Frame Selection for Automatic Comic Generation from Game Log

Tomonori Shuda and Ruck Thawonmas

Intelligent Computer Entertainment Laboratory
Graduate School of Science and Engineering, Ritsumeikan University
Kusatsu, Shiga, 525-8577, Japan
`ruck@ci.ritsumei.ac.jp`

Abstract. Recently, we have presented a comic generating system that visualizes an online-game play. Our system was inspired by a former work of Shamir et al. However, comics generated in their work can have series of similar frames when multiple actions occur near each other in both time and space. In this paper, we first present a frame-selection module that uses Habituated Self-Organizing Map. Our method prevents comic readers from boredom by getting rid of resemble consecutive frame candidates. We then evaluate the method by a subject experiment using a play from the ICE, an online-game developed in our laboratory. Experimental result confirms that our method is effective in making output comics more interesting than a baseline method.

1 Introduction

Recently, we have presented a system for generating comics from online-game play log [1]. Figure 1 gives an overview of the system. This system enables users to look back their plays or to introduce their plays to other users. Comic-style outputs have advantages than other-style outputs. For example, the comic-style enables one to distinguish important scenes based on the frame size. And one can grasp the whole story at one glance. In addition, traditional comic techniques can be introduced to achieve expression of various situations. Comic-style representation has been used for summary of conference participation [2], diary experiences [3], and video contents [4], as well as online-game events [5].

Our work in [1] was inspired by the work in [5] and adapted its scene partition method and event extraction method. However, comics generated in [5] can have series of similar frames when multiple actions occur near each other in both time and space. In this paper, we describe and evaluate our solution to this problem, i.e., the frame-selection module in Fig. 1 where Habituated Self-Organizing Map (HSOM) [6] is used. As in [1], we test the system using an online game called the ICE, under development at the authors' laboratory, where typical online-game missions, such as monster fighting and item trading, are available. A screenshot of this game is shown in Fig. 2.
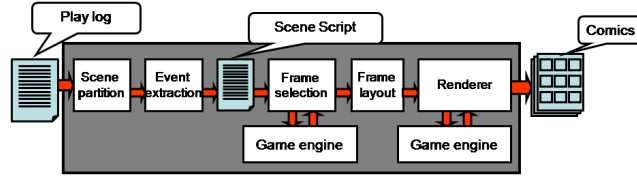
Fig. 1. Overview of the comic-generation system

Fig. 2. Screenshot of the ICE

2   Frame-Selection Module

In order to prevent comic readers from boredom, the frame-selection module removes resemble and consecutive frame candidates using HSOM. Frame candidates are generated in the previous module and stored in the scene script (c.f. Fig. 1). Each frame candidate represents an output-comic frame and has frame information such as the frame importance, in the range of 0 to 1, the frame time, and the camera parameters. HSOM enables selection of frame candidates from perspective of comic readers' interest. It is an integration of Self-Organizing Map (SOM) and a habituating layer.

Fig. 3 gives an overview of the HSOM architecture. SOM consists of an array, called the competitive layer, of k competitive neurons arranged in a 2-dimensional rectangular grid. Let assume that there are M input patterns, each corresponding to one of the M frame candidates in a scene of interest, with N dimensions, and that a competitive neuron i has an associated weight vector $\mathbf{w^c}_i = [w_{i1}^c, w_{i2}^c, \ldots, w_{iN}^c]^T$. The habituating layer is an array of k habituating neurons, each connected to the competitive neuron below, having habituating weights w_i^h. The smaller w_i^h means the higher habituating degree. For each scene, frame selection goes through the following steps.

1) Determine the elements of input patterns $\mathbf{x}(t)$, $t = 1, 2, \ldots, M$. These elements consist of the frame time and entities in the entity list. The frame time represents the time when frame candidate t occurs, and it is normalized so that the scene's start time is 0 and the scene's end time is 1. The entity
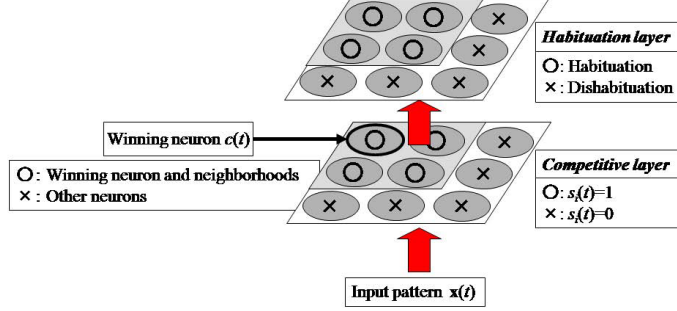
Fig. 3. HSOM architecture

list is the list of all entities in the frame candidates of the scene. If an entity exists in frame candidate t, the corresponding element will be 1; otherwise, it will be 0. Each $\mathbf{x}(t)$ is then normalized, i.e., $\mathbf{x}(t) = \mathbf{x}(t)/ \parallel \mathbf{x}(t) \parallel$.

2) Set the time step $t = 1$ and initialize all competitive-neuron weight vectors $\mathbf{w^c}_i(t)$, $i = 1, 2, \ldots, k$, randomly in the range from 0 to 1, and normalize them, i.e., $\mathbf{w^c}_i(t) = \mathbf{w^c}_i(t)/ \parallel \mathbf{w^c}_i(t) \parallel$. In addition, initialize all habituating-neuron weights $w_i^h(t)$ to 1, which means they are not habituated.

3) Input the input pattern $\mathbf{x}(t) = [x_1(t), x_2(t), \ldots, x_N(t)]^T$ and choose the winning neuron $c(t)$ that is closest to $\mathbf{x}(t)$, i.e., $c(t) = \arg_j \min \parallel \mathbf{x}(t) - \mathbf{w^c}_j(t) \parallel$

4) Select frame candidate t if the sum of $w_c^h(t)$ and its frame importance is more than threshold T_h, and then proceed to 5); otherwise, increment t and return to 3).

5) Update all competitive-neuron weights as follows:

$$\mathbf{w^c}_i(t + 1) = \mathbf{w^c}_i(t) + \alpha h_{ci}(t)[\mathbf{x}(t) - \mathbf{w^c}_i(t)] \tag{1}$$

where α is the learning rate, and $h_{ci}(t)$ is the neighborhood function defined as

$$h_{ci}(t) = \begin{cases} 1, & \text{if the competitive neuron } i \text{ is the winning neuron } c(t) \\ & \text{or } c(t)'s \text{ Moore neighborhood.} \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Then, normalize $\mathbf{w^c}_i(t + 1)$.

6) Update all habituating-neuron weights as follows:

$$w_i^h(t + 1) = w_i^h(t) + \frac{\beta(1 - w_i^h(t)) - s_i(t)}{\lambda}. \tag{3}$$

In the above formula, β is the recovery rate. The stimulus value $s_i(t)$ has the same value as $h_{ci}(t)$ of the connected competitive neuron. The constant λ influences both habituation and dishabituation speeds. The values of λ are different for the habituating neuron connected to the winning competitive
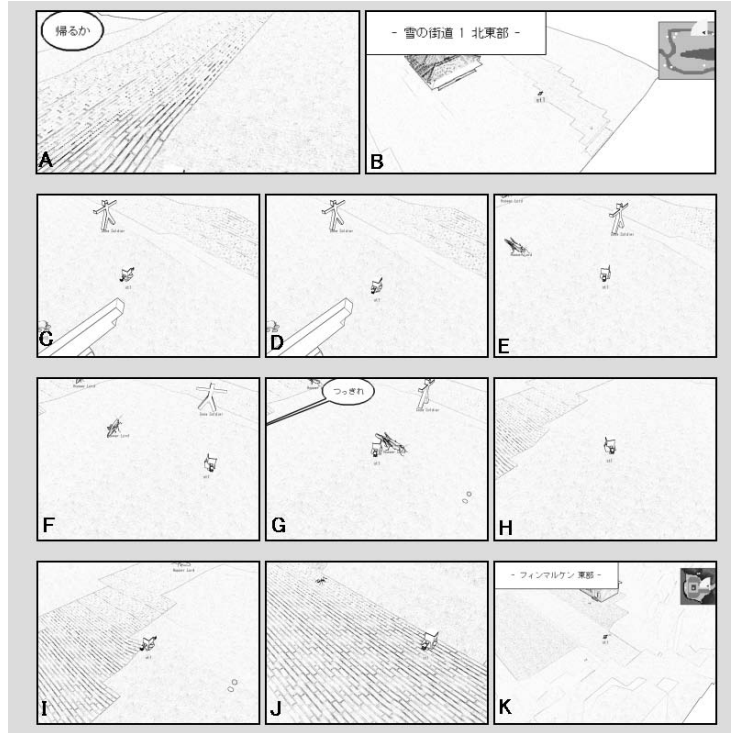
Fig. 4. Example sequence of frame candidates

neuron, for the eight habituating neurons connected to one of the Moore neighborhoods of the winning competitive neuron, and for the remaining habituating neurons.

7) If t is smaller than M, increment t and return to 3).

Figure 4 depicts an example sequence of frame candidates for Comic A described in Sec. 3. They are input to the frame-selection module. Figure 5 depicts the selected frames. Because frame candidates D, F, and I have content similar to C, E, and H, respectively, they are not selected.

3   Evaluation

We tested the frame-selection method with our comic-generation system using the ICE. We requested 12 subjects, who are students in our university, to compare two comics generated from a play[1] of the ICE. These two comics are described below while other conditions, such as camera positions, are identical.

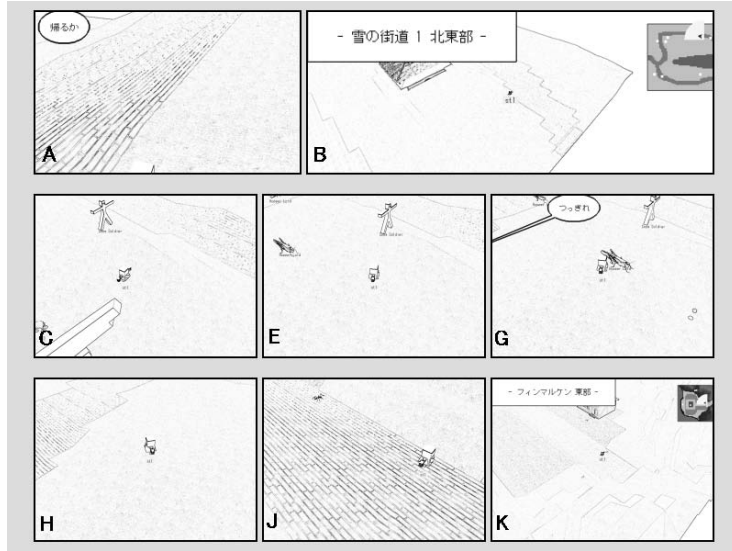[1] http://www.youtube.com/watch?v=DmvUyl6eZAs

Fig. 5. Selected frames in Fig. 4

- Comic A generated with the proposed frame-selection method described in Sec. 2.
- Comic B generated with a baseline frame-selection method described below using interaction level.

The proposed method selected 52 frames out of 135 frame candidates[2] . The baseline method was also set such that it selected 52 frames; this method selects frame candidates in decreasing order of the smoothed value of the interaction level defined in [1].

We set the number of competitive neurons $k = 16$, arranged in 4 x 4, in order to heuristically limit different situations in a scene to 16. We used $\alpha = 0.25$, $\lambda = 3.33$ for the habituating neuron connected to $c(t)$, $\lambda = 14.33$ for the eight habituating neurons connected to one of the Moore neighborhoods of $c(t)$, $\lambda = 100$ for the remaining habituating neurons. These values were determined following [6]. We chose $T_h = 0.7$ and $\beta = 1.8$ based on our experience because these values were not stated in [6].

We had all subjects watch a video clip of the play in use after explaining them the experiment procedure and how to answer. Then, we distributed the two comics to each subject and had them individually answer which comic fits each of the following questions.

- Which one has the better camera work?
- Which one has the better dynamics?
- Which one represents better the video-clip content?

---

[2] The number of frame candidates were in the range of 10 to 50 for each scene.

Table 1. Experimental Result

	Camera work	Dynamics	Content representation	Interestingness
Votes for Comic A	9	7	7	7
Votes for Comic B	3	5	5	5
Number of subjects	12	12	12	12

– Which one is more interesting?
– What points of the comics should be improved? (free description)

Table 1 gives the result of the questionnaire except for the fifth descriptive question. It can be seen that Comic A exceeds Comic B for all four questions. This result confirms that our proposed method can reduce resemble camera work and improve dynamics, content representation, as well as interestingness.

4   Conclusions and Future Work

This paper described and evaluated our HSOM-based frame-selection method. The experiment result given in Sec. 3 confirms the effectiveness of this method. Future work includes introduction of a technique that selects frames from story-coherence perspective.

References

1. Thawonmas, R., Shuda, T.: Comic Layout for Automatic Comic Generation from Game Log. In: Proc. 1st IFIP Entertainment Computing Symposium, Milan, Italy (September 2008),
   `http://www.ice.ci.ritsumei.ac.jp/~ruck/PAP/ThawonmasECS08.pdf`
2. Sumi, Y., Sakamoto, R., Nakao, K., Mase, K.: ComicDiary: Representing individual experiences in a comic style. In: Borriello, G., Holmquist, L.E. (eds.) UbiComp 2002. LNCS, vol. 2498, pp. 16–32. Springer, Heidelberg (2002)
3. Cho, S.B., Kim, K.J., Hwang, K.S.: Generating Cartoon-Style Summary of Daily Life with Multimedia Mobile Devices. In: Okuno, H.G., Ali, M. (eds.) IEA/AIE 2007. LNCS (LNAI), vol. 4570, pp. 135–144. Springer, Heidelberg (2007)
4. Calic, J., Gibson, D.P., Campbell, N.W.: Efficient Layout of Comic-like Video Summaries. IEEE Transactions on Circuits and Systems for Video Technology 17(7), 931–936 (2007)
5. Shamir, A., Rubinstein, M., Levinboim, T.: Generating Comics from 3D Interactive Computer Graphics. IEEE Computer Graphics and Applications 26(3), 53–61 (2006)
6. Marsland, S., Nehmzow, U., Shapiro, J.: A real-time novelty detector for a mobile robot. In: EUREL European Advanced Robotics Systems Masterclass and Conference (2000)