

---

# DigestManga: Interactive Movie Summarizing through Comic Visualization

## Hiroaki Tobita

Interaction Laboratory  
Sony Computer Science  
Laboratories, Inc.  
3-14-13 Higashigotanda,  
Shinagawa-ku  
Tokyo 141-0022 Japan  
tobita@csl.sony.co.jp

## Abstract

DigestManga is a system for interactively integrating movies with comics. Several systems already exist for visualizing input movies as comic books to allow users to see several frames of a movie at the same time through a comic visualization. However, their intention is mainly to represent a movie with visual summaries. Although users can edit the visualized comic book to arrange the summaries, the manipulations are not reflected in the input movie.

---

Copyright is held by the author/owner(s).

CHI 2010, April 10–15, 2010, Atlanta, Georgia, USA.

ACM 978-1-60558-930-5/10/04.

Our DigestManga system provides both visual movie summaries based on a comic and summarized movies through comic editing. Our system connects the comic and movie smoothly and interactively, because we think comic visualization is useful for movie editing and makes movies more interesting.

## Keywords

Comic and manga visualization, movie editing, contents deformation, interactive system

## ACM Classification Keywords

H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

## General Terms

design

## Introduction

Comics are a graphic art form that combines text and images to effectively represent a story [1]. Visual movie summaries based on comics are one of the comic visualization applications [2, 3]. The main advantage of these systems is to see several frames at the same time, because each page of a comic book contains several frames of an input movie. In addition, by using these systems, users can print out the movie content as paper material, so they can see the movie without using a computer. Although this approach is useful for

better understanding the movie content more quickly than if they watched the movie, but there is no support for editing the input movie through the visualized comic. We think comic visualization is useful for movie editing because comics are created frame sets along a timeline and the comic frame is a directory connected to the movie frame in the visualization. By using this comic visualization and its functions, such as speech bubbles and focus lines, users can also make movies more interesting and more original than the input movie. Thus, interaction techniques that combine comics with movies are useful for entertainment purposes.

The work presented in this paper introduces a system, called DigestManga, for interactively integrating movie contents with comic books. Although conventional systems focus on converting movies to comic books, our intention is to establish the smooth relationship between movies and comics. Thus, our goal is to provide both a visual summary of a movie and a summarized movie by using comic editing, so users can visualize an input movie as a comic book and edit the movie through the visualized comic. In order to visualize an input movie, the system provides both automated and manual approaches. In the automated approach, a comic book is automatically created from an input movie through image processes. In the manual approach, users create a comic by using simple manipulations like drag and drop. Also, the system provides several editing functions such as translating the comic frames and adding extra elements, such as speech bubbles and focus lines.

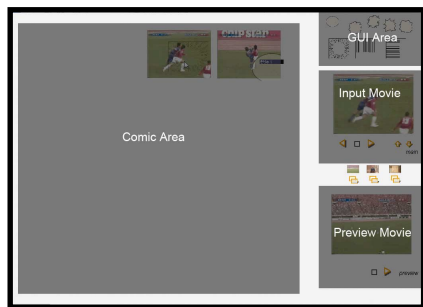
In addition, our system deforms the movie contents along with the comic visualization. Each comic frame on a page is a kind of impressive or feature scene for the

users, because the page is edited by the users and each frame is selected by them. Our deformation technique regards comic frames on a page as feature areas in an input movie. Our deformation is calculated using zooming techniques [6], so the movie contents are deformed without any information loss and naturally connected between the feature and non-feature areas. The contents in the non-feature areas are played more quickly, and the contents in the feature areas are played normally. As a result, users can quickly view the contents and understand the relationships between the entire story and the features.

## **RELATED WORK**

Comic-based systems have previously been developed. These systems are basically divided into two categories: supporting comic creation or comic visualization. Systems for comic creation focus on how to create a real comic using a computer, and systems for comic visualization focus on how to visualize complex data as a comic.

Comic visualization is useful for making complex data more easily understandable. Complex data is more interesting and fun when certain comic features such as a character, bubble, and focus line are used. For example, Comic chat [4] makes chat logs interesting and fun by including character-based comic representations. ComicDiary [5] is useful for making diaries by using predefined character and images. Our focus is a comic-based movie. Some existing systems can translate an input movie into a comic [2, 3]. These systems are very useful for converting animation movie data into a series of static images. By using comic like frame layouts, users can see many frames of a movie at the same time and can also print them out. Users



**Figure 1:** System Overview. The system has four areas: page, input movie, preview movie, and the GUI areas.

then might naturally want to use these systems to edit movies. However, these systems do not provide an editing function for movies through visualized comic books. Moreover, although comics are created automatically, they may not be sufficient enough for a user's intended designs or needs. Thus, interaction techniques for creating a smoother relationship between movies and comics are needed.

Our approach was inspired by some existing web services. Many web services for movie sharing allow users to add comments to a movie [7]. For example, NicoNico movie [8] is a movie-sharing site that allows users to add comments to a movie frame directory. A movie with added comments is very interesting and fun. These comments play the role of the audience or background sound, so the movie itself is improved. Our purpose was to achieve this kind of effect by using comic features such as speech bubbles and focus lines.

In addition, movie contents, such as recorded TV programs and home videos, are becoming pervasive in everyday life, and many people have large amounts of contents to deal with. It is important for users to know how to effectively deal with and understand these contents. Movie deformation is an effective way to roughly determine the nature of such contents to solve these problems. Many deformation techniques for movie contents are available especially in the product field [11]. As their deformation focuses on the feature areas, they create a digest by collecting only the feature areas. However, this creates two problems. Although users can quickly view the contents, they cannot understand the relationships between the entire story and the features. Moreover, their approach is

quite similar to image cropping in that the non-feature areas are discarded [9, 10]. The cropping approach eliminates some of the movie's initial information, so the total amount of information is reduced. This can make it difficult to distinguish between similar contents.

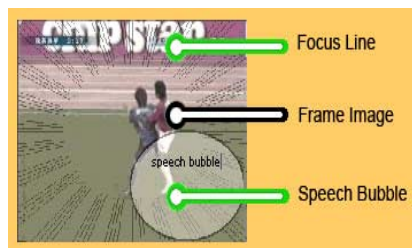
## SYSTEM DESIGN

An overview of our system is shown in Fig. 1. The system is divided into four areas: the comic, input movie, preview movie, and GUI areas. The comic area is used to edit a comic. Some GUIs to assist in editing the comic book are contained in the GUI area. Basically, this system can be used by drag and drop manipulations. After editing a movie, the user can view it by using the preview movie area.

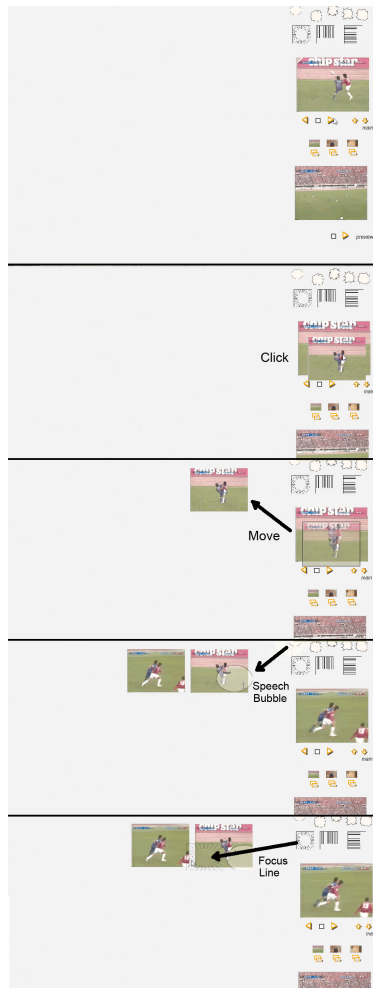
### *Visual Summaries based on Comic*

For converting a movie to a comic, the system provides two functions: automated and manual approaches. The automated approach is a static conversion, so comic books are created by the system automatically. In this case, the system automatically detects the feature scenes and sets several frames onto a page. By using the manual function, a user creates a comic by clicking the input movie area and dragging a current frame to a page area.

For both approaches, users can freely edit the comic by adding frames or extra elements (Fig. 2). As the system contains some template data related to these elements in the GUI area, users can use them through simple drag and drop manipulations. By using focus lines, features such as characters and building are clearer than with the original frames. Additional speech bubbles work like comments.



**Figure 2:** Movie area. The movie area has two layers: input movie and comic element layers.



**Figure 3:** Example of movie editing. Users can interactively change frame size and position and add extra elements.

#### *Editing Comic for Movie*

To create a concrete comic, a user edits the base comic by adding frames or extra elements for his/her design. Some examples of interactive comic editing are shown in Fig. 3. As a capture frame appears by using click manipulation onto a movie, the user drags a frame onto a page area. The frame becomes a comic frame on that page. The user can also control the frame size and position to add contrast between the frames and to emphasize the impressive frame for users. Generally, a big frame means that it is a more impressive or impacted part of the movie than normal. By adding extra elements, such as speech bubbles and focus lines, the user expresses more of his/her design. As a result, movie comic is created by simple manipulations (Fig. 4).

#### *Summarized Movie through Comic*

After editing the comic, the user can view the resulting edited input movie in the preview area. The preview area has two layers, the input movie and comic element layers (Fig. 2). In the movie layer, the system plays the input movie by controlling the play speed. In the comic element layer, the system displays the comic elements that were added. Our system treats comic frames on the page as features in a movie and other areas that do not appear on the page area as the non-features, so the system calculates the play speed along with the features.

We use fisheye calculations to control the speed. In this example, the movie is played normally using a normal frame size area and is played slowly in a large frame size area. Our current system is only a prototype, so the timing when extra elements appear and disappear is not fixed yet. In the current version, a comic element

appears five seconds before the current frame and disappears five seconds after the current frame.

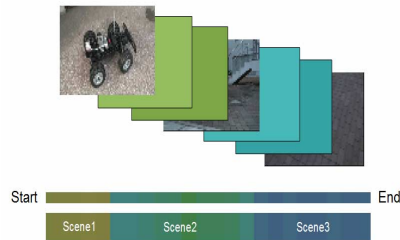


**Figure 4:** Example of movie comic. Users can create movie comic by simple manipulations.

#### **IMPLEMENTATION**

Each frame on a page has information such as the time (position of the timeline) of an input movie, the frame scale on a page, and additional elements (e.g., speech bubbles and focus lines). To manually set a comic, when a user clicks an input movie, the frame information is set and stored to a liner list. The current movie frame is connected to the play speed. For automated approaches, in order to detect scenes, the system calculates a histogram for each movie frame and compares the histograms of the current frame and the next frame (Fig. 5). If the difference is over a predefined threshold, the system recognizes that the scene has changed.

We use fisheye algorithm to control the play speed of movie, because the speed between the previous and current frames gradually changes. In this calculation, each play speed of a movie frame depends on the distance between the feature and current frames. In our approach, the feature area is played normally, and



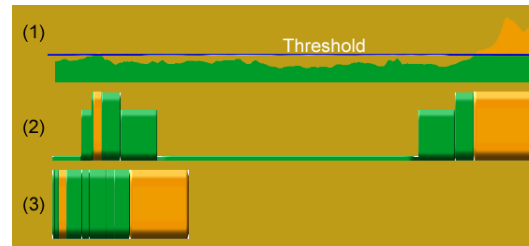
**Figure 5:** Scene detection. To detect scenes, the system calculates a histogram for each frame of the movie and compares the histograms.

the non-feature areas are played quickly depending on the distance between the frame and the feature frame. The total playing time can be as little as one-fourth that of the original movie. The frame speed calculation is defined by using the following.

$$s' = f(s).$$

$$f(s) = \left( \frac{dst_s}{src_s} s \right).$$

In the equations given above,  $src_s$  is the length of the total playtime of an input image and  $dst_s$  is the distance between the next feature and the current frame.



**Figure 6:** Deforming movie contents. The feature areas (orange) are played at normal speed and the non-feature areas (green) are played more quickly. The total playing time can be as little as one-fourth (3).

In this calculation, each play speed of movie frame depends on the distance between feature and current frames. In our approach, feature area is played normally, and non-feature areas are played quickly

depending on the distance between the frame and feature frame. The total playing time can be as little as one-fourth that of the original movie (Fig. 6).

## DISCUSSION

We have demonstrated our system many times. In particular, a demonstration of our system that was given during a laboratory open house prompted many interesting reactions and comments from visitors.

Our concept for making the input movie fun and interesting was well accepted by visitors because the users can get two kinds of summarized data simultaneously: comic summary and summarized movie.

Our goal of the system was to establish a smooth relationship between movies and comics. Our simple interactions received good reactions from users. We found that many visitors edited comic books from an input movie by using our system after our simple demonstration. They edited a comic book and checked the movie in the preview area over and over again to achieve their design. Also, some visitors commented that our interactions are too simple for use with more complex functions. Our target is not for professional designing, but for beginners, so our system provides simple edit functions for comic creation. However, several functions to make an input move more interesting and comical are necessary, so interaction techniques for freely changing and arranging a comic book are important. Also, some image effects for frame data are needed. We will combine these elements to develop our system.

Generally, people have large amounts of movie contents (e.g., TV programs and home videos) on their computers, so it is important for users to know how to effectively deal with and understand these contents. Our movie deformation was also received good reactions. In a summarized movie, the contents in the non-feature areas are played more quickly, enabling users to view the entire video and understand the importance of the contents based on the playback speed. Moreover, some visitors commented that this system was useful for movie contents that contain a user's favorite actor, singer, animation character, and sports player. Scenes with favorite characters would be played at normal speed and the other scenes would play more quickly.

We believe our comic approach would be effective for network communication. In conventional web services such as YouTube [7] and Niconiko [8], movie contents become communication trigger. Through our comic system, an input movie becomes interesting and comical. Also, the playtime of the movie is reduced by our deformation techniques. Thus, users can watch the movie interestingly and quickly. Also, by using simple manipulations, even beginners of movie creation could upload an original movie contents on the web sites and share their experiences and ideas with other users.

### CONCLUSION

We described the DigestManga system, which enables for a smooth relationship between comics and movies. We showed some examples of the comic visualization from an input movie and movie editing through the visualization. Also, we have described our movie deformation technique for deforming the features of movie contents.

### REFERENCES

- [1] S. McCloud, "Understanding Comics", Kitchen Sink Press, Northampton, MA, 1993.
- [2] J. Calic, D. Gibson, and N. Campbell, Efficient layout of comic-like video summaries". *IEEE Transactions on Circuits and Systems for Video Technology*, 17(7). ISSN 1051-8215, pp. 931–936, July 2007.
- [3] S. Uchihashi, J. Foote, A. Girgensohn, and J. Boreczky. "Video manga: Generating semantically meaningful video summaries". *In Proceedings of ACM MultiMedia '99*, pp. 383–392, 1999.
- [4] D. Kurlander, T. Skelly, and D. Salesin. "Comic chat". *In Proceedings of ACM SIGGRAPH '96*, pp. 225–236, 1996.
- [5] Y. Sumi, R. Sakamoto, K. Nakao, and K. Mase. "ComicDiary: Representing individual experiences in a comic style". *In Proceedings of UBIComp 2002*, pp. 16–32, 2002.
- [6] G. W. Furnas, Generalized fisheye views. *In Proceedings of the ACM Tran. on Computer-Human Interaction* 1, 2, pp. 126–160, 1994.
- [7] YouTube. <http://www.youtube.com>
- [8] Niconiko movie. <http://www.nicovideo.jp/>
- [9] S. H. Ling, B. B. Bederson, and D. W. Jacobs, Automatic Thumbnail Cropping and its Effectiveness, *In Proceedings of ACM UIST 2003*, pp. 95–104. 2003.
- [10] X. Fan, X. Xie, W. Ma, H. Zhang, and H. Zhou, Visual Attention Based Image Browsing on Mobile Devices, *In Proceeding of ACM Multimedia 2003*, pp. 148–155, 2003.
- [11] Iitokomi. <http://prius.hitachi.co.jp/go/prius/pc/2005sep/iitokomi/index.html>