Design facial appearance for roles in video games

Shang Hwa Hsu *, Ching-Han Kao, Muh-Cherng Wu

Department of Industrial Engineering and Management, National Chiao Tung University, 1001 Ta Hsueh Road, HsinChu 30010, Taiwan, ROC

A R T I C L E   I N F O

Keywords:
Video game
Role
Fuzzy AHP
BP neural network

A B S T R A C T

Roles in video games often serve as avatars of players. Different game players may have their particular preferences on a role’s facial appearance. It would be desirable to allow players to customize the design of roles. This paper presents two methods for recommending a role’s facial appearance for a particular game player and illustrates the two methods by using heroic roles as an example. The two recommendation methods are designated as the text-input and the picture-input approaches. The text-input approach requests the game player to carry out pairwise comparisons for determining the relative weights of 16 personality traits of heroes. The recommendation mechanism for the text-input approach is based on the fuzzy AHP (analytic hierarchy process). Whereas the picture-input approach requests the game player to view a sample set of pictures and rate his/her preferences on each picture. The recommendation mechanism for the picture-input approach is based on the BP (back-propagation) neural network. Experiments indicated that the text-input approach is more effective in terms of recommending an appropriate facial appearance, yet at the expense of needing more user time.

© 2008 Elsevier Ltd. All rights reserved.

1. Introduction

In playing a video game (called a game hereafter), a player expresses his or her intentions by manipulating the actions of a role. A role is a character, which serves as the player’s avatar or competitor. Roles to a game are as important as actors to a film. Casting appropriate actors can lead to the success of a film. In the same way, designing appropriate roles is very important to the success of a video game.

Previous researchers have published numerous studies on the role design of video games. Most of them attempted to create a life-like role by emulating human behaviors—such as dialogue (Brusk & Eladhari, 2006; Gustafson, Boye, Fredriksson, Johanneson, & Königsmann, 2005; Jan & Traum, 2005), intelligence (Frasca, 2001; Lair & Duchi, 2000; Vala, Paiva, & Prada, 2004), motor-skill actions (Blumberg & Galyean, 1997), and emotional expressions (Rizzo, Neumann, Enciso, Fidaleo, & Noh, 2001; Wallraven, Breidt, Cunningham, & Bülthoff, 2005).

Of these studies, those which analyze emotional expressions attempt to automatically create a facial expression to represent a character’s mood (e.g. happy, angry, sad, and disgusted). Such facial expression studies have an implicit objective—mood manifestation (Bartlett, Hager, Ekman, & Sejnowski, 1999; Ekman, 1993; Zhang & Ji, 2005). That is, a computer-generated facial expression should model a particular mood (e.g. sad) that is easily recognizable by humans.

In a film, proper recognition of an actors’ mood by interpreting their body languages is never enough. To produce a popular film, the attractive appearance of actors is often much more important. Likewise, the appearance of a game role may influence player involvement in the game. An attractive appearance may induce players to have affectionate feelings for the avatar, and in turn, make game play more fun (Hsu, Lee, & Wu, 2005). Even though the facial appearance of a role is very important to game design, this topic has rarely been investigated.

This study proposes two methods for automatically recommending attractive facial appearances of heroic roles to game players—in a customization manner. The observations in this study indicate that different game players have different preferences for the appearance of a hero. According to further interviews, this difference is due to the fact that players have different preferences for a hero’s personality traits. This implies that personality traits may be manifested by facial appearances.

Based on this implication, this study proposes a research framework to design a hero’s facial appearance as favored by a particular game player (Fig. 1). This research framework involves two phases: creation and application. The creation phase develops a database that relates facial appearance to personality traits, which involves two steps. Firstly, multimedia software is used to create samples of heroes’ facial appearances. Secondly, game players evaluate the personality traits for each facial appearance. In this phase, a vector comprising numeric data can model both a facial appearance and its personality traits. The created database is called a face-trait database.

The application phase implements the customization design of the heroes’ facial appearance. That is, for any given game player,
a highly favored heroes’ facial appearance can be quickly recom-
mended from the face-trait database. Two recommendation mech-
nisms can be used to retrieve these highly favored facial
appearance profiles.

One mechanism is a text-input approach, which requires the
game player to determine each personality trait’s relative weight.
This mechanism uses the fuzzy AHP technique (Laarhoven & Ped-
rycz, 1983; Saaty, 1980; Shamsuzzaman, 2000; Zadeh, 1975). The
other mechanism is a picture-input approach, which requires the
game player to evaluate their degree of preferences for a sample
set of pictures (facial appearances). This mechanism is imple-
mented using the back-propagation (BP) neural network
technique.

The remainder of this paper is organized as follows. Section 2
describes how the face-trait database is created. Section 3 first
introduces the fuzzy AHP technique, and then presents the text-
input recommendation mechanism. Section 4 describes the
picture-input recommendation mechanism. Section 5 presents
the experiments and results for justifying the effectiveness of the
two recommendation mechanisms. The last section contains
concluding remarks.

2. Face-trait database

A prototype face-trait database has been developed. In develop-
ing the database, heroes’ facial appearances were created by com-
ercially available multimedia software (Live Studio Head Tool
V2.6). We use 16 personality traits of heroes to characterize each fa-
cial appearance created. A hero’s facial appearance could then be
encoded by a vector consisting of 16 elements, which provides a
key for the recommendation facial appearances.

2.1. Facial appearance creation

The multimedia software used to create a facial appearance is
based on a feature-based approach. That is, the configuration of a
face is composed of several features. A feature may denote a part
of a face such as eyes, lips, and nose or denote a face’s characteristic
such as skin color and texture. For each feature, there are many op-
tions for selection. Various combinations of feature options result
in different facial appearances.

For features associated with geometric shapes, their feature op-
tions are created by the parametric-geometry approach (Myung &
Han, 2001; Verroust, Schonek, & Roller, 1992). That is, changing
some of its geometric parameters can vary the shape of an object.
Considering an object with rectangular shape as an example, we
could take the length-to-width ratio as one parameter and vary
shape of the object by changing the parameter value. Likewise, a
facial feature such as eyes can be modeled by a parametric-geom-
etry shape with more than one parameter; for example, the curva-
ture of upper eye lid and that of lower eye lid. By varying these two
parameters, we could have many different shapes for modeling
eyes.

Example features associated with geometric shapes include lips,
noses, eyes, eyebrows, and facial outline. These shapes are com-
monly controlled by a set of parameters, and each parameter value
is a real number in a predefined range such as [0,1]. The input of such a real-number value is through a scale-bar input device (rightmost part of Fig. 2).

Due to the completeness property of real numbers, there are theoretically an infinite number of options available for features associated with geometric types. By contrast, some other features provide only a finite number of feature options; for example—type of hair and type of moustache. The input of such a nominal-type feature option is through the clicking of a menu box (middle part of Fig. 2).

The multimedia software we used provides 12 features. In this research, only five shape-oriented features are chosen as variables in designing a facial appearance for the prototype database. These five features are facial outline, eye brows, eyes, lip, and nose, which are selected because most prior research has concluded their importance on facial expression and recognition (Adolphs, 2002; Brunelli & Poggio, 1993; Sadrô, Jarudi, & Sinha, 2003).

The prototype database which is essentially expandable, at its present state, includes 243 different facial appearances. That is, each of the five facial features has only three options for selection, which leads to $3^5 = 243$ different facial appearances. Three instances of the facial appearances are shown in Fig. 3.

2.2. Personality traits evaluation

We use 16 personality traits to characterize each facial appearance. These personality traits are a hero’s characters reported in a prior research by Hsu, Kao, and Wu (2007). According to their research, by the technique of factor analysis (principal components factoring) (McDonald, 1985), these 16 personality traits can be categorized into three groups—bravery, visionary, and moral as shown in Fig. 4.

In the characterization of a facial appearance, a five-point scale evaluates each personality trait. The higher the value, the higher degree does the facial appearance reveal—in terms of the personality trait. In the characterization process, we asked 112 subjects (61 males, 51 females at the age of 17–25) to evaluate the 16 personality traits for each of 243 facial appearances. All these subjects are all game player, who are either senior high school or college students.

Results of the characterization process yield 243 trait vectors, each of which represents a particular facial appearance. The value of each element, ranging from 1 to 5, is the mean score reported by all the subjects. Let $X = [x_1, \ldots, x_{16}]$ denote a trait vector so obtained, which is further transformed into a normalized vector, denoted by $Y = [y_1, \ldots, y_{16}]$ where $y_i = \frac{x_i}{\sum_{j=1}^{16} x_j}$ and $\sum_{j=1}^{16} y_j = 1$.

3. Text-input recommendation mechanism

To create a hero face favored by a particular game player, it is important to know how the player weights each personality trait. The procedure for determining the weights of personality traits is by applying the fuzzy AHP technique (Saaty, 1990). The technique, widely applied in various areas (Durán & Aguilo, 2007; Kim & Yoon, 1992; Mamaghani, 2002; Muralidar & Santhanam, 1990; Wu, Lo, & Hsu, 2007) is briefly stated below.

Firstly, the 16 personality traits are hierarchically clustered as shown in Fig. 4. This hierarchy indicates that these 16 personality traits, based on a prior research (Hsu et al., 2007), can be categorized into three groups—bravery, visionary, and moral, which are called group-level traits. Each personality traits within a group is called a member-level trait.

Secondly, the relative weights for group-level traits and that for member-level traits in each group have to be determined. We therefore have four weight-determination problems, one for group-level and three for member-level traits. To each weight-determination problem, we asked the game player to carry out a pairwise comparison experiment, and use a fuzzy AHP algorithm to process the experiment data for determining the relative weights.

Thirdly, we use the relative weights to compute a recommendation-priority value to retrieve a hero face from the face-trait database for the game player.

3.1. Pairwise comparison experiment

The pairwise comparison experiment is explained by using the weight-determination of group-level traits as an example. Of the three group-level traits, any two (a pair) is chosen for comparison. Assume the pair (bravery, visionary) is chosen. Then, the player is asked to answer the following question: “Consider a person to be recognized as a hero. Of the two groups of personality traits, which one is more important? Try to compare the degree of importance between them.”

Now suppose the player answers bravesty is more important than visionary. Then, five linguistic variables listed in Table 1 are provided for the player to express his/her opinion. These linguistic variables range from “absolutely important” to “equally important” on a five level scales, where between any two consecutive

![Fig. 3. Example of facial appearances.](image)

![Fig. 4. Three levels fuzzy AHP hierarchy model.](image)
scales an intermediate scale is additionally defined so that nine scales are finally created. As shown in Table 1, each linguistic variable is represented by a triangular fuzzy number—for example, \( \tilde{A} = [\tilde{a}_{ij}] \) could be obtained, where \( n \) denotes the number of traits to be compared,

\[
\tilde{A} = \begin{bmatrix}
\tilde{a}_{12} & \cdots & \tilde{a}_{1n} \\
\tilde{a}_{21} & 1 & \cdots & \tilde{a}_{2n} \\
\vdots & \ddots & \ddots & \vdots \\
\tilde{a}_{n1} & \cdots & \tilde{a}_{nn} & 1
\end{bmatrix}
\]

Notice that \( \tilde{a}_{ij} = 1/\tilde{a}_{ji} \), which ensures that the comparison for each pair \((i, j)\) is consistent; and \( \tilde{a}_{ii} = 1 \), which denotes that a self-comparison is always “equally important” and is not needed.

### 3.2. Fuzzy AHP algorithm

Two procedures are used to obtain the relative weights for group-level and member-level traits. The first one Compute_{Relative_Weight} is intended to compute the relative weights, and the second one Validity_{Check for Relative_Weights} is developed for checking the validity of the obtained relative weights. These two procedures are respectively described below, where the definitions of arithmetic operators (i.e., \( \odot \), \( \oplus \), and Defuzzy) are introduced in the Appendix.

#### 3.2.1. Procedure Compute_{Relative_Weight}

Step 1: Calculate \( W_i \), the fuzzy weight for each row \( i \) in \( \tilde{A} \) (Buckley, 1985)

\[
\tilde{Z}_i = (\tilde{a}_{11} \odot \tilde{a}_{12} \odot \cdots \odot \tilde{a}_{1n})^\frac{1}{n}, \quad \forall i = 1, \ldots, n
\]

\[
\tilde{W}_i = \tilde{Z}_i \oplus (\tilde{Z}_1 \oplus \tilde{Z}_2 \oplus \cdots \oplus \tilde{Z}_n)^{-1}, \quad \forall i = 1, \ldots, n
\]

Step 2: Defuzzication of \( \tilde{W}_i \) and \( \tilde{A} \) (Teng & Tzeng, 1993)

\[
W_i = \text{Defuzzy}(\tilde{W}_i)
\]

\[
a_{ij} = \text{Defuzzy}(\tilde{a}_{ij}), \quad \text{where } \tilde{A} = [a_{ij}] \text{ and } \tilde{A} = [\tilde{a}_{ij}]
\]

Step 3: Compute relative weights \( W[i] \)

\[
W_i = \frac{W_i}{\sum_{i=1}^{n} W_i}
\]

#### 3.2.2. Procedure Validity_{Check for Relative_Weights}

Step 1: Compute \( W_i \)

\[
\begin{bmatrix}
W_1 \\
W_2 \\
\vdots \\
W_n
\end{bmatrix} = A
\begin{bmatrix}
W_1 \\
W_2 \\
\vdots \\
W_n
\end{bmatrix}
\]

Step 2: Compute \( \lambda_{\text{max}} \), the maximum eigenvalue

\[
\lambda_{\text{max}} = \frac{1}{n} \left( \frac{W_1}{W_1} + \frac{W_2}{W_2} + \cdots + \frac{W_n}{W_n} \right)
\]

Step 3: Compute CI, the consistency index

\[
CI = \frac{\lambda_{\text{max}} - n}{n - 1}
\]

Step 4: Compute CR

\[
CR = CI/RI \text{ the values of RI are shown in Table 2 (Saaty, 1980).}
\]

Step 5: Consistency check

If \( CR < 0.1 \), pairwise comparison data \( \tilde{A} \) is reasonably consistent.

Output the resulting relative weights \( W_i \) (\( 1 \leq i \leq n \))

Else (\( CR > 0.1 \)) pairwise comparison data \( \tilde{A} \) are inconsistent.

Repeat the pairwise comparison experiment.

Endif

### 3.3. Recommending mechanism

To recommend a hero face favored by a particular game player, we first compute a recommendation-priority value for each facial appearance in the face-trait database and recommend the one with the highest recommendation-priority value.

Define \( S = \{ (Y_i, F_j) \} \) as the face-trait database developed for facial design, where \( F_i \) denotes the \( i \)th facial appearance, and \( Y = [y_{ij}] \), \( 1 \leq i \leq 16 \) denotes the personality-trait vector of \( F_i \). Notice that \( F_i \) is a 2D picture while \( Y_i \) is a vector with 16 numeric elements, where \( y_{ij} \) denotes the degree of \( j \)th personality trait that picture \( F_i \) manifest to a “common people”. Let \( W = [w_j] \), \( 1 \leq j \leq 16 \) represent the preferences of a particular game player on each of the 16 personality traits, where \( w_j \) denotes the relative weight of \( j \)th personality trait, perceived by the game player—an “individual people” rather than a “common people”.

The procedure for recommending facial appearances most favorable to a particular game player proceeds as follows.

Step 1: Compute the recommendation-priority value

\[
p_i = \sum_{j=1}^{16} w_j \cdot y_{ij}, \quad 1 \leq i \leq n
\]

Step 2: Output the recommended one

\[
i^* = \arg \max(p_i)
\]

### 4. Picture-input recommendation mechanism

The BP neural network technique has been widely used as a predictor (Dutta & Shekhar, 1988; Liu & Chen, 2005; O’Leary, 1998; Salchenberger, Cinar, & Lash, 1992; Tam & Kiang, 1992). Given a sample set of input/output data obtained from a real-world system, we can use the technique to establish a BP network that serves as an input/output mapping mechanism—a three-layer architecture as shown in Fig. 5. The established BP network can be used to predict the output for a new input data set. Details of the BP neural network technique can be referred to Wasserman (1989) and Hertz, Krogh, and Palmer (1991).

The picture-input recommendation mechanism is to establish a BP network for a particular game player. For such a BP network, its

<table>
<thead>
<tr>
<th>Table 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linguistic variables used in the fuzzy AHP</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fuzzy number</th>
<th>Linguistic variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 – (1.1, 1)</td>
<td>Equally important</td>
</tr>
<tr>
<td>2 – (3.4, 3)</td>
<td>Weakly important</td>
</tr>
<tr>
<td>3 – (4.5, 6)</td>
<td>Essentially important</td>
</tr>
<tr>
<td>4 – (5.7, 8)</td>
<td>Very strongly important</td>
</tr>
<tr>
<td>5 – (6.9, 10)</td>
<td>Absolutely important</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Values of RI</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>n</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>RI</td>
<td>0.00</td>
<td>0.00</td>
<td>0.58</td>
<td>0.90</td>
<td>1.12</td>
<td>1.24</td>
<td>1.32</td>
<td>1.41</td>
<td>1.45</td>
<td>1.49</td>
<td>1.51</td>
</tr>
</tbody>
</table>
input is a facial appearance denoted by \( F_i = [f_{ik}] \), \( 1 \leq i \leq 243 \), \( 1 \leq k \leq 15 \), where \( f_{ik} \) (a binary number) denotes the existence of the \( k \)th feature option in the \( i \)th facial appearance. As stated, a facial appearance has five features, each of which has three options for selection. This leads to \( 15 \times 3 = 45 \) types of feature options. A facial appearance can then be modeled by a vector with 15 binary elements, in which 1 denotes a feature option exists and 0 denotes its nonexistence. The output of such a BP network is denoted by \( P_i \), which denotes the game player’s preference on the \( i \)th facial appearance.

The use of the BP neural network technique involves two phases: training and predicting phases. In the training phase, we attempt to establish a BP network for the game player. Out of the 243 facial appearances, we randomly sample 81 ones and use them to build up a BP network. The algorithms of the training process can be referred to Grossberg (1974) and Rumelhart, Hinton, and Williams (1986). In the predicting phase, we attempt to use the established BP network to predict the player’s preferences on the other facial appearances not considered in training phase. That is, the BP network will predict the player’s preference on the remaining 162 facial appearances.

5. Experiments

An experiment is carried out to compare the performance of the text-input and the picture-input approaches. Twenty game players (10 males, 10 females at the age of 17–25) are invited as experiment subjects, and 243 facial appearances are created in the face-trait database. The experiment proceeds as follows.

Firstly, we carried out the text-input approach in which each subject \( i \) is requested to perform a pairwise comparison for determining his/her relative weights on the 16 personality traits. The system will output a picture (say, \( X_i \)) with the highest recommendation-priority value.

Secondly, we carried out the picture-input approach in which each subject \( i \) is requested to view 81 pictures and give his/her preference on each picture. The input/output data of the 81 pictures are used to establish a BP network. Then, all the 243 pictures are fed into the BP network, and the one (say, \( Y_i \)) with the highest preference value will be recommended.

Finally, each subject is requested to view the remaining 162 pictures and give his/her preference on each one. Out of the 243 pictures, the one with the highest preference (say, \( Z_i \)) is selected.

Define \( r(P_i) \) as the preference of picture \( P \) evaluated by subject \( i \). The effectiveness of the text-input approach is measured by the distribution of the metric: \( x_i = r(Z_i) - r(Y_i) \), and that for the picture-input approach is measured by the distribution of \( y_i = r(Z_i) - r(Y_i) \).

The distributions of \( x_i \) and \( y_i \) are shown in Table 3. The table indicates that the text-input approach is superior to the picture-input approach. In the text-input approach, 85% game players (17 out of the 20 subjects) will get their most favor picture, while only 35% can get so in the picture-input approach. However, the text-input approach needs more user time. The average time for performing a pairwise comparison is about 26 min. and that for evaluating 81 pictures is about 2.6 min.

The experiment results lead to the following implication. For a video game equipped with relatively few numbers of pictures, we would suggest an exhaustive display of pictures to a game player. In contrast, for a video game equipped with a great amount of pictures, we would suggest the use of the text-input approach.

6. Conclusions

Playing video games through manipulating a role is quite common. Role design therefore has been a significant research issue. Most prior research attempted to create a life-like role by develop software for emulating human’s capabilities, such as dialogues, intelligence, motor-skill, and emotional expressions. This research is unique in providing a customized facial appearance for each game player. A video game with such a customization function would become more popular.

Two methods for providing such a customization function have been developed. One is called the text-input approach whose mechanism is based on the fuzzy AHP technique. The other is called the picture-input approach whose mechanism is based on the BP neural network technique. The text-input approach requires a game player to perform a pairwise comparison on 16 personality traits. The picture-input approach requires a game player to view and evaluate a sample set of pictures.

Experiment results indicated that the text-input approach is superior to the picture-input approach, in terms of recommending an appropriate picture to a game player; however, at the price of needing more user input time.

Appendix. Arithmetical operators for fuzzy numbers

The fuzzy arithmetic operators used in this research are defined below (Laarhoven & Pedrycz, 1983; Zadeh, 1975), by referring two fuzzy numbers \( \tilde{a}_1 = (l_1, m_1, r_1) \) and \( \tilde{a}_2 = (l_2, m_2, r_2) \).

(1) Addition operator: \( \oplus \tilde{a}_1 \oplus \tilde{a}_2 = (l_1 + l_2, m_1 + m_2, r_1 + r_2) \)

(2) Multiplication operator: \( \odot \tilde{a}_1 \odot \tilde{a}_2 = (l_1 \times l_2, m_1 \times m_2, r_1 \times r_2) \)

(3) Defuzzification operator (Teng & Tzeng, 1993) \( \text{Defuzzy} (\tilde{a}_1) = \frac{|r_1 - l_1| + (m_1 - l_1)|}{3} + l_1 \)

---

Table 3

<table>
<thead>
<tr>
<th>Subjects</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text-input ( x_i )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Picture-input ( y_i )</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>
References


