

# An Interactive Approach Based on Genetic Algorithm Using Hidden Population and Simplified Genotype for Avatar Synthesis

Ja-Yong Lee\*, Jang-Hee Lee\*\*, Hoon Kang\*\*\*

School of Electrical and Electronic Engineering, Chung-Ang University

221, Huksuk-Dong, Dongjak-Gu, Seoul 156-756, Korea

Tel: +82-2-816-8234; Fax: +82-2-816-8234

E-Mail: jalnans@sirius.cie.cau.ac.kr\*, whizkid@sirius.cie.cau.ac.kr\*\*, hkang@cau.ac.kr\*\*\*

## Abstract

In this paper, we propose an interactive genetic algorithm (IGA) to implement an automated 2D avatar synthesis. The IGA technique is capable of expressing user's personality in the avatar synthesis by using the user's response as a candidate for the fitness value. Our suggested IGA method is applied to creating avatars automatically. Unlike the previous works, we introduce the concepts of 'hidden population', as well as 'primitive avatar' and 'simplified genotype', which are used to overcome the shortcomings of IGA such as human fatigue or reliability, and reasonable rates of convergence with a less number of iterations. The procedure of designing avatar models consists of two steps. The first step is to detect the facial feature points and the second step is to create the subjectively optimal avatars with diversity by embedding user's preference, intuition, emotion, psychological aspects, or a more general term, KANSEI. Finally, the combined processes result in human-friendly avatars in terms of both genetic optimality and interactive GUI with reliability.

**Key Words** : Interactive Genetic Algorithm, Avatar Design, Hidden Population, Evolutionary Computations, GUI

## 1. Introduction

Genetic Algorithm(GA) was proposed by John Holland in early 1970s. It is a search algorithm based on the mechanics of natural selection and evolving genetics[1]. GA has been found to be one of the most flexible, efficient and robust among all search algorithms known to artificial intelligence. Because of these properties, this method is now widely used to solve a broad range of different problems[2].

Interactive Genetic Algorithm(IGA) is an optimization method that adopts GA among system optimization based on subjective human evaluation. It is simply a GA technique whose fitness function is replaced by a human user. In general IGA systems, the fitness value is assigned to each individual by human subjective evaluation instead of a chosen fitness function, then GA optimizes the target system to obtain the preferred output based on the user's evaluation. In this sense, we can say that IGA is a technology that embeds human preference, intuition, emotion, psychological aspects, or a more general term, KANSEI, in the target system[3].

The biggest remaining problem of IGA is that the user could be tired by the iterative evaluation process. If the population size is large, it is hard to display all individuals simultaneously or sequentially and the user can be fatigued

easily because the user has to evaluate the fitness of individuals one by one. Also, if the user must pass through a large number of generations until the satisfactory avatar is produced, the user can be tired easily. Because of these reasons, most IGA techniques cannot maintain a lot of population nor a number of generations unlike other GAs.

In this paper, we propose epoch-making IGA techniques that overcome the shortcoming of IGA. To take away the limitation of population, we use the hidden population that is not evaluated by the user. All individuals that belong to the population are classified into several clusters. We use user's binary selection to evaluate the fitness of clusters as well as a fitness of displayed individuals. The hidden population is evaluated on the basis of clusters' fitness. To reduce the number of generations, we use a primitive avatar. The primitive avatar is a basis model that is expected to be the optimal avatar. To lead quick convergence, we create a primitive avatar and give a variety on the primitive avatar to make the final optimal avatar, during GA procedure.

Our avatar synthesis application consists of two stages automatically and interactively. The first stage is detecting facial feature points from the user's input picture. By using these feature points, we create avatars in the second stage. In this stage, we introduce the concept of 'hidden population', 'primitive avatar' and 'simplified genotype' and use these concepts to overcome the shortcoming of IGA.

---

This work was supported by grant no. R01-2000-000-00277-0 from the Basic Research Program of the Korean Science & Engineering Foundation (KOSEF) and by the Korea Association of Small Business Innovation Research

## 2. Detecting Facial Feature Points

To make an avatar, facial feature points must be detected. This is done with the following sequences. (1) Separating a human face from the background. (2) Defining a face elements' area about y-axis. (3) Detecting the feature points of each face element like eyebrows, eyes, nose, lip, and face out-lines.

### 2.1. Separating human face from background

We use the HSV color model rather than RGB color model. The HSV model has Hue component, Saturation component and Intensity component. It has an intensity component with other components, and this model may be free from the illumination to some degree. The human's skin color has a regular range in the hue space and also in the saturation space. We tested many images that contain human faces and easily found the skin color range both in the hue space and in the saturation space.

In spite of that processing, the result image may have some noise, because the background image has similar skin color range in some area. Therefore, we hire another processing, which is called "the size-filtering by labeling". Using this process, the background noise can be significantly eliminated.

### 2.2. Finding the vertical position of Face's elements

The edge image, which can be obtained by filtering with the 3x3 Sobel filter, has a lot of information on facial elements and it also contains facial elements' vertical position. The histogram obtained from the edge image is used to find the vertical positions of face's elements. The histogram is drawn by counting points with not the x-axis but the y-axis. The edge image and its histogram are shown in Fig. 1.

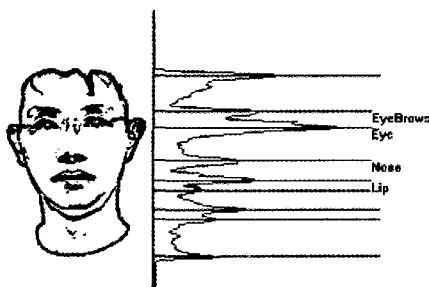


Fig. 1. Edge image and its histogram

It is easily to see that face elements exist only at each histogram's peaks, since human's face elements are generally longer in width than in height, and the edge information is also stronger too in the horizontal direction than in the vertical one. The vertical positions of face elements can be determined by this histogram.

### 2.3. Modified Laplacian of Gaussian filter

The existing edge filters such as Sobel filter, Prewitt filter, Robert filter, Canny filter, etc., have some weakness that they can never eliminate noise, and the result images from these

filters have rough edges. If an enlarged mask is used, a sharper image can be obtained. However, noise also grows with the mask size. Therefore, we use a large, 5x5, modified Laplacian of Gaussian filter to get a sharper edge image and to eliminate noise.

The output of the Laplacian of Gaussian operator,  $h(x,y)$  is obtained by the convolution operator (\*),

$$h(x, y) = \nabla^2 [g(x, y) * f(x, y)] \quad (1)$$

$$h(x, y) = [\nabla^2 g(x, y)] * f(x, y) \quad (2)$$

$$\nabla^2 g(x, y) = \left( \frac{x^2 + y^2 - 2\sigma}{\sigma^4} \right) e^{-\frac{(x^2 + y^2)}{2\sigma^2}} \quad (3)$$

where  $f(x,y)$  is the intensity of each pixel,  $g(x,y)$  is the Gaussian function, and (3) is called the Mexican hat operator.

This paper proposes a modification of the center value. Existing edge filter has characteristics that the sum of all values in the mask is zero. We modified the mask's center value. Fig. 2 represents our modified Laplacian of Gaussian mask.

		-1		
	-1	-2	-1	
-1	-2	18	-2	-1
	-1	-2	-1	
		-1		

Fig. 2. Example of the modified Laplacian of Gaussian

The center value of the mask is originally 16, but if this value increases, the black area and the white area of the result image will be reversed. So, it reduces the computational time to inverse the intensity value. Also, the more center value increases, the less noise is reduced. But it is noticed that larger center values may cause loss of image information.

When we find the feature point of eyes, the center value of mask is set to 19 because eyes have clear shapes. But when we find the feature points of other elements, it is set to 18~19 for other elements having ambiguous shapes since higher center values may cause loss of information. The result images of detecting the facial feature points are shown in Fig. 3

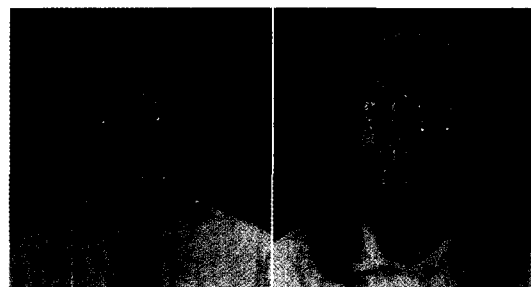


Fig. 3. The result image of detecting facial feature points

### 3. IGA-based Avatar Synthesis

The IGA-based systems use user's selections to evaluate the fitness of individuals. Therefore, if there are a lot of individuals that are evaluated or the user must pass through a large number of generations until the optimal avatar is produced, the user can be tired easily. In case of IGA-based systems, techniques to reduce user's burden have been developed. But, most of these techniques are confined to improve the human interface or GUI [4,5]. The improvement of human interface reduces a fraction of fatigue, but this isn't a fundamental solution.

This paper proposes three epoch-making techniques that reduce user's fatigue effectively to attain the optimal result in a few generations.

- (1) Hidden Population
- (2) Primitive Avatar
- (3) Simplified Genotype

#### 3.1. Hidden Population

General wisdom dictates that a larger population will work more slowly but will eventually achieve better solutions than a smaller population. A small population size causes the genetic algorithm to quickly converge on a local minimum, because it insufficiently samples the parameter space. On the other hand, a large population size takes too long to find and assemble the building blocks to the optimum solution. Experience indicates that the most effective population size is dependent on the problem being solved, the representation used, and the operators manipulating the representation [6].

In general IGA-based systems, the user evaluates fitness of all individuals. In this case, there exists restriction of population size due to user's fatigue. We solve this limitation of population by using "hidden population". The whole population consists of the hidden population and the displayed population. The displayed population is composed of individuals displayed on the screen, which are then evaluated. On the other side, the hidden population is not evaluated directly by the user and is not displayed either. All individuals in the population are classified into several predefined clusters. The fitness value of each cluster is represented by the number of individuals that belong to it.

If the user assigns a low fitness value to an individual, it is eliminated from the population. In addition, an individual that is given a high fitness value creates its clones (clone here means the copy of its gene) as a replacement of the eliminated individual. Therefore, a cluster containing high fitness individuals increases in size. On the other side, a cluster containing low fitness individuals decreases in size. However, since a cluster with individuals that belong to the displayed population doesn't changes in size, this cluster maintains its size in the next generation by the probability. Based on the roulette wheel selection, there is a strong probability that clusters with a large number of individuals could hand down their genes to the next generation. As a result, the fitness of each cluster is evaluated by the user's

subjective selection.

Fig. 4 represents a process that a user's selection is reflected the fitness of clusters. In this figure, whole population size is 15, the displayed population size is 5, the hidden population size is 10, and the number of clusters is 4. The initial population state is shown Fig. 1-(a). If the user selects the individual that belong to cluster A, this means that the user evaluates the individual that belong to cluster A higher than the individual that belong to other clusters. In this case, the individual that belong to cluster B and C is eliminated and the individual that belong to cluster A creates clones that has equivalent gene. As a result, because the size of cluster A increases, there is a strong probability that the cluster A hands down their gene to the next generation.

In the case of cluster D, this cluster has no individual that belong to the displayed population. Therefore, there is no change the fitness of cluster D.

Because this algorithm evaluates the fitness of clusters instead of the fitness of individuals, there is no restriction of the population size. If IGA-based systems use the proposed hidden population algorithm, it can take away the limitation of the population size.

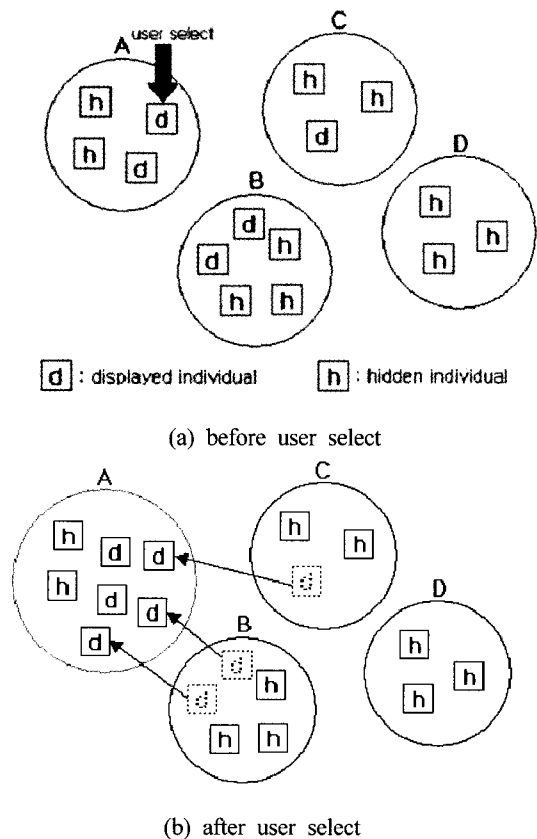


Fig. 4. How to evaluate cluster fitness

#### 3.2. Primitive Avatar

Genetic algorithms typically seed the initial population with entirely random values (i.e., starting from scratch). Often random values are generated between specified ranges. In this case, the process needs a lot of generations to obtain the

subjectively optimal avatar [2]. We use a primitive avatar to prevent the process from going through so many generations. The search space map is used to create a primitive avatar. Fig. 5 is a sample of the search space map.

Its x-axis is the morphing rate of a real avatar that is created by referring directly to the user's facial feature points and an ideal avatar that is created by the ideal facial feature points. Its y-axis is a variation that expresses an impression. If that area is smaller than a criterion, then a primitive avatar is created with the parameters at that point.

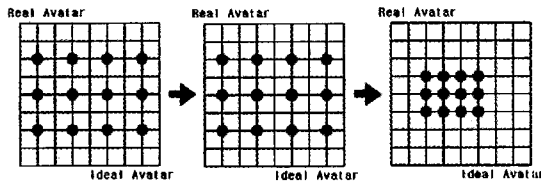


Fig. 5. Search Space Map

### 3.3. Simplified Genotype

Fig. 6 describes how a gene is encoded. Total 8 bits are needed to complete gene encoding. We encoded a cluster (pattern) with upper 4 bits, and lower 4 bits for encoding a morphing rate to merge a primitive avatar with a pattern.

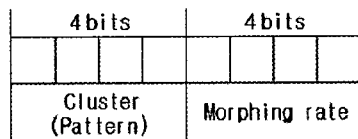


Fig. 6. Simplified Genotype

The regular genotype to be mapped into the phenotype consists of 6 components (face, ear, eye, eyebrow, nose, lip). Each component has floating point parameters (8 facial feature points, 8 texture points, and local coordinate).

The size of the gene or the number of bits used to represent a parameter is important to the length of time needed for the genetic algorithm to converge [2]. If each parameters of component is encoded, then 6528 bits,  $(8+8+1)$  parameters  $\times$  2 xy-point  $\times$  6 components  $\times$  32 bits, are needed to complete the gene encoding. Longer genes slow down the convergence of the genetic algorithm due to a large search area. Each parameter of the component used in the regular gene represents a pattern.

Various patterns pre-designed by the professional designer are classified into several clusters by the impression or emotion. Because this application has been developed for the sake of a non-professional user, these patterns must help a user to choose his or her own optimal avatar.

## 4. System Implementation

The block diagram of the entire system is shown in Fig. 7. First of all, it starts with the detection of the facial feature

points automatically from the user's face. By using these feature points, our application creates a primitive avatar. Next, an initial population is created by using the primitive avatar, and the displayed population is shown on the screen for user's evaluation. If the displayed avatars satisfy the user, one saves the satisfactory avatar and finishes the procedure. However, if there is not the satisfactory avatar, the user evaluates the fitness values of the displayed individuals. The fitness of each cluster is evaluated using the user's binary selection. Then, a new set of offspring is reproduced from existing parents by applying crossovers and mutations. This evaluation and reproduction processes is iterated until the satisfactory avatar is produced.

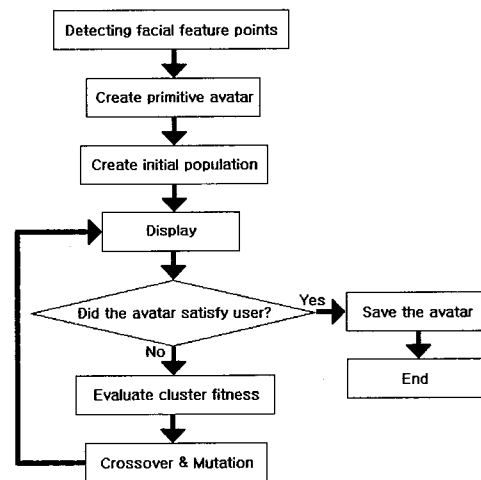


Fig. 7. Block diagram of system

Our avatar synthesis application using IGA is shown in Fig. 8. It shows the current displayed population composed with 12 individuals on a screen. The user selects satisfactory avatars and clicks the next button.

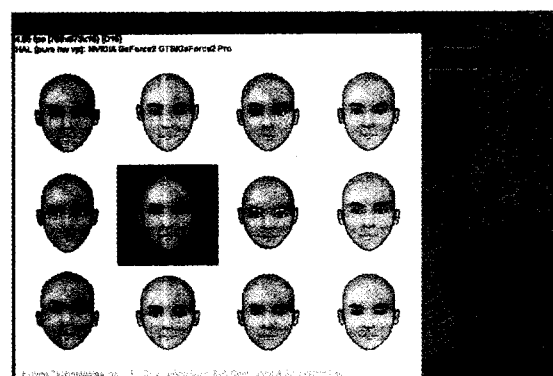


Fig. 8. Avatar synthesis application

This application is performed with Microsoft Visual C++ 6.0 and drawing avatar is done with Microsoft DirectX 8.1 for future extension to 3D avatars

## 5. Experimental Results

### 5.1. Environments

The population is composed of 100 individuals. 12 out of 100 are the displayed population and the rest, 88, are the hidden population. We use one-point crossover that swaps the right part of crossover point, and the crossover rate is 50%. The mutation rate is 3%. As a strategy of evolution, the elitist individuals selected by the user are preserved and transferred into the next generation. Twenty students are employed as subjects. We direct all subjects to create avatars that satisfy themselves. They try the same procedure a week later.

To evaluate the performance of this system, we carry out subjective and convergence tests.

### 5.2. Reliability

The reliability is calculated by comparing the current user's selection with his selections in the recent 10 generations. If the reliability is too low, we can regard user's selection as inconsistent. Judging from our experimental experience, if the reliability is lower than 50%, it is hard to say that we have obtained a convergent result. If so, the best and average fitness values represent low values. Because of these reasons, the experimental data that its reliability is above 50% is authorized.

### 5.3. Fitness Evaluation

In the case of IGA-based system, it is difficult to evaluate the fitness arithmetically, because the user evaluates the fitness of individuals. Therefore, we need the particular yardstick to evaluate the fitness.

To prepare the yardstick for evaluating the fitness, we pick out 96 sample avatars at same intervals from the searching area. We show these sample avatars randomly to the user, and one evaluates the fitness of these sample avatars. The application to evaluate sample avatars is shown in Fig. 9. We accumulate the yardstick to evaluate the fitness by this application.

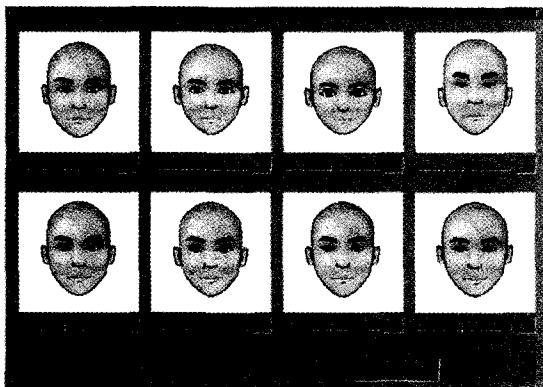


Fig. 9. Sample avatar evaluation application

Another yardstick to evaluate the fitness is the fitness of clusters. Since the users selection is used to evaluate the

fitness of clusters, it is valid that the fitness of individuals is influenced by the fitness of its clusters. The fitness of each individual is evaluated by uniting the sample avatar yardstick and the fitness of its cluster.

### 5.4. Subject Test

A subjective test is to show how much the user is satisfied with the system running. The change of fitness on the average and best is shown in Fig. 10. The best fitness is the fitness of individual that has the highest fitness value in each generation. The average fitness is the arithmetic mean of 100 individuals' fitness values. In this figure, we find that the best and average fitness value are monotonically increase.

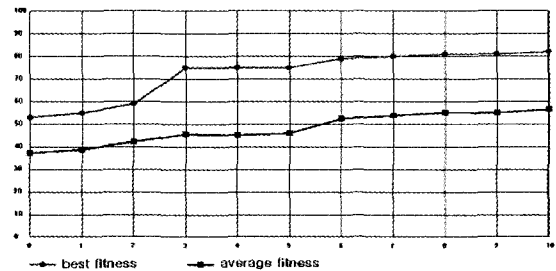


Fig. 10. Change of fitness on the average and best

### 5.5 Convergence Test

It is difficult to show the convergence of IGA with quantitative analysis because IGA is based on the evaluation of human, very different from standard GA [10]. The change of each cluster size is shown in Fig. 11. Because the cluster size is equivalent to the fitness of cluster, this figure means the variation of cluster's fitness. If the subject's reliability is high, the largest cluster size increases about 50 percentages out of the whole population.

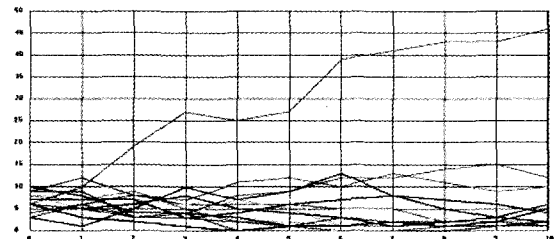


Fig. 11. Change of cluster size

In the upper figure, the largest cluster size increases about 50 individuals. But a non-selected cluster size also increases about 15 individuals occasionally. Consequently, we can regard this application as a system that secures diversity of individuals

### 5.6 Experimental Results

Table 1 shows experimental results. In this table, the average number of generations to obtain the optimal avatar is 13. It means that the user can take the optimal avatar by only 13 binary selections.

The optimized avatar and the original image are shown Fig. 12. Fig. 12-(a) shows the primitive avatar, Figure 12-(b) is the original image, and Fig 12-(c) represents the optimal avatar.

Table 1. Experimental results.

	Max.	Min.	Average
Reliability	100	65	89
Best fitness	90	68	81
Average fitness	65	48	58
Largest cluster size	62	41	49
Generation	22	8	13

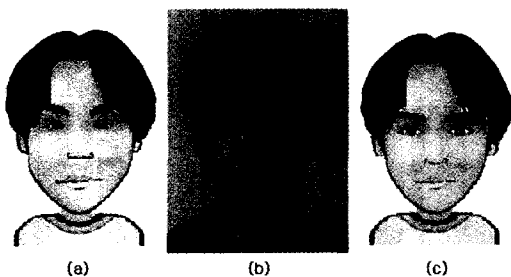


Fig. 12. Avatars (left, right) and the original image(center)

### 6. Conclusion

In the IGA-based system, the user evaluates the fitness of individuals. If there are a lot of individuals or the user must pass through a large number of generations until the optimal avatar is produced, the user encounters some fatigue with less convincing results.

Therefore, we address a new novel approach of IGA using hidden population which is applied to automated synthesis of 2D avatars. Three epoch-making techniques are proposed here which not only reduce user's fatigue but also attain the optimal result effectively and reliably. These are (a) hidden population, (b) primitive avatar, and (c) simplified genotype.

In the experimental results, it is demonstrated that users are significantly satisfied with automatically created avatars with less fatigue and higher reliability.

### Reference

[1] D.E.Golbber, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Massachusetts, Addison-Wesley, 1989.  
 [2] P.Bentley, *Evolutionary Design by Computers*, California, Morgan Kaufmann, 1999.  
 [3] H.Takagi, "Interactive Evolutionary Computation: Fusion of the Capabilities of EC Optimization and Human Evaluation", *Proc. of the IEEE*, vol: 89 Issue: 9, pp. 1275-1296, 2001  
 [4] M.Ohsaki and H.Takagi, "Improvement of Presenting

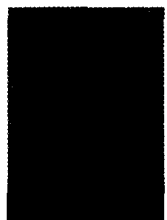
Interface by Predicting the Evaluation Order to Reduce the Burden of Human Interactive EC Operators", *IEEE Int'l Conf. on Systems, Man, and Cybernetics*, vol: 2, pp.1284 -1289, 1998.  
 [5] H.Takagi, K.Ohya, "Discrete fitness values for improving the human interface in an interactive GA", *IEEE 3rd Int'l Conf. on Evolutionary Computation*, pp.109-112, 1996.  
 [6] R.L.Haupt and S.E.Haupt, *Practical Genetic Algorithms*, New York, Wiley-Interscience, 1999.  
 [7] J.H. Lai, et al., "Robust Facial Feature Point Detection Under Nonlinear Illumination", *Proc. of IEEE ICCV Workshop*, pp. 168- 174, 2001  
 [8] J.D.Foley and A.v.Dam and S.K.Feiner and J.F.Hughes *Computer Graphics*, Addison Wesley, 1995  
 [9] R.Jain and R.Kasturi and B.G.Schunck, *Machine Vision*, Singapore, McGraw-Hill, 1995.  
 [10] Hee-Su Kim and Sung-Bae Cho, "Development of an IGA-based fashion design aid system with domain specific knowledge", *Proc. of IEEE SMC'99 Conf.*, vol: 3 pp.663 ?668, 1999  
 [11] H. Nishino and H. Takagi and K.Utsumiya, "Implementation and evaluation of an IEC-based 3D modeling system", *IEEE Int'l Conf. on Systems, Man, and Cybernetics*, vol: 2, pp.1047-1052, 2001



#### Ja-Yong Lee

received the B.S. degree in electrical and electronic engineering from Chung-Ang University, Seoul, Korea, in 2001. He is currently pursuing the M.S. degree. His research interests are Computer Graphics, Genetic Algorithm, Intelligent Control, A.I.

Phone : +82-2-816-8234  
 Fax : +82-2-816-1856  
 E-mail : jalnans@sirius.cie.cau.ac.kr



#### Jang-Hee Lee

received the B.S. degree in electrical and electronic engineering from Chung-Ang University, Seoul, Korea, in 2001. He is currently pursuing the M.S. degree. His research interests are Neural Networks, Vision Navigation Systems, Robotics and Intelligent Control.

Phone : +82-2-816-8234  
 Fax : +82-2-816-1856  
 E-mail : whizkid@sirius.cie.cau.ac.kr



**Hoon Kang**

received the B.S. and M.S. degrees in electronic engineering from Seoul National University, Seoul, Korea, in 1982 and 1984, respectively. He received the Ph.D. degree in electrical engineering in Georgia Institute of Technology in 1989. Since 1992, he has been a faculty member of School of Electrical and Electronic Engineering at Chung-Ang University. His research interests are Fuzzy Systems, Artificial Life, Cellular Automata, Neural Networks, Pattern Recognition, Evolutionary Computation, Vision System, Robotics and Intelligent Control.

Phone : +82-2-820-5320  
Fax : +82-2-816-1856  
E-mail : [hkang@cau.ac.kr](mailto:hkang@cau.ac.kr)