3D Face Modeling Support System for Avatar by Using Interactive Genetic Algorithm

FangWei Huang, Ivan Tanev, Kastunori Shimohara
Graduate School of Science and Engineering,
Doshisha University, Kyoto, Japan
Mailing address: 1-3 Tatara Miyakodani, Kyotanabe, Kyoto 610-0321, Japan
Email:fhuang2014@sil.doshisha.ac.jp, {itanev, kshimohara}@mail.doshisha.ac.jp

ABSTRACT

We propose a 3D face modeling system which can help 3D game users to create a unique favorite avatar. Generally, if we want to create a unique avatar in a 3D game, we need to adjust a huge number of parameters and we do not know what will happen until we see the avatar applied the change to. In reality, it costs a lot of time, and is difficult to get an avatar which user wants. To solve this problem, we employ Interactive Genetic Algorithm (IGA) for a user to find an avatar which the user favorites, introduce presentation population to improve the compactification and use idol face data to reduce the search space. The objective of this research is to help users to create a unique avatar easily, quickly and intuitionally.

KEYWORDS

Interactive genetic algorithm, 3D modeling, idol face data, immediate evolution, pre-evaluate

1. INTRODUCTION

In recent years, with the development of computer science, the graphics of games has achieved a notable advancement. Especially, 3D role playing games, in which gamers can control one avatar to do some adventures, become the most popular game type over the world.

With the popularization of 3D games, the competition between game makers becomes stronger and stronger. For their own survival, makers try to enhance users' attention by allowing users to create a unique avatar which can improve users' experience. Through the users' own unique avatar rather than the controlled 3D avatar, the users as gamers can feel as if they are doing those adventures. Such immersive visualization is one of the key factors for game makes to survival in the competition.

Generally, if users want to create an avatar as they like, they have to adjust a huge number of parameters without any direct mapping relationship to physical face. It means that users do not know which part will be changed, and how the part will be changed. What can users do is to adjust one parameter, compare, and then repeat. Due to the enormous search space of the problem, and the manual, tedious nature of the process of adjusting the parameters, finding the appropriate solution requires a lot of time.

This paper focuses on avatar modeling (3D face modeling) and proposes a new system to help users to create a unique avatar easily, quickly and intuitively via Interactive Genetic Algorithm (IGA). To achieve it, we have to address the following three challenges:

1) To design the genetic representation of the 3D avatar, i.e., to encode 3D avatar as a string of numbers,
2) To reduce the size of the search space of IGA, and

3) To reduce the amount of user inputs required for the interactive evaluation of the 3D avatars in IGA.

The remaining of this article is organized as follows. Section 2 presents the research framework. In Section 3 we discuss the genetic representation of 3D avatars. Sections 4 and 5 elaborate on the proposed approaches of reducing the search space of IGA and the amount of user interactions, respectively. Section 6 shows the experimental setup and Section 7 draws a conclusion.

2. RESEARCH FRAMEWORK

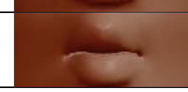
2.1 Interactive Genetic Algorithm

The research objective is to help users to get their favorite 3D avatar using an automated, evolutionary approach. Evolutionary computation paradigms, such as Genetic Algorithms (GA), IGA, and Genetic Programming (GP) need a fitness value that represents the quality of each of candidate solutions, and automatically evolve the candidate solutions based on this value. However, in many practical cases, it is impossible to formally define the fitness function either due to the lack of understanding of the underlying formal (e.g., physical or mathematical) models or, because the estimation of the quality of candidate solution is based on personal preferences of each of the end users of the evolutionary system. IGA offers a solution to these challenges by incorporating the user into evolutionary loop and requesting him to (subjectively) evaluate the quality of the evolved candidate solutions.

2.2 MakeHuman

To develop a system using IGA, we have to decode the evolved genotype (which is a string of numbers) into a 3D avatar. In our research, we use the *MakeHuman*[1] open source tool to do so. This tool allows us to modify a 3D model by means of modifying the values of a set of various numerical parameters. The value of each parameter is normalized within the range [-1…1]. Table 1 shows some sample values of the parameter *Width of the Mouth* and their corresponding decoded representations of the mouth of the 3D avatar.

Table 1. Encoding and decoding of the parameter *Width of the Mouth*

Width of mouth	Decoded representations of the mouth
-1	
-0.5	
0	
0.5	
1	

2.3 XML-based Genetic Programming

For an evolutionary methodology, we use the XML-based Genetic Programming tool (XGP) to manage the population of genetic representations of candidate solutions and the genetic operations on these solutions – selection, crossover, and mutations. It communicates with the fitness evaluator (*MakeHuman*) by sending the XML-representation of the genotype of candidate solution and receiving the fitness value via UDP channel [2].

3. RELATED WORK

3.1 Interactive Genetic Algorithm

In our research, we employ Interactive Genetic Algorithm (IGA) for a user to design his/her favorite avatar, because IGA provides users with a way to solve such problems on implicit human preferences and emotions [3] as just our case.

Considering the research objective mentioned above, we have to focus on two points: design using IGA and analysis of favorite face.

3.2 Design Using IGA

As a matter of fact, IGA has been used in many studies about design. According to them, by evolving well-encoded genes, we can design many things like Japanese yukata [4] and interior design [5]. Even we can design a 3D model by evolving its graph mesh [6] that is similar to our research.

Based on such achievements, we also employ IGA for users to evolve several 3D models to design their favorite avatar.

3.3 Analysis of KAWAII

KAWAII in Japanese means cute, and we focus on evolving the cute. A study by Nakayama team, for example, examined Japanese idols' eyes, nose, mouth, face shape and hairstyle [7]. In this research, we focus on the same parts of face except hairstyle because *MakeHuman* does not support to modify hairstyle.

4. GENETIC REPRESENTATION of 3D AVATARS

In *MakeHuman*, there are more than 100 features which we can modify various aspects of the presentation of 3D avatar. As the same as the research by Nakayama team [7], we focus on eyes, nose, mouth, and face shape, but we pay less attention to nose, because nose does little effect to the totally image of face. Considering a trade-off between the expressiveness of the genetic representation and the search space of IGA, in addition, we chose 28 features with which we can modify the parts of face mentioned above, and which are the most relevant features among all the features. The list of these most relevant features, used in the genetic representation of the avatar is elaborated in Table 2.

Table 2. The list of most relevant features used in genetic encoding of 3D avatar

Feature	Value range
head-vertically-scale	[-1, 1]
head-horizontally-scale	[-1, 1]
left-right-eye-inner-position	[-1, 1]
left-right-eye-outer-position	[-1, 1]
left-right-eye-scale	[-1, 1]
left-right-eye-inner-height	[-1, 1]
left-right-eye-outer-height	[-1, 1]
left-right-eye-middle-height	[-1, 1]
left-right-eye-inner-angle	[-1, 1]
left-right-eye-outer-angle	[-1, 1]
left-right-eye-horizontally-move	[-1, 1]
nose-vertically-position	[-1, 1]
nose-width	[-1, 1]
mouth-vertically-position	[-1, 1]
upper-lip-height	[-1, 1]
lower-lip-height	[-1, 1]
mouth-vertically-corner	[-1, 1]
chin-angular-scale	[-1, 1]
eyebrows-angle	[-1, 1]

5. REDUCTION OF SEARCH SPACE

5.1 Gene Value Range

In *MakeHuman*, all the parameters are floating point numbers, so for each parameter, both the (i) ranges of possible values and (ii) number of combinations of values of parameters are unlimited. An eventual direct encoding of these parameters in the genotype would result in the following problems:

1) The search space of GA would be unbounded.
2) Eventual a too small change in the value of parameters in genotype (e.g., from 0.001 to 0.002) might not result in any observable change. This, in turn implies that the precious runtime is wasted for producing (via crossover and mutation) and evaluating such slightly different (but phenotypically identical) genotypes.

Therefore, in our research, we propose a range of the genes between [-10…10]. The values from this range are mapped into the range of real numbers [-1…1] via floating point division by 10. These real numbers are then fed to *MakeHuman* for rendering the corresponding 3D avatar.

5.2 Bilateral Symmetry Features

We are aiming to help users to create an avatar in 3D game. Generally, people like rather a beautiful avatar than a "special" avatar. So we focus on creating a beautiful avatar.

The first factor for a beautiful avatar is bilateral symmetry. People would not think a face with two different eyes a beautiful face. To keep bilateral symmetry, we just evolve the features of the left eye, and set the values of the features of the right eye to be identical to those of the left one.

5.3 Incremental Evolution

Generally, in evolutionary computation like IGA, GA and GP, the initial population of candidate solutions is created randomly. However, if we initialize the facial features of the avatar randomly, we would generate an initial population of rather unpleasantly looking avatars. An example of such an avatar is shown in Figure 1.

Figure 1. Avatar with all 27 facial features set randomly

It would be possible to incrementally evolve such (randomly created) avatar; however, such an evolution would require a significant amount of runtime. In order to improve the efficiency of IGA, we propose two-phased evolution, in that we first decide the overall features of the avatar, and then evolve the details. Two-phased evolution is intended to split the whole evolution into two phases so that the 2nd phase will keep all the values of the features, evolved at the 1st phase intact. In other words, we try to introduce a seed feature – e.g., eye width – that is most influential regarding the other features according to some relationship. At the 1st phase, we continue evolving the seed feature and its relationship to other parameters until we can get an avatar with a good totally image. At the 2nd phase, we keep the result of the 1st phase and evolve other independent features left.

This approach can help us to reduce search space significantly. For example, if we have 8 features totally, we evolve 4 of them in each phase. As a result, the size of search space would be reduced from 20^8 to just 20^4+20^4, or 80,000 times lower.

5.4 Idol Face Data

As mentioned above, when we introduce the incremental evolution with two phases, we need some data on relationship between the seed feature and other facial features.

We utilize some data for Japanese idol face from a research about the mathematical approach to the cute [8]. Concretely, we use the data for the 1st phase evolution. Those data contain 7 kind of relationship of two features, and the detail is shown as Table 3, and Figure 2 and 3. Especially in Figure 2 and 3, X axis shows the ratio of the two features and a Y axis shows the number of idols among totally 57 idols.

Table 3. Relationship of features

Feature	Abbrev.
Hairline to nose apex : nose to chin	ABBD
Mouth center to chin : nose apex to mouth center	CDBC
Face height : face width	FHFW
Eye distance : nose width	EDNW
Eye distance : eye width	EDEW
Eye width : eye height	EWEH
Face width : eye width	FWEW

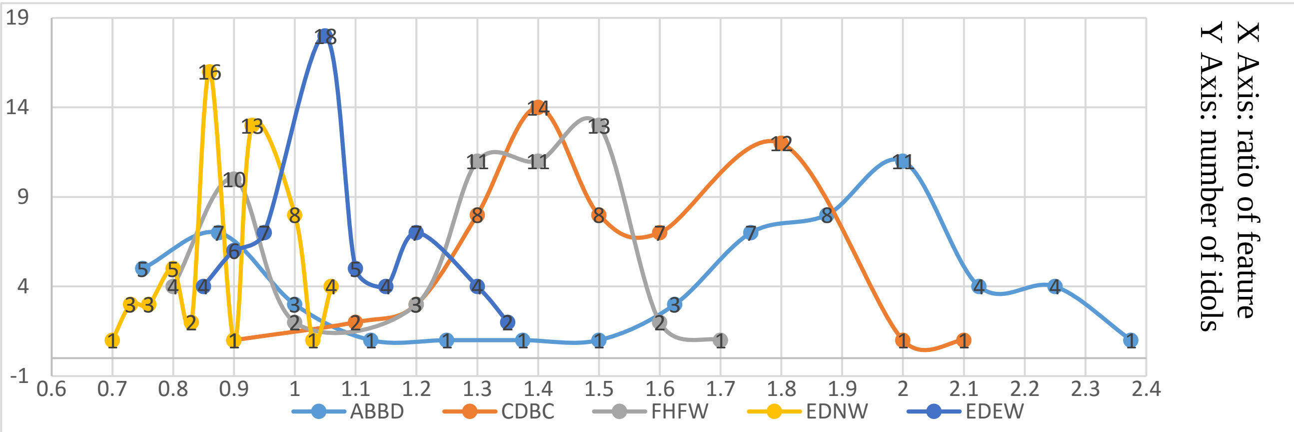


Figure 2. Idol face data #1

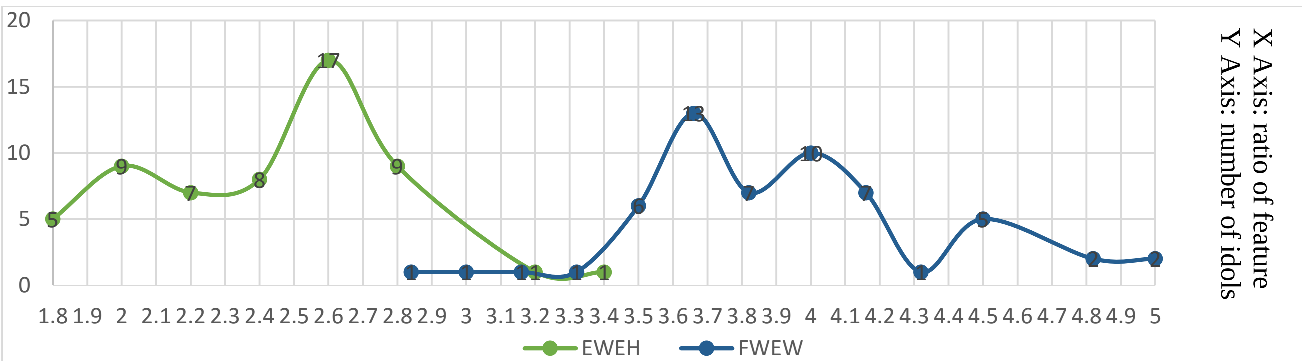


Figure 3. Idol face data #2

For example, EWEH in Figure 3 shows that there are 9 idols whose eyes' width is two times of its height.

5.5 How to Use Idol Face Data

From Figure 2 and 3, we can know that, for each relationship, different ratio has different weight. We propose a proportional, roulette-wheel selection to represent the characteristics in this system.

6. REDUCING the AMOUNT of USER INTERACTIONS

6.1 Presentation Population

Generally, IGA has a compactification trade-off problem, because all the genetic operations such as selection, crossover and mutation are random, and all of evaluation is done by users. So, the larger population size makes better compactification, but users have to find their favorite avatar from a huge number of samples, and it may result in users' heavy fatigue. And vice versa, the small population size makes less fatigue, but worse compactification means that users have to click many times to get the best avatar. This trade-off problem is the main task on IGA-based applications [9].

To solve this trade-off problem, we propose to add a middle layer, which we call presentation population, to mother population. The proposed presentation population is one sampled from mother population, and its individuals are rendered on GUI for user's selection. At the generation 0, it is sampled at random. But from generation 1, individuals, which are more similar to a user's selection in pre-generation, should a higher possibility to be selected as presentation individuals, so we propose pre-evaluation to calculate their pre-fitness values. Using those values, we can sample presentation population in binary tournament and make sure that the individuals rendered on GUI have high fitness and variety. In this way, we can expect that our IGA can keep in a high performance, by reducing user's interactions in each generation with small population size compared to the big size of the mother population.

6.2 Pre-evaluation

As mentioned above, we use presentation population to make sure that the presentation individuals have high fitness and variety reducing the amount of user interaction. To achieve it, we proposed pre-evaluation to calculate pre-fitness.

In this research, pre-fitness means the similarity to user's selection in pre-generation. For example, if we want to create presentation population for the generation 1, then we use user's selection in the generation 0 to calculate the pre-fitness value of the mother population in generation 1, at last, we do selection based on the pre-fitness value. The flow of pre-evaluation is shown in Figure 4.

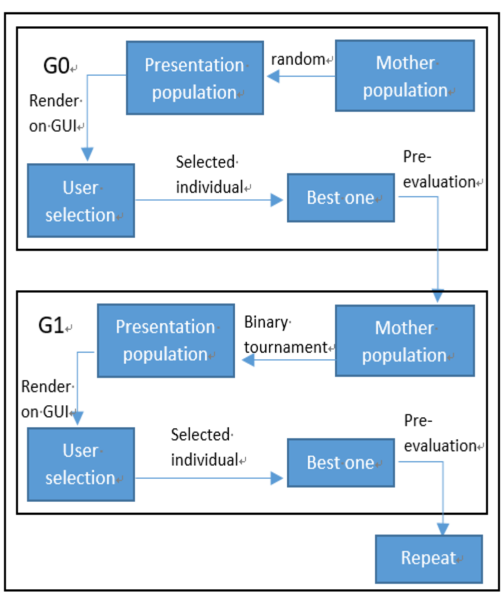


Figure 4. The flow of pre-evaluation

The reason why we can do this is because the offspring (generated by crossover and mutation) in evolutionary computing are similar to their parents; consequently, using their parents in previous generation, we can obtain the best pre-fitness value [10].

6.3 Result of Pre-evaluation and Presentation Population Simulation

To confirm how this approach works, we did a simulation. The result is shown as Figure 5:

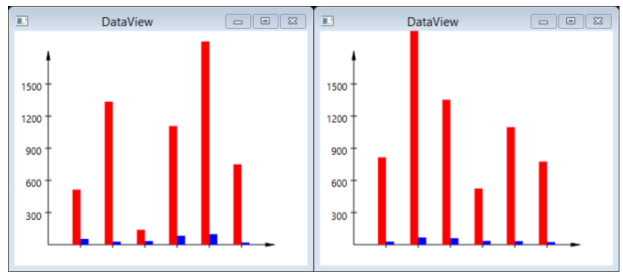


Figure 5. Simulation results

These graphs show the result of a simulation with 6 runs using the same initialized mother population. In these graphs, the ordinate shows the number of generations. The red bars show the number of generations, needed to be evolved in the canonical IGA before obtaining a desired 3D avatar. The blue bars show the number of generations needed to be evolved via the proposed IGA. Thus, we can confirm experimentally that using pre-evaluation and presentation population can reduce the total number of generations significantly, which, in turn, results in a better compactification.

7. EVOLVING 3D FACE VIA IGA

7.1 System Structure

The system consists of four modules: modeling module, GUI, control module and evolving module. In them, modeling module, control module and evolving module communicate with each other by UDP or TCP. The image of system structure is shown in Figure 6:

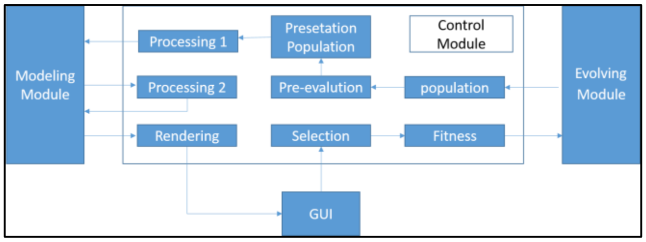


Figure 6. System structure

We use TCP for the communication between the control module and the modeling module because we need to send or receive big data (e.g., pictures).

In the processing 1 as shown in Figure 6, the system picks up the features which are used to modify eye, and send them to the modeling module to get eye width. In the processing 2, the system uses the eye width, decides the left genes representing the relationship between the eye width and other parts to calculate the parameters to modify avatar, and sends them to the modeling module.

7.2 Parameters of Evolution

The parameters of the evolutionary framework are shown in Table 4:

Table 4. Parameters of the evolutionary framework

Parameter	Value
Selection type	Binary tournament
Probability of selection	20%
Crossover type	One point
Probability of Crossover	80%
Mutation type	Single point, random subtree
Probability of mutation	Dynamically adjustable between 20% and 60%
Population size	40
Presentation population size	4

7.3 Phases of Incremental Evolution

In this research, we proposed incremental evolution and split the evolution into two phases. The features which have relationship with others are as shown in Table 5.

Table 5. Evolution parameters in 1st phase

Feature	Value range
head-vertically-scale	[-1, 1]
head-horizontally-scale	[-1, 1]
left- eye-inner-position	[-1, 1]
left- eye-outer-position	[-1, 1]
left- eye-middle-height	[-1, 1]
left- eye-scale	[-1, 1]
left-eye-horizontally-move	[-1, 1]
nose-vertically-move	[-1, 1]
mouth- vertically-move	[-1, 1]

From these features, we use the following three features as the seed features: left-eye-inner-position, left- eye-outer-position and left-eye-scale (the right eye should similar to the left without evolution), because we find the most important measuring features (can be measured in centimeter, like width of eye, distance between eyes) from the idol face data in Figure 2 and 3. It means that the other features in Table 5 can be calculated by using three features on the idol face data. Figure 7 shows the relationship tree between them.

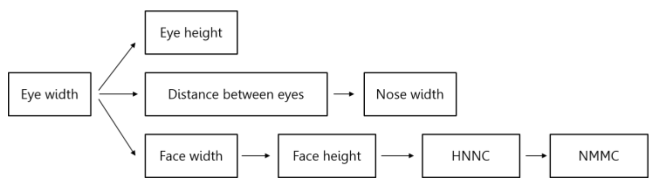


Figure 7. System structure

Totally, in the 1st phase, we need to evolve three features and the relationships. The corresponding genotype is shown in Table 6:

Table 6. Gene for 1st phase

Gene	Genotype Range	Value
left- eye-inner-position	[-10, 10]	[-1, 1]
left- eye-outer-position	[-10, 10]	[-1, 1]
left- eye-scale	[-10, 10]	[-1, 1]
ABBD	[-10, 10]	[0.75, 2.375]
CDBC	[-10, 10]	[0.9, 2.1]
FHFW	[-10, 10]	[0.8, 1.7]
EDNW	[-10, 10]	[0.7, 1.06]
EDEW	[-10, 10]	[0.85, 1.35]
EWEH	[-10, 10]	[1.8, 3.4]
FWEW	[-10, 10]	[2.84, 5]

The chromosome is be encoded using integer numbers which have the value in the range as shown in Table 6 (e.g. [-1, 8, -5, 5, -9, 9, 3, 2, 6, -7] for each gene respectively). Thus, after using the incremental evolution, the size of search space of this system becomes $20^{10}+20^{10}$, which is significantly smaller than 20^{19} in the case without the use of incremental evolution.

7.4 Mapping Table

In the 1st phase, we need to get parameters based on the calculated measuring features, for example, a parameter which can makes nose width as 2.8cm. By a reason that *MakeHuman* does not support this function, we create some mapping table to achieve it. Those tables were created by the sampling parameters and recording both parameters and measuring feature values as Figure 8:

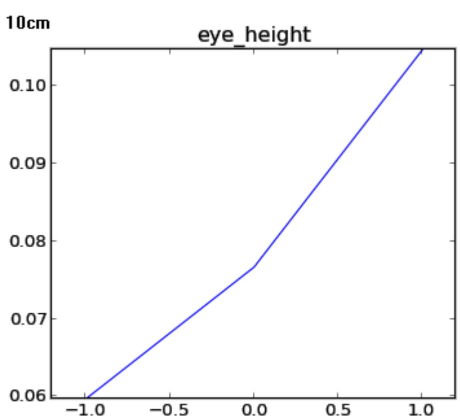


Figure 8. Mapping table

7.5 Evaluation and Pre-evaluation

Is our system, we need to evaluate fitness values for all individuals at every generation. We use Euclidean distance as fitness and pre-fitness.

$$\mathrm{F}_g = \sqrt{\sum_{n=1}^{N} \left(indiv_n - sel_g\right)^2} + 10 \qquad (1)$$

$$\mathrm{PF}_g = \sqrt{\sum_{n=1}^{N} \left(indiv_n - sel_{g-1}\right)^2} + 10 \qquad (2)$$

In them, F is fitness value, PF is pre-fitness value, g is the current generation, N is the population size, *indiv* means the individual in mother population, and *sel* means user’s selection.

7.6 XGP for IGA

We customize XGP to adapt IGA, call it IXGP, and use it as the evolution module. IXGP sends a string of numbers which contain genes of all the individuals, and receives a string of numbers contain fitness values of all the individuals.

7.7 System Flow

When the system starts, it creates the initial population for XGP, creates presentation population at random, and then modifies 3D avatars and renders on GUI as shown in the left side of Figure 9. If the user clicks one of the 4 avatars, this one will used as the best result in this generation to calculate fitness value and pre-fitness value for the next generation. Of course, the user can shuffle and create a new presentation population if all of them are not good.

After user’s selection, the calculated fitness values will be sent back to XGP. The latter uses the fitness to bias the following selection and crossover operations. This process repeats until the user obtains a good result (XGP will terminate on achieving this result) as shown in the right side of Figure 9.

Table 9. System GUI image

ISBN: 978-1-941968-17-8 ©2015 SDIWC

8. CONCLUSION

We have presented an avatar creation system to support uses' avatar creation via IGA. Pursuing three challenges: encoding avatar, reducing the search space and improving compactification, we have proposed approaches for them or used some tools to solve them. The most difficult task is to create the mapping table, because there is not any API to use. So, we have to find some way to measure the size of every parts of 3D model.

Right now, this system is able to evolve and obtain a satisfactory avatar. From now, we will continue completing the 1st phase and add the 2nd phase, and then we are planning conducting experiments aimed at verifying the results of the integration of all of the proposed approaches.

REFERENCES

[1] MakeHuman–open source tool for making 3D characters, URL: http://www.makehuman.org/, (accessed 2015-7-6).

[2] I. Tanev and K. Shimohara, "XML-based Genetic Programming Framework: Design Philosophy, Implementation, and Applications", Artificial Life and Robotics, Vol.15, No.4, pp.376-380, 2010.

[3] S. Ono, H. Maeda, K. Sakimoto, S. Nakayama, "User-System Cooperative Evolutionary Computation for both Quantitative and Qualitative Objective Optimization in Image Processing Filter Design", Applied Soft Computing, Vol.15, No.2, pp.203-218, 2014.

[4] M. Sugahara, M. Miki and T. Hiroyasu, "Design of Japanese Kimono using Interactive Genetic Algorithm", Systems, Man and Cybernetics, 2008. SMC 2008. IEEE International Conference, pp.185-190, 2008.

[5] A. Nishioka, "Proposal of Personal Sentiency---Personal Model Searching System for Sentiency", Undergratuate Thesis, Doshisha University, 2007.

[6] H. Nishino, T. Sueyoshi, T. Kagawa and K. Utsumiya, "An Interactive 3D Graphics Modeler Based on Simulated Human Immune System", Journal of Multimedia, Vol.3, No.3, pp.51-60, 2008.

[7] K. Nakayama, T. Hashimoto, Y. Noridomi and C. Oshima, "Analysis of Users' Favorite Faces", Proceedings of the 8th International Conference on Humanized Systems, pp.46-51, 2012.

[8] K. Wakamatsu and K. Maekawa, "Mathematical apporach to KAWAII", Special Interest Group on Mathematical Science, Shibaura Inst. of Technology, 2012.

[9] H. Takagi, T. Unemi and T. Terano, "Perspective on Interactive Evolutionary Computing", The Japan Society for Artificial Intelligence, Vol.13, No.5, pp.692-703, 1998.

[10] M. Ohsaki and H. Takagi, "Reduction of the Fatigue of Human Interactive EC Operators", The Japan Society for Artificial Intelligence, Vol.13, No.5, pp.712-719, 1998.