

Enabling Portable Animation Browsing by Transforming Animations into Comics

Wei-Ta Chu
National Chung Cheng University
Chiayi, Taiwan
wtchu@cs.ccu.edu.tw

Hsing-Han Wang
National Chung Cheng University
Chiayi, Taiwan
xinghan177@gmail.com

ABSTRACT

This paper presents a media transformation system that transforms animations into comics. The comic presentation not only enhances the efficiency of animation browsing, but also enables users to browse animations just like reading comic books, which is one of the favorite experiences in many people's childhood. We first select representative keyframes, and then devise an optimization framework based on the genetic algorithm to determine how keyframes are grouped and are arranged into the same comic pages. For keyframes in the same page, shot characteristics and keyframe importance are jointly considered to determine the best presentation layout. Subjective evaluation results demonstrate the attractive comic pages please readers and may facilitate efficient browsing and management for animations.

Categories and Subject Descriptors

H.5.1. [Information Interfaces and Representation]: Multimedia Information Systems – *Animations, evaluation/methodology, video*. I.2.10 [Artificial Intelligence]: Vision and Scene Understanding – *Representations, data structures, and transforms, video analysis*.

General Terms

Algorithms, Performance, Experimentation.

Keywords

Genetic algorithm, layout selection, animations, comics.

1. INTRODUCTION

The emergence of large volumes of video data urges efficient video browsing technologies. Video segmentation techniques that split videos into segments with concrete and clear semantics, and video summarization/highlight techniques that provide compact representations, thus play more and more important roles for video browsing and management. The request for efficient browsing with a limited-sized interface becomes increasingly urgent as high-definition portal devices like tablet PCs are popularized in recent years. In this paper, we discuss how to transform an important type of videos, i.e., animations, into a

compact representation, so that animations can be efficiently browsed on portable devices with high interactivity.

Although large amounts of approaches have been proposed to summarize news videos, home videos, movies, and sports videos, relatively fewer researchers pay attention to animations. Recently, more than 2000 hours of animations are produced every year only in Japan. Many comic fans keep their eyes on the newly-released episodes every week, and are often eager to quickly browse the latest content. In this case, converting animations into comics provides a compact presentation, especially comics are widely accepted to present information in various cultures and regions. Hong et al. [7] have similar ideas on transforming movies into comics. They mainly put efforts on key scene selection and cartoonization, but in this work we focus on developing a framework to find optimal page allocation and layout selection.

Another reason that we focus on animation is quite intuitive. Reading comic books would be one of the favorite experiences in many people's childhood. It would be an efficient and broadly acceptable manner to browse animations by just sliding our fingers to turn comic pages. Therefore, transforming animations into comics is not only a process that conducts modality transformation and media adaption, but also enhances interactivity and efficiency in browsing animations on portable devices.

In contrast to thumbnails and video mosaics, comics as a kind of static presentation provide richer information about video pace and highlighted objects. Figure 1 shows sample pages of our results. There may be different numbers of frames in different pages, depending on the video pace. Varied space may be allocated to frames in the same page. To fit a frame into a cell, less important part of a frame may be removed so that the aspect ratio of the remaining region matches with that of the targeted cell. In summary, we formulate the following four problems needed to be solved to transform an animation into a comic.

- Which frames should be selected? In a 30-min animation, for example, we have to select just hundreds of representative frames from tens of thousands of video frames, so that information can be efficiently presented.
- Among the selected frames, which frames should be put into the same page? Frames representing the same event or the same place are better to present together. Furthermore, by varying the number of frames in a page, we can control "the pace of comics." For example, we feel that the top row of Figure 1 shows a higher pace than the bottom row, because content in the former one changes more frequently.
- What is the best layout to display frames in the same page? We design page layouts that contain varied-size cells. If K frames are determined to put into a page, the layouts

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

IMMPD'12, November 2, 2012, Nara, Japan.

Copyright 2012 ACM 978-1-4503-1595-1/12/11...\$15.00.

containing K cells are selected. Intuitively, we tend to allocate more space for more important frames.

- Which region in a frame is best for the corresponding cell? The aspect ratio of the targeted cell may be different from the aspect ratio of a frame. Crudely resizing the frame would distort important content in it. Therefore, we would need to find a region that has the same aspect ratio as the targeted cell, and meanwhile contains important content in this frame.

In this paper we mainly focus on the second and the third problems described above. We propose a framework based on the genetic algorithm for optimized page allocation. For layout selection, we describe predefined layouts and characteristics of keyframes in shots as vectors, and find their best matching to determine the most appropriate display layout. Optimized page allocation and layout selection constitute the main contribution of this paper.

The rest of this paper is organized as follows. Section 2 provides brief literature survey, and Section 3 describes system framework and processing. In Section 4, we transform the second issue as an optimization problem and exploit the genetic algorithm to find solutions. Corresponding to the third issue, Section 5 describes how to select the most appropriate layout. Section 6 describes page composition, Section 7 provides experimental results, and Section 8 concludes this paper.



Figure 1. Sample pages of the generated comics. Top: generated from “One Piece”; Bottom: generated from “The Squid Girl.”

2. RELATED WORKS

2.1 Video Summarization

Video summarization has been widely studied for years, and it can be roughly categorized into two types: static video abstract and video skimming. Static video abstract is also called a static storyboard, and it collects the most relevant images or keyframes extracted from the original video sequence. On the other hand, dynamic video skimming focuses on selecting most informative short clips to compactly represent the original video.

There are many studies on summarization for various types of videos, such as summaries of home videos [9], events in surveillance videos [10], and skimming for movies [11]. Considering human perception, Ma et al. [1] exploit the user attention model to examine salient parts of a video and

concatenate them as a video summary. Focusing on sports videos, Tjondronegoro et al. [2] present a framework that integrates game highlights into plays and shows complete sports video summarization. Due to the wide interests of video summarization, TRECVID [3] poses challenges on rush video summarization in years 2007 and 2008, and inspire broad ranges of studies.

2.2 Media Representation Transformation

To facilitate efficient video browsing, many researchers transform visual content in one modality into another modality. The simplest representation transformation scheme is video storyboard. For example, Ciocca et al. [12] transform videos into multilevel storyboards that are constituted by hierarchical representation of keyframes. Storyboard presentation is also provided by commercial software, such as KMPlayer¹.

Another presentation type is collage, which sticks many photos in a large image. Wang et al. [13] propose a video collage system that automatically constructs a compact and visually appealing synthesized collage from a video sequence. With similar concepts, Pritch et al. [10] propose a video synopsis that provides a static presentation showing essential activities of a video.

In the animation movie industry, to prompt animation movies, transforming part of animation into colorful comics is a common way. The colorful comics are one of the popular peripheral merchandise for animation fans. However, colorful comics are specially edited for a very limited number of animation, by professional editors spending a lot of time to extract appropriate frames from videos.

Hong et al. [7] propose the Movie2comics system that provides comic-like presentation to show movie’s keyframes. They mainly put efforts on three components: script-face mapping, key-scene extraction, and cartoonization. While we have similar concept to the Movie2comices system, in this paper we focus on developing a framework to find optimal page allocation schemes and select the most appropriate presentation layouts.

3. SYSTEM FRAMEWORK

Figure 2 shows the flowchart of the proposed system. We first segment the animation into shots, and then select keyframes as the representation of each shot. To address the second problem mentioned in Sec. 1, we consider the principles of moviemaking. Similar to movies that have clear story structure, comic presentation also requires clear section division. Scene detection is thus conducted for the animation. We then allocate appropriate number of comic pages that may include various numbers of cells and to display the content in scenes. We transform this challenge as an optimization problem and devise a method based on the genetic algorithm to solve it. To address the third problem, we estimate importance value for each keyframe and each layout, and then determine the most appropriate layout by matching the importance distribution (represented by vectors) between them. Finally, after determining the region of interest in each keyframe, appropriate cropping and resizing are conducted, and the comic version of the animation is generated by composing resized keyframes together.

We briefly describe methods for shot change detection and scene change detection in the following. For a given animation, we first

¹ The KMPlayer, <http://www.kmplayer.com/forums>

detect shot boundaries by checking RGB color histogram difference and edge change ratio [8] between temporally consecutive frames. For each video shot, the frame right at the middle of the shot is first selected as in the keyframe set. A frame in the same shot that has a large distance to all of the current selected keyframes is also put into the keyframe set. By repeating this process, all video shots are represented by varied numbers of keyframes.

Similar to movies that have clear story structure, comic presentation also requires clear section division. A series of semantics-related video shots, i.e., video scenes, provide important clues for us to arrange comic pages for different story units. Based on color histograms extracted from keyframes, we calculate backward shot coherence (BSC) for each shot [4]. The BSC value represents content similarity between the current shot and previous shots in a temporal locality, which mimics short-term memory of the human’s visual system. A shot with a BSC value lower than a threshold is detected as the start of a new scene. The detected scene change candidates are further examined by shot motion content, because high-motion shots often infer false alarms and should not be claimed as scene change boundaries [4].

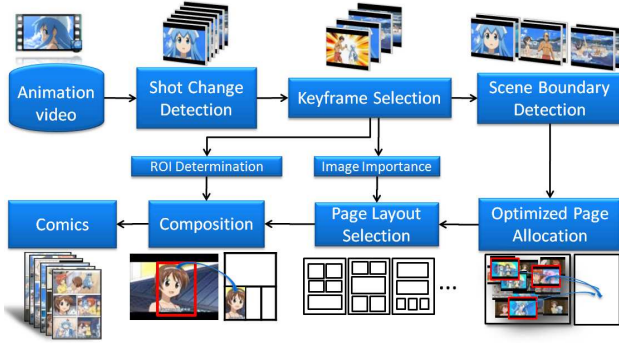


Figure 2. Flowchart of the proposed system.

4. OPTIMIZED PAGE ALLOCATION

With the scene boundary information, we would like to arrange a number of comic pages to keyframes in the same scene. Assume that there are N shots S_1, \dots, S_N in a scene. In this component we would like to allocate appropriate number of comic pages that may include various numbers of cells and exhibit the following characteristics:

- Keyframes put into the same page likely present consecutive or similar visual content, such as color and motion magnitude. Of course keyframes in the same video shot should be put into the same page.
- If video shots present high video tempo (e.g., higher shot change frequency or higher motion), more corresponding keyframes should be placed into the same page to build tense browsing experience.

Basically this task is a labeling problem, in which N items are needed to be labeled as appropriate number of classes. There may be tremendous number of labelings. Fortunately, in our work the temporal continuity of consecutive shots should be maintained. For example, if five shots are needed to be labeled, the labeling 11223 is legal, but the labeling 12213 is not allowed, in which each number indicates the ID of a class. Although the number of

feasible labelings is largely reduced by this constraint, we further exploit the genetic algorithm (GA) to efficiently find the optimal assignment.

Before the GA process, we randomly generate initial chromosomes, each of which consists of a label sequence $x = (x_1, x_2, \dots, x_N)$, where $x_i = n, n \in \{1, 2, \dots\}$, denotes that the shot S_i is assigned to the n th comic page. Note that if $x_i \neq x_{i+1}, x_i = x_{i+1} - 1$. The first shot x_1 is always assigned to the first page, i.e., $x_1 = 1$. Because a page should not include too many shots, in this work we allow at most R shots assigned into the same page. Therefore, an initial chromosome is generated by the following process.

- (1) Randomly pick a number from 1 to R , say m_1 , and assign the first m_1 shots into the first page, i.e., $x_1 = x_2 = \dots = x_{m_1} = 1$.
- (2) Randomly pick a number from 1 to R , say m_2 , and assigned the following m_2 shots into the second page, i.e., $x_{m_1+1} = x_{m_1+2} = \dots = x_{m_1+m_2} = 2$.
- (3) The same process repeats until all shots have been assigned.

Figure 3 (a) shows an example of a chromosome where there are 22 shots in the same scene. Each block represents a shot, and is also viewed as a gene in the chromosome. Each gene is labeled as a number, which denotes the corresponding page a shot is assigned. The corresponding page arrangement is shown in Figure 3(b). The chromosome represents that the 22 shots are arranged into 4 pages. The first page has 7 shots, the second page has 4 shots, the third page has 6 shots, and the fourth page has 5 shots.

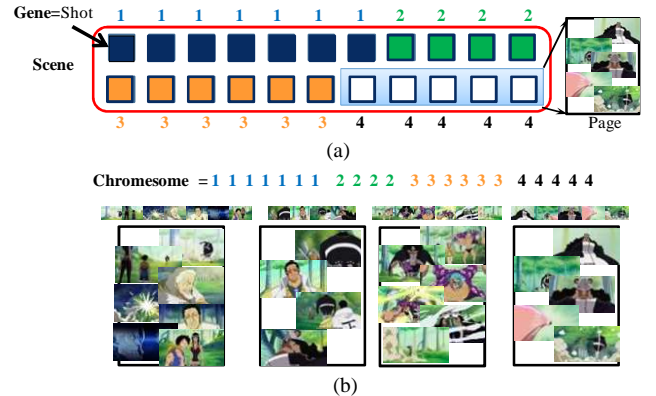


Figure 3. (a) An example of chromosome and (b) the corresponding page arrangement.

Let $\{P_1, P_2, \dots, P_M\}$ denote the page assignment for a scene, in which $P_i = \{S_j | x_j = i\}$ consists of shots assigned to the i th page. To quantify the factors described in the beginning of this section, the average color histogram similarity corresponding to a chromosome is defined as

$$A_c = \frac{1}{M} \sum_{i=1}^M \left(1 - \frac{\min_{S_a \in P_i, S_b \in P_i} d_c(S_a, S_b)}{\max_{\substack{S_a \in P_i \\ S_b \in P_{i-1} \text{ or } S_b \in P_{i+1}}} d_c(S_a, S_b)} \right), \quad (1)$$

$$d_c(S_a, S_b) = \max_{f_a \in S_a, f_b \in S_b} CHD(f_a, f_b). \quad (2)$$

The terms f_a and f_b denote the keyframes selected from S_a and S_b , and the function $CHD(f_a, f_b)$ calculates the RGB color histogram difference between two keyframes. The value A_c is

designed that shots in the same page should be visually similar, but shots in different pages should be visually distant.

The average motion similarity corresponding to a chromosome is defined as

$$A_m = \frac{1}{M} \sum_{i=1}^M \left(1 - \frac{1}{Z} \sum_{S_a \in P_i, S_b \in P_i} MHD(S_a, S_b) \right), \quad (3)$$

where $Z = \binom{|P_i|}{2}$, and $|P_i|$ denotes the number of shots in page P_i . The function $MHD(S_a, S_b)$ calculates the motion histogram difference between two shots.

If shots change frequently, we tend to put more shots into the same page to build higher pace. In other words, we tend to allocate pages such that the total length of shots in each page is similar. Therefore, this factor is defined as

$$A_t = 1 - std(T_1, T_2, \dots, T_M), \quad (4)$$

$$T_i = \frac{\tau_i}{\max(\tau_1, \dots, \tau_M)}, \quad (5)$$

where τ_i is the total length of shots in the page P_i , and the $std(\cdot)$ function calculates the standard deviation.

The fitness (objective function) of the chromosome x in the GA process is then defined as a linear combination of these three factors:

$$f(x) = \alpha A_c + \beta A_m + \gamma A_t, \quad (6)$$

where $\alpha + \beta + \gamma = 1$.

We randomly generate eight chromosomes as the initial population $\Omega(0)$. At each iteration k of the GA process, we evaluate the fitness $f(x^{(k)})$ of each member $x^{(k)}$ of the population $\Omega(k)$. Based on the fitness, we select a set $\Phi(k)$ with the same number of elements as $\Omega(k)$ by the roulette-wheel scheme, i.e., chromosomes are selected with probabilities proportional to their fitness. The operations of crossover and mutation are then applied to elements of $\Phi(k)$ to generate a new population $\Omega(k+1)$. Note that from the resulting chromosomes, only the eight ones with the highest fitness are selected to constitute the new population. The GA process keeps running until the number of iteration reaches a predefined value, say 50 in this work.

Figure 4 shows two examples of the average, worst, and best objective functions values at each iteration of the GA process. With the designed features and evolution operations, the process converges quickly and the best labeling can be determined efficiently.

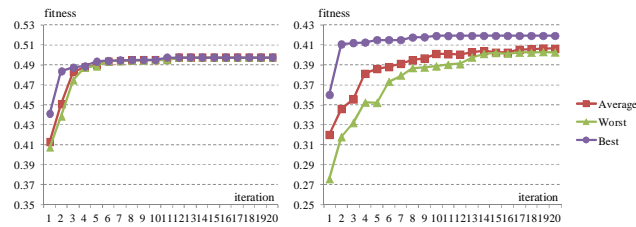


Figure 4. The average, worst, and best objective function values at each iteration of the GA process. Left: “The Squid Girl”; Right: “One Piece.”

5. LAYOUT SELECTION

The GA process finally determines the optimal labeling. For example, the final chromosome $x^* = (1112233444)$ means that the first three shots are put into the first page, and the fourth and the fifth shots in the second page, and so on.

The next step is to find the best layout to display shot information. Assume that there are totally K keyframes in the shots at the same page. We would like to find the best layout that has been designed in advance and contains K varied-sized cells. Each cell is a room allocated to display a keyframe. For the K keyframes, the desired best layout should have the following characteristics: 1) More important keyframes should be allocated into larger cells; 2) keyframes in the same shot are preferred to put at the same row of cells.

To tackle with the first issue, we characterize each layout by a vector: $\ell_i = (a_1, a_2, \dots, a_K)$, where a_j is the ratio of the area of the j th cell to the area of the whole page. Similarly, we characterize the K keyframes by $f = (\alpha_1, \alpha_2, \dots, \alpha_K)$, where α_j is the ratio of the area of region-of-interest (ROI) to the area of the whole j th keyframe. The ROI is determined by the method proposed in [6].

To find how appropriately a layout matches with the given keyframe sequence, we refer to the inner product matching scheme proposed in [5], and determine the best layout ℓ_{i^*} by

$$i^* = \arg \max_i \langle \ell_i, f \rangle. \quad (7)$$

To further consider the second issue, we would like to describe how cells in a layout are arranged into rows and represent it as a K -dimensional binary vector. If the i th cell is in the j th row, and the $(i+1)$ th cell is in the $(j+1)$ th row, the bit at the $(i+1)$ th dimension is set as 1, otherwise 0. Figure 5 shows three sample layouts containing five cells. The cell arrangement for the first layout is $r_1 = (00011)$, because the fourth cell is located at the next row to that for the third cell, and the fifth cell is located at the next row to that for the fourth cell. Similarly, the cell arrangements for the second and third layouts are $r_2 = (00110)$ and $r_3 = (01000)$.

Assume that $K = 5$, and if the five keyframes to be placed into a 5-cell layout belong to three shots, say $S_1 = \{f_1, f_2\}$, $S_2 = \{f_3, f_4\}$, and $S_3 = \{f_5\}$. We also transform how keyframes belong to different shots into a binary vector. If the i th keyframe is in the j th shot, and the $(i+1)$ th keyframe is in the $(j+1)$ th shot, the bit at the $(i+1)$ th dimension is set as 1, otherwise 0. Therefore, in this example, the keyframe sequence is represented as $q = (00101)$. How different cell arrangements fit with shot characteristics is described as

$$w_i = 1 - \frac{XOR(q, r_i)}{K}. \quad (8)$$

This value represents the degree of matching between the situation of shot changes and the situation of keyframes putting in different rows. This value ranges from 0 to 1, and a larger value means higher matching.

Finally, by jointly considering keyframe importance and cell arrangement, the best layout ℓ_{i^*} is actually determined by

$$i^* = \arg \max_i \langle \ell_i, f \rangle \times w_i. \quad (9)$$

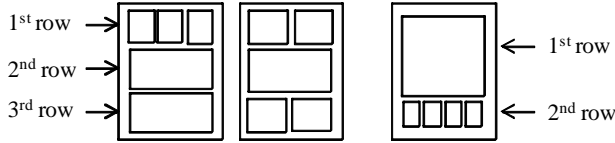


Figure 5. Three sample layouts containing five cells.

6. COMPOSITION

We have determined where each keyframe should be placed after the aforementioned process. As we see from Figure 5, different cells may have significantly different aspect ratios, and undoubtedly directly resizing each keyframe and pasting it into the corresponding targeted cell would largely distort visual information. Therefore, from each keyframe, it is necessary to find an important region that simultaneously covers salient objects and meets the aspect ratio of the targeted cell.

We accomplish this task by a method similar to that proposed in [5]. The composition process is illustrated in Figure 6. First, based on global color contrast [6] a saliency map is constructed to represent saliency characteristics of a keyframe. The centroid of the saliency map is accordingly determined. Starting from this position, we expand the region towards four directions (top, down, left, right) according to the aspect ratio of the targeted cell. The expansion stops when at least two boundaries of this region reach the boundaries of the keyframe. The selected region is then resized to stick on the targeted cell. After sticking regions from keyframes on all cells, a comic page is finally generated. Figure 1 shows two sample results.

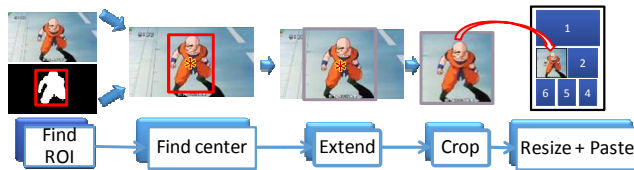


Figure 6. Illustration of the composition process.

Table 1. Information of the evaluation dataset.

ID	Name	Avg. length	Avg. #pages
1~3	The Squid Girl	6 min 52 sec	24.67
4~6	Crayon Shin-chan	7 min 6 sec	19.33
7~8	Dragon Ball	19 min 27 sec	58.5
9~11	Chibi Maruko Chan	20 min 43 sec	58.67
12~13	One Piece	18 min 52 sec	47
14	Tiger X Dragon	18 min 35 sec	63

7. EXPERIMENTS

Table 1 shows six animations, containing totally 14 episodes, in the evaluation dataset, mainly Japanese animations. We invited 17 subjects, including 12 males and 5 females, to evaluate how they feel from the generated comic pages. They are all students major in computer science, and are all familiar with animation and comics. They were asked to watch the original animation, browse the generate comics, and finally give scores from 1 to 5, where larger score means higher satisfaction. The experiments are divided into two parts. In the first part, we designed four questions to evaluate performance of the overall system:

Q1: Do you think the generated comic presentation is more interesting than the storyboard presentation generated by the software KMPlayer?

Q2: Do you think the generated comic presentation well conveys information in the original video?

Q3: Do you think the generated comic presentation concisely presents information of the original video?

Q4: Do you like to use the generated comics as the preview that assists you to decide whether to watch the original video?

For each question, subjects give one score for each comic book generated from an episode. Table 2 shows the average opinion scores for the 14 generated comics. Comparing with conventional storyboard presentation, most users prefer our comic presentation (Q1). The comic presentation obtains very high agreement to be a good preview of the original video (Q4).

The second part of experiments is designed to evaluate whether the system has good performance on page arrangement, cell arrangement, and ROI determination.

Q5: Do you think the keyframes putting at the same pages are appropriate?

Q6: Do you think the layouts for displaying keyframes are appropriate?

Q7: Do you think the content displayed in cells is appropriate? (evaluating the composition process)

Table 3 shows the average opinion scores for these three questions. High satisfaction is obtained for page arrangement (by the proposed optimization process, Q5) and cell arrangement (by the proposed layout selection process, Q6). We obtain slightly worse performance on Q7. This may be because the saliency analysis method proposed in [6] is mainly designed for natural images. Therefore, in the future we will investigate more about ROI determination in animation.

We show average opinion scores over eight questions for these fourteen animations in Figure 7, respectively. From this result we see the worst performance is obtained for ‘‘Crayon Shin-chan.’’ The video quality for the three episodes (IDs from 4 to 6) is the poorest in the evaluation dataset. This result drives us to exploit more advanced features for describing frames in the future, instead of just color histogram and motion information.

Figure 8 shows some samples of displaying the generated comics on portable devices. With the design of page allocation and layout selection, the proposed system presents animation in an efficient way. Informative content, the pace of the original animation, and important regions of frames are preserved and displayed in limited-sized space. Users can browse animations just like reading electronic comic books. Moreover, the comic presentation also serves as a type of preview for the original animation, especially when bandwidth of wireless communication is limited so that downloading the whole animation takes much time.

Although this work is motivated by efficient animation browsing with comic presentation, we have to note that the designed framework is actually not limited to animations. We also try to transform movies and dramas into comics, and the results are also promising. Analyzing other videos would need further designs on visual features and objective functions, and thus we do not extensively claim the generality of the proposed framework right now.

Table 2. Subjective evaluation results for the first experiment.

	Q1	Q2	Q3	Q4
Avg. scores	3.78	3.63	3.83	4.29

Table 3. Subjective evaluation results for the second experiment.

	Q5	Q6	Q7
Avg. scores	3.70	3.64	3.37

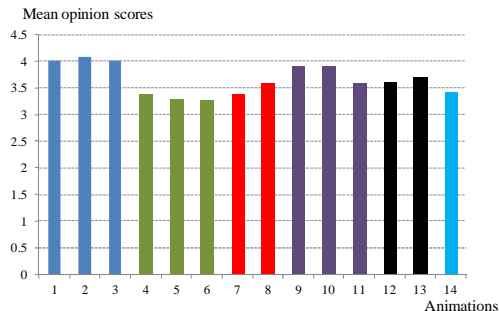


Figure 7. Mean opinion scores for the 14 animations.

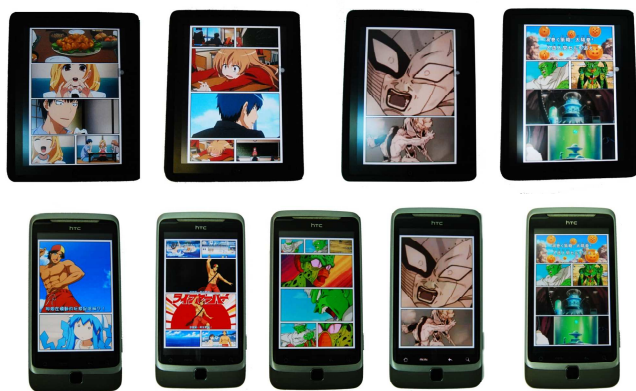


Figure 8. Displaying the generated comics on portable devices. Top: display on tablet PCs; bottom: display on mobile phones.

8. CONCLUSION

We have presented a system that automatically transforms animations into comics. After shot change, scene change, and keyframe selection, we propose an optimization framework to determine how keyframes are arranged into pages. Based on saliency analysis and shot characteristics, we devise a layout selection scheme to determine the space allocated to each keyframe. Optimized page allocation based on the genetic algorithm and layout selection based on vector form representation constitute the major contribution of this paper. The results of subjective evaluation show that the proposed systematic approaches generate attractive comic presentations, which would facilitate efficient browsing and management functionalities for large volumes of animations.

In the future, we would like to incorporate subtitle information and design an optimization method to appropriately locate speech bubbles so that the generated comics are more readable. Usability study for showing comics on portable devices will also be conducted.

Acknowledgement

The work was partially supported by the National Science Council of Taiwan, Republic of China under research contract NSC 101-2221-E-194-055-MY2.

9. REFERENCES

- [1] Ma, Y.-F., Lu, L., Zhang, H.-J., and Li, M. 2002. A user attention model for video summarization. In Proc. of ACM Multimedia, 533-542.
- [2] Tjondronegoro, D., Chen, Y.-P. P., and Pham, B. 2004. Highlights for more complete sports video summarization. IEEE Multimedia, vol. 11, no. 4, pp. 22-37.
- [3] Smeaton, A. F., Over, P., and Kraaij, W. 2006. Evaluation campaigns and TRECVID. In Proc. of ACM International Workshop on Multimedia Information Retrieval, pp. 321-330.
- [4] Rasheed, Z., and Shah, M. 2003. Scene detection in Hollywood movies and tv shows. Proc. of CVPR, vol. 2, pp. 343-348.
- [5] Chen, J.-C., Chu, W.-T., Kuo, J.-H., Weng, C.-Y., and Wu, J.-L. 2006. Tiling slideshow. In Proc. of ACM Multimedia, pp. 25-34.
- [6] Cheng, M.-M., Zhang, G.-X., Mitra, N.J., Huang, X., Hu, S.-M. 2011. Global contrast based salient region detection. Proc. of CVPR, pp. 409-416.
- [7] Hong, R., Yuan X.-T., Xu, M., Wang, M., Yan, S., and Chua, T.-S. 2010. Movie2Comics: A feast of multimedia artwork. In Proc. of ACM Multimedia, 611-614.
- [8] Lienhart, R. 1999. Comparison of automatic shot boundary detection algorithms. In Proc. of SPIE Storage and Retrieval for Image and Video Databases VII, vol. 3656, pp. 290-301.
- [9] Takeuchi, Y., and Sugimoto, M. 2007. User-adaptive home video summarization using personal photo libraries. In Proc. of ACM International Conference on Image and Video Retrieval, pp. 472-479.
- [10] Pritch, Y., Rav-Acha, A., and Peleg, S. 2008. Nonchronological video synopsis and indexing. IEEE Transactions Pattern Analysis and Machine Intelligence, vol. 30, issue 11, pp. 1971-1984.
- [11] Li, Y., Lee, S.-H., Yeh, C.-H., and Kuo, C.-C.J. 2006. Techniques for movie content analysis and skimming: tutorial and overview on video abstraction techniques. IEEE Signal Processing Magazine, vol. 23, pp. 79-89.
- [12] Ciocca, G., and Schettini, R. 2006. Dynamic storyboards for video content summarization. In Proc. of ACM International Workshop on Multimedia Information Retrieval, pp. 259-268.
- [13] Wang, Y., Mei, T., Wang, J., and Hua, X.-S. 2010. Dynamic video collage. In Proc. of International Conference on Multimedia Modeling, LNCS 5916, pp. 793-795.