Rule-Based Camerawork Controller for Automatic Comic Generation from Game Log

Ruck Thawonmas, Ko Oda, and Tomonori Shuda

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

Abstract. We propose a rule-based camerawork controller for a recently proposed a comic generation system. Five camerawork rules are derived through an analysis of online-game webcomics about Lineage 2, one rule for each of the five event types: chatting, fighting, moving, approaching, and special. Each rule consists of three parts relating to the three camera parameters: camera angle, camera position, and zoom position. Each camera-parameter part contains multiples shot types whose value indicates the frequency of their usages in the analyzed webcomics. In this paper, comic frames generated with the proposed camerawork controller are shown and compared with those generated with our previous controller based on heuristic rules, confirming the effectiveness of the proposed camerawork controller.

1 Introduction

Comic is a promising media for summarizing experiences in an entertaining fashion. Recent comic applications include summarization of activities in a conference [1], daily activities [2], video sequences [3], and game-play activities [4–6]. Comic-based experience summaries facilitate augmenting personal memories as well as promoting communication among user communities.

Since 2008, we have developed a number of techniques [7,8,9] for our comic generation system [5] that aims at automatically generating comic from game log. In our system, we adopted the same approach as in [4] where the game engine of a game of interest is used for rendering comic based on information in the game log. Another approach adopted in [6] is that of composing comic from selected screen shots. Although the game engine must be accessible, the former approach provides more room to play with camerawork and thus more varieties in comic.

Recently, in order to utilize the advantage of the game-engine approach, we proposed in [9] a module for manually editing the camerawork of generated comic. However, its initial camerawork, heuristically decided by the authors, lacks varieties and requires a large amount of editing work and thus causes a high burden to the user. To ease this burden, we address here the issue on automatic camerawork control and propose a camerawork controller that automatically

H.S. Yang et al. (Eds.): ICEC 2010, LNCS 6243, pp. 326–333, 2010.

Fig. 1. Screenshot of The ICE

decides the camerawork of each frame based on rules derived from an analysis of online-game webcomics.

The contributions of this paper are as follows:

1. Universal and reliable camerawork rules derived from webcomics about Lineage 2,
2. The proposed camerawork controller utilizing the derived rules and providing rich varieties of shots,
3. Outline of our automatic comic generation system applicable to any game, provided that its game engine is accessible.

As with our recent papers, we use, as the research platform, an online game called The ICE (Fig. 1), under development at the authors' laboratory. In The ICE, typical online-game activities, such as monster fighting, chatting with other characters, and item trading, are available.

2 Comic Analysis

We selected the web comics [10] in Japanese whose stories are about Lineage 2 for analyzing camerawork rules. Our reasons for this selection, besides the third author himself being a resident of Lineage 2 and thus familiar with the contents, are as follows:

Table 1. Resulting camerawork rule for each event type

Event Type	LA	EL	HL	BE	F	TF	S	TB	B	SC	C	M	K	L	SL
Chatting	266	5807	344	74	2847	2489	512	242	255	432	1265	2939	866	770	136
Fighting	65	631	49	0	276	265	66	48	65	40	74	281	162	147	33
Moving	4	97	13	0	35	31	26	5	15	5	13	30	18	33	13
Approaching	2	21	1	0	19	1	2	0	1	0	5	9	4	4	2
Special	122	1733	199	74	962	779	140	58	110	188	357	897	341	243	44

- Their stories are based on Lineage 2, fitting to our comic genre.
- Each story has a sufficient length, ensuring the reliability of analysis results.
- Multiple comic writers contribute to the above site, preventing the analysis results from being biased by a single comic writer.

In order to increase the quality and reliability in analysis, we asked two college students to independently read and analyze, based on the same analysis manual, all comics in this site. Their task was to find the frequency of each shot type relating to the camera angle, camera position, and zoom position in the comic frames for five event types defined as follows:

Chatting event that includes at least one dialogue or chat balloon
Fighting event that includes a content relating to fighting activities
Moving event that shows movement of characters
Approaching event where at least two characters or objects are approaching each other
Special event to which the above four events do not apply.

Note that multiple event types might be applicable to an analyzed comic frame. For each of such frames, the shot types for the camera angle, camera position, and zoom position were shared by and counted for all applicable event types.

Table 1 shows the analysis results integrating the individual results from the two analyzers. An element in this table indicates the frequency of a shot type of interest in the corresponding event. The number of shot types relating to the camera angle, camera position, and zoom position are four, five, and six, respectively; i.e.,

Camera angle: Low Angle, Eye Level, High Angle, and Bird Eye shots;
Camera position: Front, Tilted Front, Side, Tilted Back, and Back shots;
Zoom position: Super Close-up, Close-up, Medium, Knee, Long, and Super Long shots;

where the shot target is the last character who chatted and who performed a fighting action for the chatting event and the fighting event, respectively; for the other event types, the shot target is the player character.

For each shot type of a camera parameter of interest, we use the frequency in Table 1 as its weight in the corresponding event type. The camerawork controller, described in the next section, uses these weights in a roulette-wheel fashion for

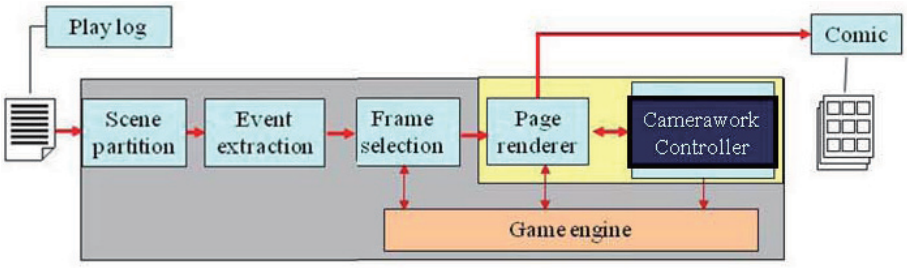


Fig. 2. Architecture of the comic generation system

deciding a shot-type of each camera parameter for a given event type. For example, considering selection of a shot type of the camera angle for the fighting event, the roulette wheel has four areas with the sizes in the proportion of 65:631:49:0.

3 Camerawork Controller and Comic Generation System

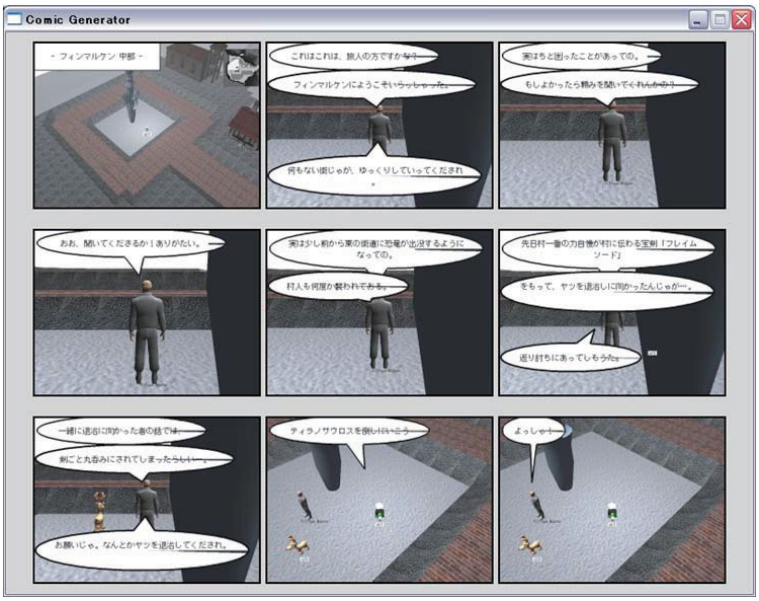
Figure 2 depicts an overview of our comic generation system [5] that includes the proposed camerawork controller as the most right module. In this system, first, the play log is chronologically divided into multiple partitions, each called a scene, connecting a story (if any) in the play. At the event extraction module, a scene is further divided into multiple chunks, each called an event composed of multiple actions such as fighting, chatting, and moving. After frames have been selected from those events at the frame selection module [8], the parameters for deciding frame size and shape are decided. These parameters are used together with the camera angle, camera position, and zoom position for rendering comic frames.

The proposed camerawork controller decides the camerawork parameters for a given frame through the following three steps:

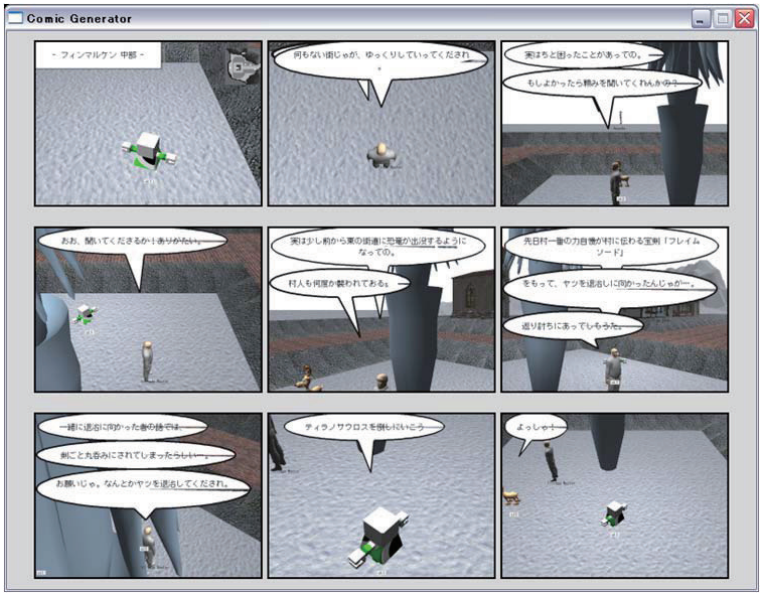
1. Determine the event type by using the majority voting of all action types occurring in the frame
2. Perform roulette-wheel selection of a shot type of each camera parameter for the determined event type
3. Set the value of each camera parameter to the predefined value of the selected shot type.

4 Results and Discussions

Figures 3.a and 3.b show a comic page generated with the previous camerawork controller, used in 5,7–9, and that with the proposed camerawork controller, respectively. This is from a scene where the player, the robot-like character, and his friend, the dog-like character, are talking to a mission master, the human-like character, in order to receive a game mission. Because the former controller uses

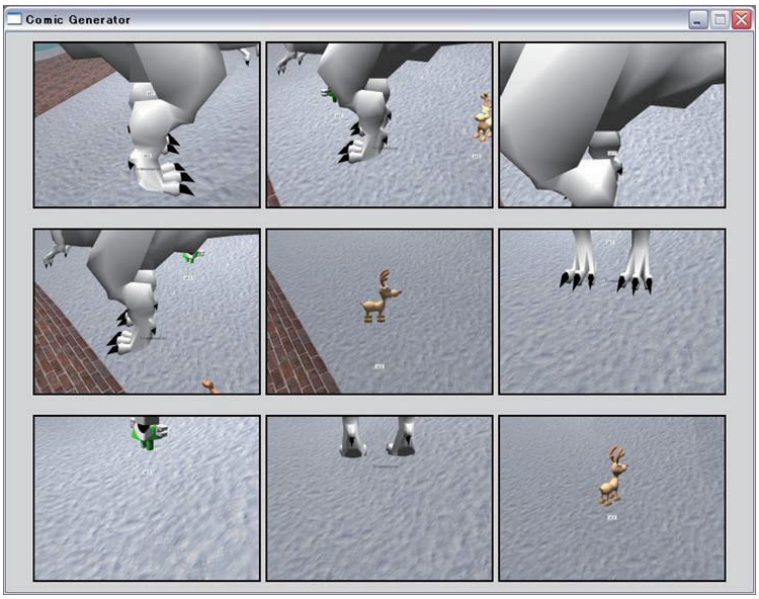


(a) Previous

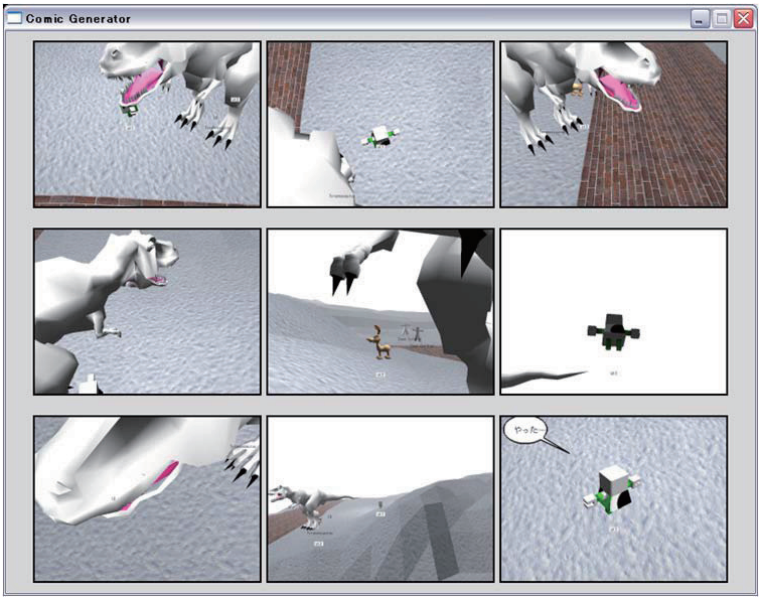


(b) Proposed

Fig. 3. Comic page for a chatting scene with the previous camerawork controller and the proposed camerawork controller



(a) Previous



(b) Proposed

Fig. 4. Comic page for a fighting scene with the previous camerawork controller and the proposed camerawork controller

only one combination of the camera angle, camera position, and zoom position for each event type, there exist two frame subsequences, each having a same shot type, in the former figure. On the contrary, the latter figure has a richer variety of shot types. Visual comparison between these figures and that between Figs. 4.a and 4.b for a fighting scene also confirm the superiority of the proposed camerawork controller.

5 Conclusions and Future Work

This paper described our camerawork controller that decides a combination of the camera angle, camera position, and zoom position for a frame according to the frame's event type. Such a decision is done for a given event type using roulette-wheel selection where each shot-type's weight of a camera angle of interest represents its portion in the wheel. An analysis of Lineage 2 webcomics was conducted to derive those weights. Visual comparison between our previous camerawork controller and the proposed camerawork controller confirms the superiority of the latter.

Our future work includes fine tuning of the camera angle, camera position, and zoom position based on the entropy of each frame segment formed with a technique such as the Berkeley Segmentation Engine [11].

Acknowledgements

This work was supported in part by Grant-in-Aid for Scientific Research (C), No. 20500146, the Japan Society for Promotion of Science, and by Global COE (Center of Exellence) Progam "Digital Humanities Center for Japanese Arts and Culture".

References

1. 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)
2. 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)
3. 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)
4. Shamir, A., Rubinstein, M., Levinboim, T.: Generating Comics from 3D Interactive Computer Graphics. IEEE Computer Graphics and Applications 26(3), 53–61 (2006)
5. Shuda, T., Thawonmas, R.: Automatic Comic Generation Using Online-Game Play Log. Journal of Game Amusement Society 3(1), 40–45 (2009) (in Japanese)

6. Chan, C.-J., Thawonmas, R., Chen, K.-T.: Automatic Storytelling in Comics: A Case Study on World of Warcraft. In: CHI Extended Abstracts 2009, pp. 3589–3594 (2009)
7. Thawonmas, R., Shuda, T.: Comic Layout for Automatic Comic Generation from Game Log. In: Proc. IFIP International Federation for Information Processing (ECS-2008), vol. 279, pp. 105–115 (2008)
8. Shuda, T., Thawonmas, R.: Frame Selection for Automatic Comic Generation from Game Log. In: Stevens, S.M., Saldamarco, S.J. (eds.) ICEC 2008. LNCS, vol. 5309, pp. 179–184. Springer, Heidelberg (2008)
9. Thawonmas, R., Oda, K., Shuda, T.: Camerawork Editor for Automatic Comic Generation from Game Log. In: CD-ROM Proc. of Nicograph International 2009, Kanazawa, Japan, June 19-20 (2009)
10. Lineage 2 webcomics: `http://lineage2.plaync.jp/l2fun/comic/menu.aspx` (last accessed on June 22, 2010)
11. Martin, D., Fowlkes, C., Malik, J.: Learning to Detect Natural Image Boundaries Using Local Brightness, Color and Texture Cues. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(5), 530–549 (2004)