Evolution of Camerawork in Automatic Comic Generation Using Interactive Genetic Algorithm with Feedbacks from the User Model

Kenta Tsuji and Ruck Thawonmas
Intelligent Computer Entertainment Laboratory, Ritsumeikan University
1-1-1 Nojihigashi
Kusatsu, Shiga, 525-8577, Japan
ruck@ci.ritsumei.ac.jp

II. PROPOSED METHOD

A. User Modeling by Sub-Camera Module

This module simulates the user’s game experience from the log. The user manually operates this module in order to determine the timing for snapshotting a comic frame and the camerawork parameters therein. Such information determined by the user is then recorded. In IGA, described later, the best/worst individual is the one whose parameters are closest/furthest to the parameters determined by the user at the same frame.

B. IGA

An individual in the population consists of three elements (angle, position, and shot distance) as shown in Fig. 1. Each element is quantized to have 8 patterns represented using a 3-digit binary representation. To reduce a chance in getting trapped by a local optimum, we adopt the distributed genetic algorithm (DGA) [6] in this work. In DGA, the population is divided into multiple sub-populations, and genetic operations are conducted independently in each sub-population.

![Three camerawork parameters: Angle, Position, and Shot](image)

Fig. 1. Three camerawork parameters: Angle, Position, and Shot

C. Fitness

We use the fitness that is constructed by a combination of the subjective fitness and the objective fitness described below. Our aim is that of maximizing these two fitness functions, each normalized to have a value in the range \([0, 1]\). The subjective fitness \(Sub\) is defined as

\[
Sub = 1 - \frac{\sum_{i=1}^{N} \text{Euclid}(Best, i) - \text{Euclid}(Worst, i)}{2N\sqrt{3}}
\]  

(1)
where \( N \) is the number of comic frames, \( Euclid(Best, i) \) and \( Euclid(Worst, i) \) represent the Euclidean distance from individual \( i \) to the best and worst individual in the corresponding sub-population, respectively.

![Image](https://via.placeholder.com/150)

**Fig. 2.** Example of a case where the imaginary line is crossed and is not crossed.

Next for the objective fitness, we introduce two criteria: the first one for dealing with the smoothness in the transition of camerawork parameters and the second one for dealing with the exposurer degree. As for the first criterion, we divide the frames in chronological order into multiple chunks, each consisting of three frames. For each chunk, we then examine each pair of two consecutive frames and set the smoothness of pair \( i, F(i) \), to 1 if the camerawork parameters change smoothly, otherwise to 0. In the example in Fig. 2 for camera position, assume that the first frame has the camera position at \( A \); if the camera in the next frame is at \( B \), then \( F = 1 \); however, if the camera is at \( C \), then \( F = 0 \). In this example, the camera does not cross the imaginary line in the former while crossing the line in the latter case. The objective fitness function about the smoothness \( \text{Obj}_1 \) is given by

\[
\text{Obj}_1 = \frac{N \sum_{i=1}^{2} F(i)}{N/3},
\]

(2)

The other objective fitness \( \text{Obj}_2 \) defines the degree the main character is not blocked by any obstacles in a frame of interest. It is given as

\[
\text{Obj}_2 = (\text{The number of unblocked frames})/N
\]

(3)

Finally, the total fitness is defined as

\[
\text{fitness} = 1/2 \text{Sub} + 1/4\text{Obj}_1 + 1/4\text{Obj}_2
\]

(4)

Figure 3 shows an example of a resulting comic, the one with the highest fitness value and the end of evolution.

III. USER EVALUATION

We carried out a user evaluation where 10 evaluators were asked to compare a comic generated by the proposed method, comic with randomly assigned camerawork parameters, and comic with the user-model camerawork parameters, all from their own experiences. Each evaluator first plays the game developed for this experiment using Unity\(^1\), from which the game log is recorded. Then, the above three comics are generated accordingly and presented to the evaluator. The evaluator must rank the three comics. Table 1 shows the evaluation results, which confirm the effectiveness of the proposed method over the other two methods.

**Table 1. Evaluation Results**

<table>
<thead>
<tr>
<th></th>
<th>Rank 1st</th>
<th>Rank 2nd</th>
<th>Rank 3rd</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>User Model</td>
<td>3</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>Proposed Method</td>
<td>5</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

IV. CONCLUSIONS

By introducing the user model into IGA, we can reduce a burden on the user while maintaining the user’s satisfaction to the generated comics. However, the proposed method still generated a comic which includes unacceptable frame, where some parts of the main character are hidden behind obstacles. We consider this is due to our mechanism for \( \text{Obj}_2 \) and therefore plan to advance the mechanism in near future.

REFERENCES


\(^{1}\)http://www.unity3d.com/