Computational Cartoonist: A Comic-Style Video Summarization System for Anime Films

Tsukasa Fukusato[1(✉)], Tatsunori Hirai[1], Shunya Kawamura[1], and Shigeo Morishima[2,3]

[1] Waseda University, Tokyo, Japan
tsukasa@moegi.waseda.jp
[2] Waseda Research Institute for Science and Engineering, Tokyo, Japan
[3] JST CREST, Tokyo, Japan
shigeo@waseda.jp

Abstract. This paper presents Computational Cartoonist, a comic-style anime summarization system that detects key frame and generates comic layout automatically. In contract to previous studies, we define evaluation criteria based on the correspondence between anime films and original comics to determine whether the result of comic-style summarization is relevant. To detect key frame detection for anime films, the proposed system segments the input video into a series of basic temporal units, and computes frame importance using image characteristics such as motion. Subsequently, comic-style layouts are decided on the basis of pre-defined templates stored in a database. Several results demonstrate the efficiency of our key frame detection over previous methods by evaluating the matching accuracy between key frames and original comic panels.

Keywords: Comic generation · Shot clustering · Shot boundary detection

1 Introduction

Comic has grown to become one of the most efficient storytelling mediums across the world, with many cartoonists creating their own compositions and imagery [10]. Recently, cartoonists extend comic techniques to cartoon animation (or anime films), and this has resulted in the increasing general population in producing and viewing many anime films. However, the number of anime episodes is very large. When people choose their favorite anime contents from the large number of anime films, to read comics for understanding all episodes of the anime is more effective than to watch the films.

To facilitate intuitive access to video archives, the main challenge for video summarization and browsing systems is achieving a satisfactory balance between removing redundant sections and maintaining representative coverage of the video. Some works developed comic-style browsing tools, to display thumbnails

Q. Tian et al. (Eds.): MMM 2016, Part I, LNCS 9516, pp. 42–50, 2016.
DOI: 10.1007/978-3-319-27671-7_4

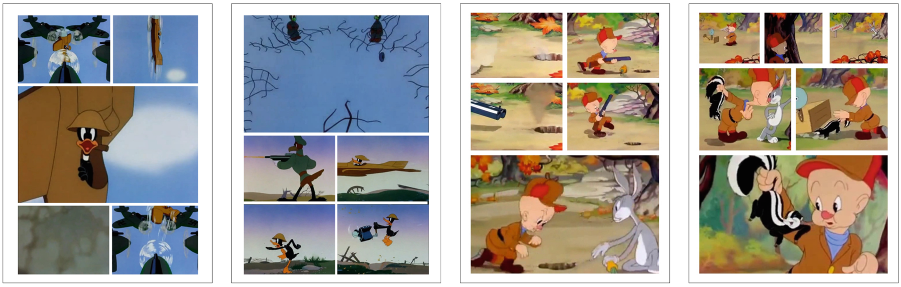

Fig. 1. Two comic-style summarizations generated by our method. Left: “Daffy The Commando”(1943, public domain). Right: “A Wild Hare”(1941, public domain).

appropriately indicating the contents of input video by computing scene importance. For example, comic-style video summarization methods based on image features or annotation features (e.g., subtitles or game play logs) have been proposed [1,12,14,16,19]. However, previous methods did not provide evaluation criteria to determine whether the key frames could be appropriately used in comic-style summarization. The domain-specific approach of cartoonists cannot be easily extended to generate comic-style summaries such as a film comic.

Our goal is to generate a comic-style video summary for anime films (referred to as comicalization) by computing the importance of shots and frames from anime sequence (Fig. 1). In this paper, to quantify the rule of key frame, we focus on Japanese anime films that are generally created by interpolating the panels of original comics. There is the correspondence between the keyframes of the anime film and the original comic panels. The main idea of our approach is to measure the matching accuracy between key frames and original comic panels. This system detects key frames based on image features and then automatically generate the comic layout. The proposed method's overall process is as follows:

1. Segment an input anime film into shots using image features.
2. Determine the cost function based on the correspondence between anime films and original comics.
3. Extract key frames of each shot using image features.
4. Compute the paneling score according to the importance of key frames and shots.
5. Generate the comic layout from a template layout stored in a database.

The summary should clearly present the meaningful content throughout the scenes of the anime film. The proposed method enables automatic and unsupervised key frame detection using various image features from an input anime film. By calculating the frame importance, we improve the appropriateness of detected key frames for comic-style summaries. Using the calculated importance values, the system automatically generates comic paneling from database of template layout. As the result, we can reduce the manual labor required for creating comics from anime films (Fig. 2).

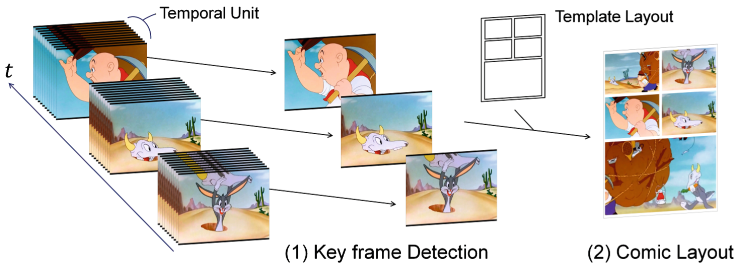

Fig. 2. System overview.

The remainder of this paper is organized as follows. Related works are reviewed in Sect. 2. We discuss the main ideas underlying the algorithms used in the proposed method in Sect. 3. Section 4 presents experimental results for some anime films, and we conclude this paper and discuss limitations and future work in Sect. 5.

2 Related Work

There are some researches of comic-style video summarization, which determine key frames in video sequence. Cho et al. [4] use Lifelog data on mobile devices, and Thawonmas [15] proposes comic summaries based on Gamelog, i.e., the playing data in an online game. Furthermore, Shamir's method [13] can be configured to accommodate user preferences. By contrast, these methods require recorded content information.

In image-based methods, Zhuang's [20] method provides an unsupervised clustering method based on hue-saturation (HS) color features (16 × 8). Calic's [2] method generates a comic-style summarization system based on HSV features and clustering. These methods allow users to visualize key frames in videos easily; however, it does not focus on shot importance.

Uchihashi et al. [18] propose Video Manga, which computes the shot importance (using the total length of each shot cluster) and layouts key frames based on a film comic format. However, they do not consider the frame importance for key frame detection. Kasamatsu et al. [7] propose a method to detect key frames using the central frames of video shots, and determine panel sizes using YCbCr color clustering. With these methods, it is possible to visualize video content with relative ease. However, their methods does not consider the temporal information in each shot. Hence, to generate more effective summaries, computing the frame and shot importance for key frame detection and comic layout is essential.

These researches cannot assign optimum key frames since they do not consider a large number of the scene structure in anime films. Therefore, we propose an evaluation criteria based on the correspondence between anime films and original comics.

3 Comic-Style Video Summarization

3.1 Shot Transition Detection

A shot is a series of interrelated consecutive pictures taken continuously by a single camera, representing a continuous motion in time and space. in particular, each anime shot (e.g., character motion) is created on the vasis of the composition of original comic panels. First, we investigate whether each shot in 20 Japanese anime films matches the original comic panel, and confirm that approximately 88.7 % of the anime shots include key frames matched with the original comic panels. Therefore, we assume that each anime shot has one key frame, and segment an anime film into basic temporal units (shots). Kasamatsu [7] uses shot segmentation based on the sum of absolute differences (SAD) between two consecutive frames. However, the shot transitions of the anime films include both abrupt transitions between two consecutive frames and gradual transitions, e.g., white/black fades and various camera techniques. Therefore, to determine abrupt shot transitions, we use the edge of orientation histograms [8] and Lian's temporal segmentation method [9]. Frame differences based on pixel differences, RGB color histograms, and a block-matching motion estimation algorithm are performed. These measurements are robust against camera operation (e.g., zoom in/out, pan, and tilt) and object motions. In this study, the minimum shot sequence length is defined as ten frames.

We have verified the results of our shot segmentation with an accuracy evaluation using precision and recall rate. In the experiments, 20 Japanese anime films are used. The mean accuracy of the shot segmentation are $Precision = 92.34\,\%$, $Recall = 86.09\,\%$. These results show that this approach enables the classification of abrupt transitions, white-fades, black-fades, pans, and zooms in anime films.

3.2 Key Frame Detection

In this section, we describe a method to detect key frames in anime films. Ideally, key frames, which are matched with original comic panels, should capture the semantics of a shot. However, current techniques in Computer Vision techniques are not advanced enough to automatically generate such key frames. Instead, we have to rely on low level visual features, such as color, motion of the object in a shot. Therefore, we detect the key frames based on (1) color transitions, e.g., black and white reversal, (2) characters' motion, and (3) frame composition.

In the color features, Zhuang [20] proposed an unsupervised clustering method based on HS color features. The frame closest to the cluster center is detected as the representative key frame for a given shot. However, capturing consecutive and similar key frames is problematic, because a substantial number of shots are reused in anime films. Therefore, we focus on the cluster outliers such as cutaways and establishing shots. To determine the centroid features for HSV histograms in scene clusters, we perform the k-means clustering algorithm.

The cost function E_1 of the i th frame $(i = 1, \cdots, N)$ is defined as follows

$$E_1(i) = \alpha \cdot \left\{ 1.0 - \sum_k \beta_k \exp\left(\frac{-D_k(i)^{\mathrm{T}} \sum^{-1} D_k(i)}{2} \right) \right\} \tag{1}$$

$$\beta_k = \frac{\omega_k}{\sqrt{(2\pi)^k |\sum|}}$$

$$D_k(i) = H(i) - \hat{H}_k$$

where $\sum$ is a covariance matrix and ω_k is a weight value for the kth cluster. $E_1(i)$ is normalized to the maximum row height α, $H(i)$ is the HSV histogram of ith frame, and $\hat{H}_k$ is the HSV histogram's centroid of the kth cluster. As a result, the cluster outliers are presented as more important and attract user's notice compared to key frames concentrated around the cluster center. This grouping around the cluster centers is caused by common repetitions of similar content in the video, often adjacent in time.

The color-based term (Eq. (1)) does not place constraints on object motion. In the original comics, cartoonists draw the special motions of characters, such as punch motions. Then, we compute the moving object area, i.e., the number of grid flow vectors E_2 having large flow vectors, based on Farneback's [5] optical flow algorithm. The second cost function is expressed as follows:

$$E_2(i) = \iint_{(x,y)\in f} \phi_1(\boldsymbol{v_f}(i,x,y))dxdy \tag{2}$$

$$\phi_1(\boldsymbol{v}_f(i,x,y)) = \begin{cases} 1.0 \ |\boldsymbol{v}_f| > threshold \\ 0.0 \ else \end{cases}$$

In this function, $E_2(i)$ is normalized to the maximum height E_{max} and $\boldsymbol{v_f}$ is a grid flow vector based on optical flow. However, this energy is not strong enough to control the frame comparison (motion of the object). Then we assume that the edge intensity of important frame is high. A regularization term based on the edge intensity, which is computed by Canny's edge detection, is added. The third cost function E_3 is defined as follows:

$$E_3(i) = \iint \phi_2(i,x,y)dxdy \tag{3}$$

$$\phi_2(i,x,y) = \begin{cases} 1.0 \ \ if f(x,y) \in edge \\ 0.0 \ \ else \end{cases}$$

In this function, $E_3(i)$ is normalized to the maximum height E_{max}.

By integrating all the energy terms, the key frame detection of the jth shot is formulated as:

$$\max_{i \in j} \lambda_1 E_1(i) + \lambda_2 E_2(i) + \lambda_3 E_3(i) \tag{4}$$

Generally, the weights are set as $\lambda_1 = 2.0$ and $\lambda_2 = \lambda_3 = 1.0$ to balance the contribution of different terms.

We note that when our key frame detection is applied to anime films (total length of approximately 30 minutes), the number of key frames is typically lower than 100.

3.3 Comic Layout

We describe a method to generate comic-style layouts using key frame importance (as described in Sect. 3.2). This system is mainly inspired by Uchihashi's [17] Video Manga, Cao's [3] and Myodo's [11] layout method. In this paper, we design various layout templates, which contains one, five, and six panels per page. We assume that large panels are very important frames that help the reader understand the comic content, and define the displayed function F_j, which relates the panel size to the layout template (according to the paneling score) for paneling key frames.

To determine paneling size (represented by the paneling score), we use the key frame importance of each shot (Eq. (4)) and the shot importance. Given C clusters in an anime film, a measure of normalized weight W_k for kth cluster (as described in Sect. 3.2) is computed as

$$W_k = \frac{S_k}{\sum_{l=1}^{C} S_l} \tag{5}$$

where S_k is the total length of all shots in the kth cluster, computed by summing the length of all shots in the cluster. W_k is the proportion of shots from the entire anime film that are in the kth cluster. We assume that a shot is important if it is both long and unique, i.e., it does not resemble other shots. Thus, weighting the shot length with the inverse of the cluster's weight yields a measure of shot importance. The importance I of the jth shot is

$$I_j = \frac{L_j}{L_{max}} \log \frac{1}{W_k} \tag{6}$$

where L_j is the length of the jth shot and L_{max} is the maximum length of the kth cluster shot. By combining Eq. (4) and the shot importance (Eq. (6)), the paneling score for the key frames of the jth shot is defined as follows:

$$F_j = \lambda_4 I_j + \lambda_5 E_j \tag{7}$$

where the weights are set as $\lambda_4 = 0.6$ and $\lambda_5 = 0.4$ to balance the contribution of different terms.

To determine the page label associated with key frames, we utilize the Euclidean distance from the Eq. (7) to the paneling score of the layout template.

$$\min_n \sum_j |F_j - \Omega_n(j)|^2 \tag{8}$$

where $\Omega_n(j)$ is the jth paneling score of the nth template comic page ($large\ panel = 3.0$, $middle\ panel = 2.0$, $small\ panel = 1.0$).

In addition, we consider the number of comic panels for the summary. Realizing the level of detail control for users, Kasamatsu [7] reduces the displayed key frames by eliminating less important frames. This technique automatically manages the process, by comparing adjacent two key frames in time, and eliminating the frame with the lower importance score. However, this method does not focus on the key frame's positions in anime films, and there is a high probability that this method will lead to the deflection of key frames. Thus, to prevent key frame deflections, we divide the input anime film into four categories of comic-based composition: introduction, development, turn, and conclusion. In each category, we reduce the number of comic panels (displayed key frames) using Eq. (7).

4 Evaluation

We verify the effectiveness of our key frame detection method using an accuracy evaluation. To evaluate anime film summarization, we propose a criteria that measures the matching accuracy of key frames and original comic panels (precision rate, recall rate and F-measure). In the experiments, we use 20 different types of Japanese anime films, including martial arts, romantic comedy, mystery, and fantasy. These anime films are among the top 20 films in annual sales of DVD and comic book. The ground truths of key frames are manually labeled according to the original comic panels.

For comparison, we use Kasamatsu's [7] method, wherein key frames are the center frame in each shot sequence (SC), and the equal interval method, wherein the key frames are detected by equal time intervals in an anime film (EQL). The results of key frame detection are shown in Table 1. These results indicate that our key frame detection method provides higher accuracy than the previous methods. In contrast to the previous methods, our precision rate is higher than our recall rate. Consequently, false-positive detection of key frames is significantly reduced. Moreover, the recall rate for our method is higher than that of SC; this occurred because the SC method does not consider the frame importance of each shot. The results indicate that our method successfully improves the accuracy of key frame detection.

Table 1. Key frame detection result.

	Precision (%)	Recall (%)	F-measure (%)
Our method	83.47	80.17	81.17
SC	41.95	76.76	53.15
EQL	56.58	58.46	56.99

In addition, we apply our key frame detection to original anime films *'Spirited Away'* and *'Puella Magi Madoka Magica,'* which were released as movies before serialized in a comic magazine. These mean scores of key frame detection were $Precision = 77.62\,\%$, $Recall = 88.59\,\%$, and $F-measure = 82.74\,\%$. These results show that our method can create highly accurate summaries for any anime films without original comics. This evaluation measure represents how well the method generates video summaries in an aspect of completion of an original comic from the anime film.

5 Conclusions and Future Work

We have presented Computational Cartoonist, a video summarization system that generates key-frame-based video summaries from anime films. We define the correspondence between the anime's key frames and the original comic panels. In the future, we are planning to take high-level image features such as character faces to understand the specific character's movement in anime films. In addition, we append to add speech bubbles and sound effects (e.g., onomatopoeia) using acoustic features or physics parameters [6]. Furthermore, we are planning to help users review the information they need, and create richer comic summaries by using subtitle features.

Acknowledgements. This research is supported in part by OngaCREST, CREST, JST.and by Research Fellowship for Young Scientists of Japan Society for the Promotion of Science (JSPS).

References

1. Boreczky, J., Girgensohn, A., Golovchinsky, G., Uchihashi, S.: An interactive comic book presentation for exploring video. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 185–192. ACM (2000)
2. Calic, J., Gibson, D.P., Campbell, N.W.: Efficient layout of comic-like video summaries. IEEE Trans. Circ. Syst. Video Technol. **17**(7), 931–936 (2007)
3. Cao, Y., Chan, A.B., Lau, R.W.: Automatic stylistic manga layout. ACM Trans. Graph. (TOG) **31**(6), 141 (2012)
4. 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)
5. Farneback, G.: Very high accuracy velocity estimation using orientation tensors, parametric motion, and simultaneous segmentation of the motion field. In: Eigth IEEE International Conference on Computer Vision, ICCV 2001, Proceedings, vol. 1, pp. 171–177. IEEE (2001)
6. Fukusato, T., Morishima, S.: Automatic depiction of onomatopoeia in animation considering physical phenomena. In: Proceedings of the Seventh International Conference on Motion in Games, pp. 161–169. ACM (2014)
7. Kasamatsu, S., Itoh, T.: A browser for summarized multiple videos. In: Proceedings of the 8th NICOGRAPH International (2009)
8. Levi, K., Weiss, Y.: Learning object detection from a small number of examples: the importance of good features. In: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2004, vol. 2, pp. II-53. IEEE (2004)

9. Lian, S., Dong, Y., Wang, H.: Efficient temporal segmentation for sports programs with special cases. In: Qiu, G., Lam, K.M., Kiya, H., Xue, X.-Y., Kuo, C.-C.J., Lew, M.S. (eds.) PCM 2010, Part I. LNCS, vol. 6297, pp. 381–391. Springer, Heidelberg (2010)
10. McCloud, S.: Making Comics: Storytelling Secrets of Comics, Manga and Graphic Novels Author. William Morrow Paperbacks, New York (2006)
11. Myodo, E., Ueno, S., Takagi, K., Sakazawa, S.: Automatic comic-like image layout system preserving image order and important regions. In: Proceedings of the 19th ACM International Conference on Multimedia. pp. 795–796. ACM (2011)
12. Ryu, D.S., Park, S.H., Lee, J.w., Lee, D.H., Cho, H.G.: Cinetoon: A semi-automated system for rendering black/white comic books from video streams. In: IEEE 8th International Conference on Computer and Information Technology Workshops, CIT Workshops 2008, pp. 336–341. IEEE (2008)
13. Shamir, A., Rubinstein, M., Levinboim, T.: Generating comics from 3d interactive computer graphics. IEEE Comput. Graph. Appl. **26**(3), 53–61 (2006)
14. Taniguchi, Y., Akutsu, A., Tonomura, Y.: Panoramaexcerpts: extracting and packing panoramas for video browsing. In: Proceedings of the fifth ACM International Conference on Multimedia, pp. 427–436. ACM (1997)
15. Thawonmas, R., Shuda, T.: Comic layout for automatic comic generation from game log. In: Ciancarini, P., Nakatsu, R., Rauterberg, M., Roccetti, M. (eds.) New Frontiers for Entertainment Computing. IFIP International Federation for Information Processing, vol. 279, pp. 105–115. Springer, Heidelberg (2008)
16. Tobita, H.: Comic engine: interactive system for creating and browsing comic books with attention cuing. In: Proceedings of the International Conference on Advanced Visual Interfaces, pp. 281–288. ACM (2010)
17. Uchihashi, S., Foote, J.: Summarizing video using a shot importance measure and a frame-packing algorithm. In: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 6, pp. 3041–3044. IEEE (1999)
18. Uchihashi, S., Foote, J., Girgensohn, A., Boreczky, J.: Video manga: generating semantically meaningful video summaries. In: Proceedings of the Seventh ACM International Conference on Multimedia (Part 1), pp. 383–392. ACM (1999)
19. Wang, M., Hong, R., Yuan, X.T., Yan, S., Chua, T.S.: Movie2comics: towards a lively video content presentation. IEEE Trans. Multimedia **14**(3), 858–870 (2012)
20. Zhuang, Y., Rui, Y., Huang, T.S., Mehrotra, S.: Adaptive key frame extraction using unsupervised clustering. In: 1998 International Conference on Image Processing, ICIP 1998, Proceedings, vol. 1, pp. 866–870. IEEE (1998)