velocity values, since velocity values will be symmetrical around 0 and will change between [-0.3,0.3] for our problem	0.6
c1, c2	used in the velocity update	2
dimension	the size of the vector of AAM [4] parameters to represent the face	17
t_max	the maximum number of iterations and is taken as a high value to get better results	2000
a	used in velocity update	10
n	the number of particles which determines the number of pictures seen on the screen in each iteration	4

3. EXPERIMENTS

The algorithm is run for 4 faces by 8 different users. The 4 faces are selected as follows:

- two faces which exist in the database [4] and are well known by the users
- one face which does not exist in the database [4] and is well known by the users
- one face which does not exist in the database [4] and is not well known by the users

The average value of these 8 runs for each face, is taken to see the algorithm's success. These results are printed as a hard copy and 10 persons, who are different from these 8 users are asked to name the persons in the pictures. The rate of successful recognitions is used to evaluate the success of the approach. In our experiments, from the first picture to the fourth picture, respectively 2, 3, 4 and 2 persons from the 10 people who were asked, recognized the faces correctly. Of course, the results are not sufficient to make a generalization, because a small number of pictures are used and the number of tests must be more. The average values of the face parameters can be made more meaningful by increasing the number of the tests.

Original Picture	Generated Picture	Recognition percentage
		This person is known and his picture is in the database % 20
		This person is known and his picture is in the database % 30
		This person is known and his picture is not in the database % 40
		This person is not well known and his picture is not in the database %20

12 pictures are displayed to the user per iteration. For the above pictures:

i. The users evaluated 6 * 12 = 72 pictures on average, to generate this picture
ii. The users evaluated 5 * 12 = 60 pictures on average, to generate this picture
iii. The users evaluated 4 * 12 = 48 pictures on average, to generate this picture
iv. The users evaluated 5 * 12 = 60 pictures on average, to generate this picture

4. CONCLUSION AND FUTURE WORK

At the time of submission, the first stage of the project has been successfully completed. Further performance tests and analysis are being performed.

As seen in section 3 the performance of algorithm is not good enough. Further improvements in implementation, especially in the interaction of the approach with the user interface, is needed. It is believed that appropriate fine tuning and modifications will increase the performance of algorithm.

Apart from the tests described above, further enhancements may be made to improve the program such as adding the ability to edit the image, the ability to keep certain facial parts fixed while others are being changed, the ability to include accessories (such as glasses, eye patches, ear or nose rings etc) and different hair styles on the generated images.

5. ACKNOWLEDGEMENTS

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

6. REFERENCES

[1] Pomeroy P., “An Introduction to Particle Swarm Optimization“, http://www.adaptiveview.com/articles/ipsoprnt .html, 2003.
[2] Madar J., Abonyi J., Szeifert F., "Interactive Particle Swarm Optimization", 5th International Conference on Intelligent Systems Design and Applications (ISDA'05), pp. 314-319, 2005.
[3] Eberhart R. C., Kennedy J., Shi Y., Swarm Intelligence, 1st Ed., Morgan Kaufmann, 2001.
[4]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.