Proc. of the 2nd IEEE Global Conference on Consumer Electronics (GCCE 2013), Tokyo, Japan, pp. 185-186, Oct. 1-4, 2013.

Frame Layout Determination with IGE for an Automatic Comic Generation System

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

***Abstract*—We propose a frame layout determination method for an automatic comic generation system. Most previous work applied simple frame layouts or did not consider user preferences. In this work, we divide frame-layout types into four patterns and introduce interactive grammatical evolution (IGE) for coping with user preferences. We also verify if the proposed method can generate a comic with a high IGA fitness value from a user experience at a museum in Second Life, one of the most popular metaverses.**

***Keywords*— comic generation, frame layout, interactive grammatical evolution**

I. Introduction

A major advantage in summarizing a given experience as a comic is that of enabling representing and sharing such information in a style familiar to the user or others they would like to share such an experience with. Challenging research issues for automatic comic generation include frame extraction, frame layout, and camera work. We focus here on frame layout due to lack of previous work tackling this issue seriously, i.e., either using simple layouts or not considering user preferences.

In this work, we first divide frame-layout types into four patterns and then propose a method using interactive grammatical evolution (IGE) for having the user narrow down frame-layout candidates according to their preferences. The effectiveness of the proposed method is confirmed by the increase in the IGE's fitness value and visual of a resulting comic. We target our work in user experiences in a Second Life museum. In this work, we use the same camera work as the play time, and consider frames important at the timing where the user stops moving.

II. Comic Generation System

In order to incorporate user preferences into the frame layout, we adopt IGE, which was previously used by us for extraction of comic frames [1]. We show below the proposed algorithm for frame layout.

1. Present the user with *n* comics, each with a different set (individual) of frame layouts for their pages, and ask the user to evaluate such comics and select the best and worst.
2. Evolve each individual for a number of generations by using genetic algorithm, with the goal to move them towards the best (*best*) and away from the worst (*worst*).
3. If the terminal condition does not hold, repeat Step 1, otherwise, select the individual with the highest fitness value and generate a comic whose frame layouts are determined based on this individual.

This work was supported in part by Grant-in-Aid for Scientific Research (C), No. 23500192, the Japan Society for Promotion of Science.

Frame layout is done by partitioning the comic page with straight lines. This process can be represented by a tree structure [2], whose example is shown in Fig. 1. We propose four frame-layout types and define the GE grammar for each of them. Due to limited space, we only explain two of them below, where the right-hand side of "::=" represents a set of child nodes of the one on the left-hand side, and the operator "|" indicates a choice. In addition, "leaf" and "big" represent a frame and a large frame while "hor" and "ver" are horizontal and vertical partitions, respectively.

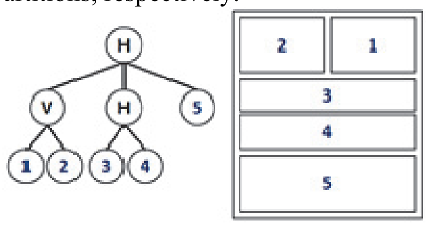

Fig. 1. Example of a frame-layout tree and the resulting layout

Impact: This layout places a large frame and other smaller frames in one page. Comic writers use this style to draw an interest from the reader towards the large frame. Both horizontal and vertical layouts are used, as shown in Fig. 2.

<root>::=<hor_impact> | <ver_impact>
<hor_impact>::=big <hor_i> | <hor_i> big
<hor_i>::=leaf leaf | leaf leaf leaf | leaf leaf leaf leaf |
<ver> <ver> | <ver> leaf | leaf <ver>
<ver>::= leaf leaf
<ver_impact>::=big <ver_i> | <ver_i> big
<hor_i>::=leaf leaf | leaf leaf leaf | leaf leaf leaf leaf |
<hor> <hor> | <hor> leaf | leaf <hor>
<hor>::=leaf leaf

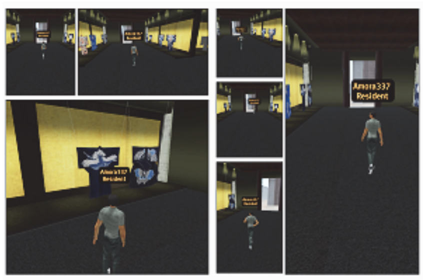

Fig. 2. Example of the impact-type layout.

Turning: This type uses vertical layouts for modulating the reading pitches, as shown in Fig. 3. Comic writers use it when a scene change occurs.

<root>::=<ver_turning>
<ver_turning>::=leaf leaf <hor> | leaf <hor> leaf |<hor> leaf leaf | <hor> <hor> leaf |<hor> leaf <hor> | leaf <hor> <hor>
<hor>::=leaf leaf

Fig. 3. Example of the turning-type layout

For more efficient search, we adopt Distributed Genetic Algorithm (DGA) [3]. DGA divides the population into small groups called islands and conducts genetic operations independently in each island. It also performs the process called immigration that exchanges individuals among different islands. Due to the advent of DGA, the chance of being trapped in local optima is decreased, resulting in satisfactorily high fitness values.

III. User Evaluation

For Step 1 in Sec. II, the individual with the highest fitness value of each island is selected for being directly evaluated by the user. This feedback is then used for calculating the fitness value for all individuals in the population. For this task, we use the tree edit distance (TED) [4]. The similarity between two trees A and B is represented by (1), which is the number of additions, deletions and/or replacements of B (*TreeB*) in order to exactly match A (*TreeA*). *TED*(*BigTree*, φ) is a TED of the tree which has more nodes, between A and B, and the empty tree φ.

$$sim(TreeA,TreeB)=\frac{TED(BigTree,\phi)-TED(TreeA,TreeB)}{TED(BigTree,\phi)} \quad (1)$$

The fitness value of each individual is represented by (2). In this formula, *comic* is an individual of interest, *page* shows the number of pages, and *sim*(A, B, n) is formula (1) for page n of A and B. Note that the fitness value of an individual increases as it becomes more similar to *best* and decreases as it becomes more similar to *worst*.

$$Fitness=\frac{\left(\frac{1}{page}\left(\sum_{k=1}^{page} sim(best,comic,k) - weight\sum_{l=1}^{page} sim(worst,comic,l)\right)+1\right)}{2} \quad (2)$$

The weight in (2) gradually decreases as the number of generations grows. If this is not done, the individually fitness cannot increase because *worst* will become more similar to *best* through evolution.

IV. Experiment

Here, we verify if the fitness value of an individual increases trough evolution. In this experiment, the number of islands was set to five, and the user was asked to perform user evaluation four times at every 25 generations, i.e., at the 0th, 25th, 50th, and 75th generations. Figure 4 shows this transition for each of the five islands A, B, C, D, E. Figure 5 shows the resulting comic, whose layouts are impact, turning, impact from right (the first page) to left (the last one), a Japanese comic style.

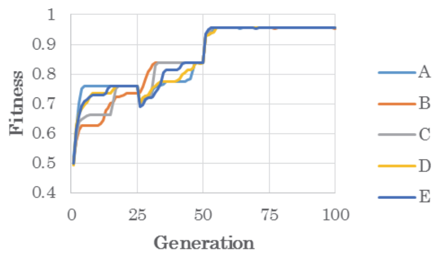

Fig. 4. Transition over generations of the average fitness of each island

Fig. 5 Comic output in the experiment

V. Conclusion

In this work, we proposed a method for performing page-layout of a comic for summarizing user experiences in a metaverse museum. By dividing fame layouts into four types, it is possible for the evolution process to converge to the user's preference with a small amount of calculation. As our future work, we plan to add new elements of comics into the fitness calculation, e.g., the size of each frame, the importance of the extracted image, etc.

References

[1] Ruck Thawonmas and Yoshinori Tani, "Frame selection using interactive grammatical evolution for automatic comic generation from game logs," Proc. of the 2011 IEEE Conference on Computational Intelligence and Games (CIG2011), pp. 31-38, 2011.

[2] Takamasa Tanaka, Kenji Shoji, Fubito Toyama, and Juichi Miyamichi, "Layout analysis of tree-structured scene frames in comic images," Proc. of the 20th International Joint Conference on Artificial Intelligence (IJCAI'07), pp. 2885-2890, 2007.

[3] Reiko Tanese, "Distributed genetic algorithms," Proc. of the 3rd International Conference on Genetic Algorithms, pp. 434-439, 1989.

[4] Erik D.Denaine, Shay Mozes, Benjamin Rossman, Oren Weimann, "An optimal decomposition algorithm for tree edit distance, automata, languages and programming," Lectire Notes in Computer Science, Vol. 4596, pp. 146-157, 2007.

The second author is now with DENSO WAVE INCORPORATED.