Automatic E-Comic Content Adaptation

Kohei Arai arai@is.saga-u.ac.jp
Information Science Department
Saga University
Saga, 840-0027, Japan

Herman Tolle emang@brawijaya.ac.id
Software Engineering Department
Brawijaya University
Malang, 65145, Indonesia

Abstract

Reading digital comic on mobile phone is demanding now. Instead of create a new mobile comic contents, adaptation of the existing digital comic web portal is valuable. In this paper, we proposed an automatic e-comic mobile content adaptation method for automatically create mobile comic content from existing digital comic website portal. Automatic e-comic content adaptation is based on our comic frame extraction method combine with additional process to extract comic balloon and text from digital comic page. The proposed method work as a content adaptation intermediary proxy server application, while generating a Comic XML file as an input source for mobile phone to render a specific mobile comic contents. Our proposed method is an effective and efficient method for real time implementation of reading e-comic comparing to other methods. Experimental results show that our proposed method has 100% accuracy of flat comic frame extraction, 91.48% accuracy of non-flat comic frame extraction, and about 90% processing time faster than previous method.

Keywords: E-comic, Content Adaptation, Comic Frame Extraction, Text Extraction, Mobile Application

1. INTRODUCTION

Reading comic is one of popular thing in the world, especially in Japan. Everyday hundreds of printed comic book is produced and most of printed comic book then digitized into web contents for reading comic through the internet. As the usage of mobile device such mobile phone, PDA and laptops growth, reading comic through mobile device is also demanding. The recent trend is that comic content are largely demanded and became one of the most popular and profitable mobile contents. The challenge in providing mobile comic contents for small screen devices is how to separate comic frames and display it in the right order to read. However, the existing mobile comic content is mainly produced manually or automatically from offline comic book. Instead of create a new mobile content from digitized comic book in offline way, we propose a new method for automatically adapting digitized comic page from existing website into mobile comic content.

Several research projects [1-5], proposed systems that automatically convert web-based documents that were designed for desktop into appropriate format for viewed in mobile devices.

In [6], propose the concept of automatic mobile content conversion using semantic image analysis, but this method still using offline comic book as a comic page sources. In [6], authors propose *automatic content conversion* (ACC) ontology that using *X-Y recursive cut* algorithm for extracting comic frame. Like other method on comic frame extraction [7-10], those methods cannot detect frames when the comic balloon or picture is drawn over the frames. Then Tanaka proposes layout analysis of comic page using density gradient method [11], which applied to comic page with balloons or pictures drawn over the frames. However, in [11] method has some limitation in processing of comic image and not sufficient for real time application since computation of the process. Also success rate of frame extraction and processing time should be improved.

In this paper, an approach for automatically adapting existing online digital comic content – or electronic comic (e-*comic)* - into mobile comic contents based on automatic comic frame extraction (ACFE) method [13] is presented. We propose a new method for automatically extracting comic frame and frame contents such us comic balloon and text inside balloon from e-comic page. Comic frame contents such us balloon and text inside balloon is extracted for further purpose, for example for language translation, multimedia indexing or data mining. Our propose method is an efficient and effective comic content adaptation method that sufficient for real time online implementation. The experimental results of our method had shown the better results on accuracy and processing time comparing with other methods.

The reminder of this paper is organized as follows. In section 2, a detail description of the proposed method is given. Section 3 and 4 describe the detail process on frame content extraction and e-comic reader application. Experimental results with comparing to conventional method are presented in Section 5. Finally, conclusions are drawn in Section 6.

2. E-COMIC CONTENT ADAPTATION SYSTEM

Figure 1 shows the illustration of automatic e-comic content adaptation system. There are 3 parts involved in the concept of content adaptation systems [3], part A is a content provider, part B is an intermediary proxy server application, and part C is a mobile terminal. The concept of using content adaptation intermediary proxy server is related with current web technology and device independent paradigm [3]. Intermediary proxy server application will automatically adapt the comic page from existing e-comic website into mobile content specific to display on user mobile devices.

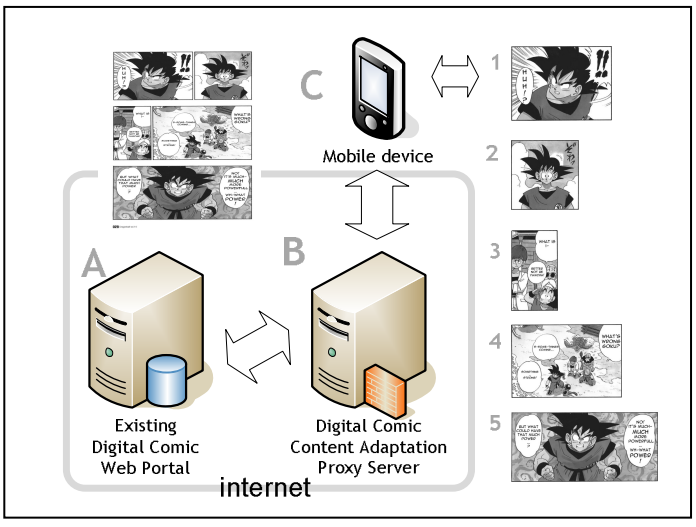


FIGURE 1: Illustration of E-Comic Content Adaptation System

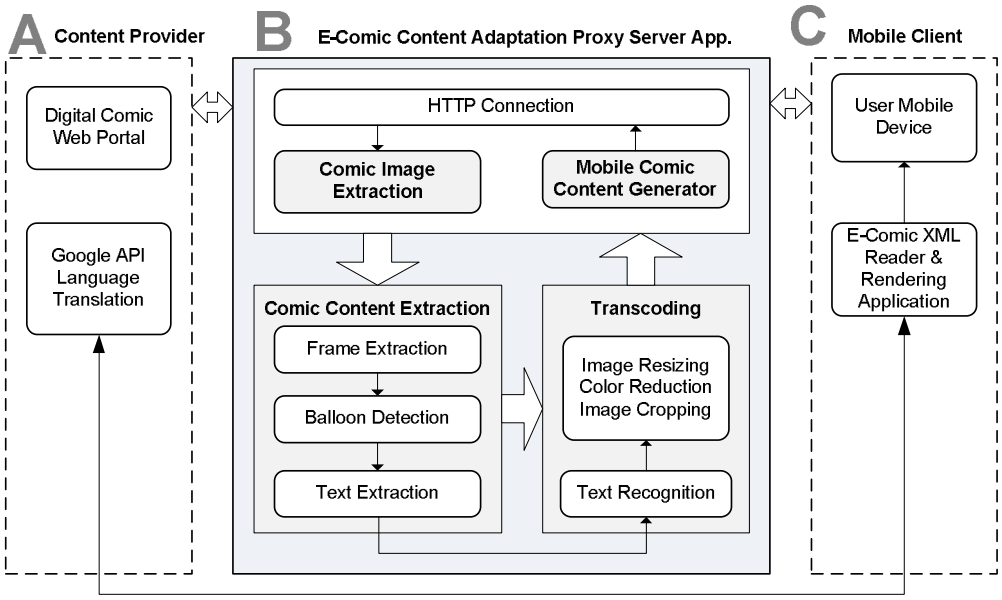


FIGURE 2: Architecture of E-Comic Content Adaptation System

2.1 E-Comic Content Adaptation Intermediary Proxy Server

Figure 2 shows the architecture of the automatic e-comic content adaptation systems. The process begins when a user (part C) uses a mobile device submit a request to the system—that is, to the content provider via an intermediary proxy server. After that, system connected to content provider for getting the comic page and then precedes the content adaptation to generate mobile specific content before deliver it to user. Architecture of e-comic content adaptation intermediary proxy server system consist 4 main parts as follows:

- *Comic Image Extraction*, comic page is grabbed from existing e-comic websites through HTTP connection and HTML parsing process. Database is needed to store information about comic portal URL and data about comic pages.

- *Comic Content Extraction*, useful information is extracted from a single comic page. The process of detecting and extracting the information about frame position, balloon position and text position based on e-comic content extraction method. Firstly, comic frames are extracted from comic page, then comic balloon is extracted from each frame, and the last is extracting text from each balloon.

- *Comic Content Trans-coding*, transform the source comic page into mobile specific content. There are 2 modes in our *transcoding* system: image *transcoding* and information *transcoding*, describe further on next sub chapter. In this part, also text image from previous process will recognizing as text using text recognition process. Extracted text from comic page is useful for language translation using Google translation services, data mining or multimedia indexing.

- *Mobile Comic Content Generator*, based on *transcoding* mode chooses by user, mobile content generator part will automatically create the output as mobile content to user mobile device. Output is Comic XML files for data and combine with XHTML mobile profile (MP) for presentation. After a complete process, system will store the data about comic page and adapted results to database for future usage. When another user requests the same comic page, system will only responds with stored data from database without any processing to reduce server load.

2.2 Comic Content Adaptation Strategies

Although data rate and file size is not a significant issue in recent internet wireless technology, we design our content adaptation system to support user with low speed internet connection likewise to user with high speed of internet connection. We design our system for generate adaptation results in 2 modes: *image transcoding* and *information transcoding*. Those two different adaptation modes are processed within comic content *transcoding* and mobile content generator parts.

- *Image transcoding mode*, system will reproduce comic frame content as new images with special treatment to fulfill user device requirement, for example: image resizing, color depth reduction or image cropping. Reproduction comic page is stored in proxy server and replace the original comic page. Image *transcoding* mode is useful for the user with limited internet connection. In this mode, output of the systems is Comic XML files for data combines with XHTML MP for presentation, and also generates new comic frame images.

- *Information transcoding mode*, system will produce only XML text files that store information about extracted frame content. Information transcoding mode is designed for user with high speed internet connection through wireless connection, because user device will display comic page in frame by frame using original comic page image. In this mode, an output of the systems is Comic XML files for data combines with XHTML MP for presentation. This Comic XML only contains information about comic frame content location within comic page.

2.3 E-Comic XML

Information about comic content is generated automatically and store in XML file for usage on mobile phone to render a comic content. Our E-Comic XML is improved from ComicsML version 0.2 by Jason McIntosh [12]. New Comic XML included the layout information of comic frame, balloon and text, which is not exist before. The layout information getting from comic frame content extraction process is useful for frame by frame displaying on user mobile devices. In *information transcoding* mode, layout information of comic frame content is stored as information of rectangle start point (x_1, y_1) and end point (x_2, y_2) of frame's blob, balloon's blob or text's blob. In *image transcoding* mode, layout information is no need but URL location of new images of frames, balloons or texts. Figure 3 show the data structure define in document type definition (DTD) of E-Comic XML.

```
<?xml version="1.0"?>
<!ELEMENT comic(title, url,
   readingorder?, language?, person+,
   icon?, description?, panels*)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT url(#PCDATA)>
<!ELEMENT creator(#PCDATA)>
<!ELEMENT readingorder(#PCDATA)>
<!ELEMENT language (#PCDATA)>
<!ELEMENT panels (number, panel+)>

<!- Information about frame -->
<!ELEMENT number (#PCDATA)>
<!ELEMENT text (#PCDATA)>
```

```
<!ELEMENT panel (order, panelurl,
   panelpos*, balloons*)>
<!ELEMENT order (#PCDATA)>
<!ELEMENT panelurl (#PCDATA)>
<!ELEMENT panelpos (posx1, posy1,
   posx2, posy2)>

<!- Information about balloon -->
<!ELEMENT balloons (balloon*)>
<!ELEMENT balloon(text?, textpost*)>
<!ELEMENT text (#PCDATA)>
<!ELEMENT textpos (posx1, posy1,
   posx2, posy2)>
```

FIGURE 3: Document Type Definition (DTD) of E-Comic XML,

3. E-COMIC CONTENT EXTRACTION

E-comic frame content extraction is based on our previous research work on automatic comic scene frame extraction [13]. For each comic page, we extract frames and then checking if any overlapped frames situated on the extracted frames. If overlapped frames detected, then system precede the overlapped frame division process. After all frame extracted, then balloons within

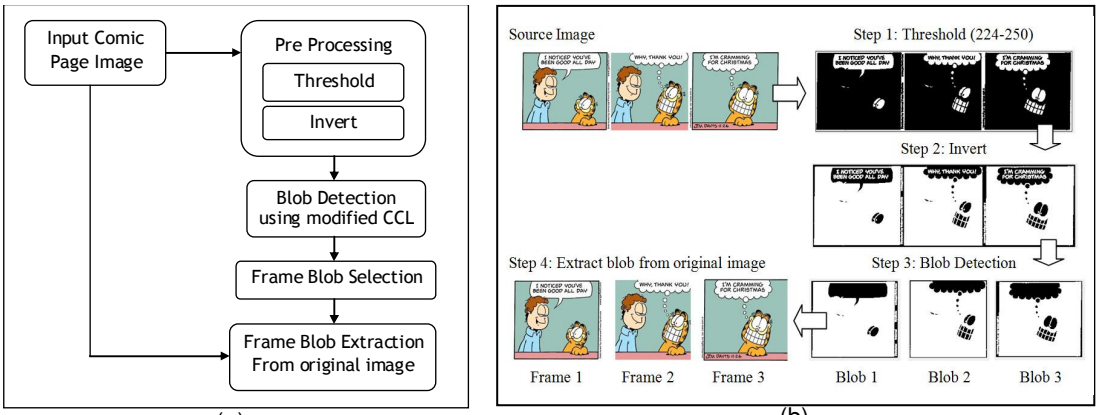
frame and texts within balloons are processed. All process is done base on modified of connected component labeling (CCL) algorithm [14] as our comic blob extraction function.

3.1 Comic Frame Extraction

Common comic frames are separated by white pixel line or white region, so the rest of white pixel region must be the frames. While the conventional method [7-11] tries to track the white line, our method finds the rest area of white line. We investigated many traditional and conventional comics those in case of there is no balloon or comic art is overlapped on frames - it is called '*flat comic*' hereafter-, each frame can be detected as a single blob object. In our propose method, we define all connected white pixels as a single blob object, and then each comic frames can be identified as an individual blob object.

We modify connected component labeling algorithm [14] for specific function on comic frame blob extraction. Figure 4.a show the flow diagram of the process of modified CCL for comic frame blob extraction function and Figure 4.b shows the results in step-by-step basis. Firstly, *binarization* is applied to converting color comic images into black and white images. *Binarization* with an appropriate threshold number will produce each frame as separate blobs. The heuristic value of threshold is 250 (for the images with quantization bits of 8 bits) that chosen empirically based on experiments. After that, color inversion is done to switch color between blobs and background, because our blob extraction method assume black pixel as background color. Then *blob detection* process generates blob object from each connected pixels. Last process is *frame blob selection* to select only blob with minimal size that determine as comic frame. The minimal size of selected frame blob is one sixth of the image size ([*Image.Width*/6] x [*Image.Height*/8]).

The proposed methodology has 100% of success rate for extract comic frames from complete *flat comic* page like comic "*Nonono Volume 55*" and other *flat comic* pages that we use in our experiments. The proposed method also can easily detect frames in comic pages that contain only one frame, which is problem in Tanaka's [11] method. The modified CCL for comic frame blob extraction method is, however, not perfect because comic image includes not only '*flat*' frame but also more complicated frame images those are overlapped with other comic balloons or comic arts. Then we improved our comic frame extraction method with overlapped frame checking and extraction using division line detection method.



(a) (b)

FIGURE 4: Flow diagram of comic frame extraction using comic blob extraction method, (b). Step-by-step process and result on frame extraction

3.2 Overlapped Frame Extraction using Division Line Detection

Using only blob extraction method, overlapped frames are not detected and will recognize as single frame. So, each frame should pass the overlapped frame checking process to detect the occurrence of division line between frames. If the division lines detected, then we will add new white line overlaid to create separate line between overlapped frames. Then overlapped frame can be extracted using our base function on blob extraction method.

The division line detection methods work by detecting the appearance of white area within a thick line that assumed as frame border line. For example, it is assumed that two frames are situated at the top and the bottom and overlapped by a comic balloon. The overlapped frame extraction process step is as follows:

1. Find the left and right frame border line, indicated by the X_{th} line with maximum number of black pixel, selected as candidate border line. X_1 in the left side and X_2 in the right side.
2. Find white area within along of the candidate border line (X_1 and X_2)
3. Decide one point in the white area of line X_1 as Y_1 and in line X_2 as Y_2. Thus we have $P_1(X_1, Y_1)$ and $P_2(X_2, Y_2.)$
4. Add a white pixel line between P_1 to P_2 as frame separator line.
5. Implement blob extraction method to separate two frames.

First, we try to detect border line by investigate on the edge area of comic page. Assume that edge area is N far from the edge, where N is empirically equal than one fifth of the page width. Estimated frame border line is determined from the line with maximum number of black pixel occurrence. After the candidate borderline, X_1 and X_2 are nominated, white pixel region within the lines is investigated. If X_1 and X_2 is real frame border of the images, it is possible to detect the division line between that indicated by the occurrence of white pixel areas within X_1 or X_2 lines. Thus one point in left side $P_1(X_1, Y_1)$ and the other one point in right side $P_2(X_2, Y_2.)$ determined. The line that connects P_1 and P_2 is estimated as separation line. Figure 5 shows the illustration of division line detection process and addition of separator line.

After two points detected, addition of a new white line between P_1 to P_2 will create separate top frame and bottom frame as two blobs. Then, using our blob extractions functions will successfully extracting two connected frames. For two frames connected in horizontal direction, do the same process while change the direction of border searching to top and bottom of the image. This method can also work well for comic art with straight division line in specific angle.

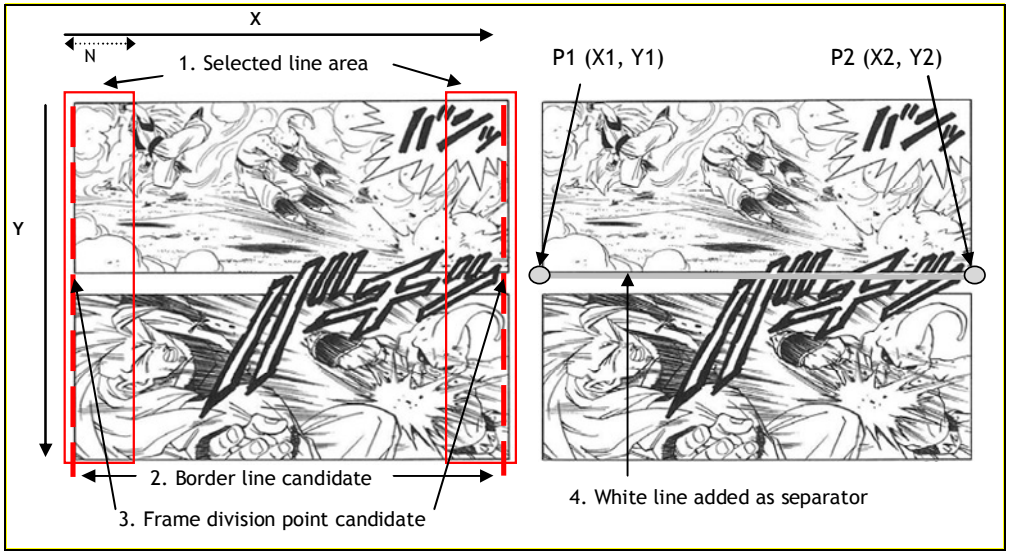


FIGURE 5: Division Line detection and line adding process.
(Source Images: Dragon Ball Volume 42 p.113)

3.3 Comic Balloon Detection

Comic balloon detection method is needed for correct the overlapped frame separation and for the purpose of text extraction process. However, while the additional of a white line in between of two intersections frames can separate frames properly; it sometime appears that intersection of content like balloon text is cut off. Therefore, it cannot be read properly. In order to overcome this situation, comic balloon detection method is proposed to detect comic balloon text areas that are

situated in between two comic frames. If a comic balloon detected in between of two frames, then the area of this balloon will add to one of the intersection frames where the balloon area is situated more than 50%.

The method for balloon detection is similar with frame blob extraction method but without *inversion* process. In typical comic images, balloon text usually has a white background. So, using base blob extraction method without inversion can detect comic balloon as a white pixel area. Balloon blob selection is base on 3 rules for classification as follows:

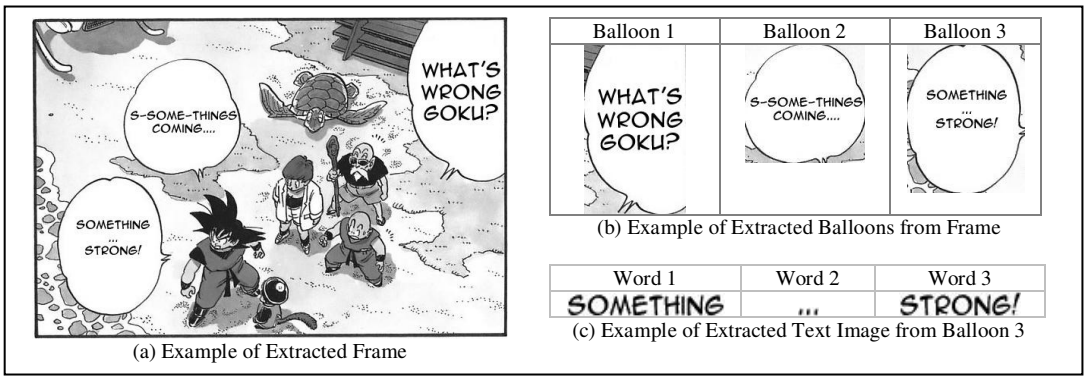
1. Minimal size of the blob is about [*Image.Width*]/10 and [*Image.Height*]/8 of frame image size.
2. Minimal number of white pixels in blob is 45% of blob area.
3. At least one text image is detected.

3.4 Text Detection and Extraction

Text extraction method is proposed for extracting text content from a comic balloon. The method for extract text from a balloon is also base on same method with frame extraction and balloon detection method. First, we implement modified CCL with morphology filter on pre-processing to make near word image collide as a single blob. In pre-processing, erosion and opening filter is applied with left and right side priority rather than top and bottom side. Balloon text blob selection base on some rules for classification as follows:

1. Minimal size of the blob is 40 pixel width and height
2. Ignored all blobs that related with border of balloon, approximately 5 pixels far from the balloon edge.

Figure 6 show the results sample on extraction of comic frame (a), comic balloon (b) and comic text inside balloon (c) from "*Dragon Ball Chapter 192*" comic. Frame and balloons are extracted in rectangle area while text is also extracted in rectangle area in the size of each word.



(a) Example of Extracted Frame

(b) Example of Extracted Balloons from Frame

(c) Example of Extracted Text Image from Balloon 3

FIGURE 6: Result samples on (a) Frame Extraction, (b) Balloon Extraction, (c) Text Extraction (Source Images: Dragon Ball Chapter 192 p.10)

4. ONLINE E-COMIC READER

Online e-comic reader is a special application for mobile devices that separated from comic content adaptation systems. People can build their own application for reading comic on mobile phone as long as they can interpret e-comic xml file into mobile comic application. That is major point in content adaptation method when intermediary proxy server content adaptation is applied. Figure 7 shows our simple e-comic reader application on PDA to display comic page in frame by frame basis. Comic image is relatively convenient to read in each frame image size rather that whole comic page size. The illustration of an online e-comic reader application with special features for language translation is shown in Figure 8. We can combine comic reader application with Google language translation features to generate language translation of comic from XML files.

FIGURE 7: E-Comic Reader Application on PDA
(Source Images: Zettai Kareshi Manga, Vol. 1, p.171)

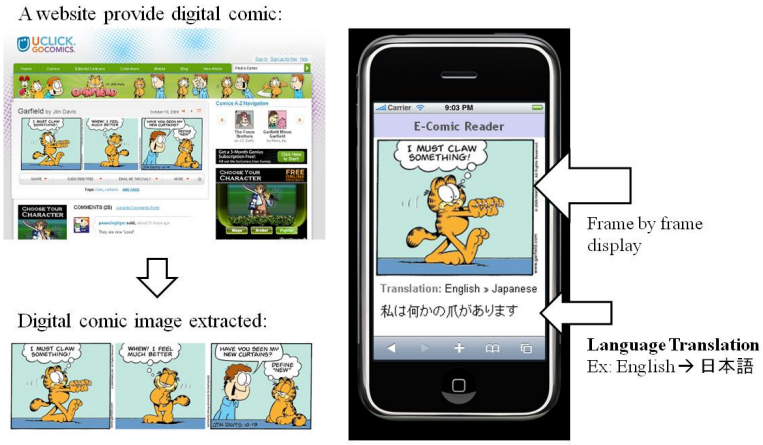


FIGURE 8: Illustration of Online E-Comic Reader Application with Language Translation Features
(Source Images: Garfield Comic from Gocomics)

5. EXPERIMENTS

The proposed methodology for automatic e-comic content adaptation has been evaluated using various comic image pages in offline and online situation. We implement the proposed method in real time online and offline situation using Microsoft.Net environment with C# as native language for proxy server application and frame content extraction process. We use desktop computer with Pentium Dual Core processor and 1 Mbyte of RAM. Experiment is conducting through 634 comic pages to evaluate the success rate (accuracy) of frame extraction and processing time. Common comic image size that we use in our experiment is 800x1200 pixels. The results of the experiment then reported and compared with other methods.

5.1 Comic Frame Extraction Experimental Results

Experimental result of frame extraction method has shown in Table 1. The results were classified into 3 groups such as "*correctly extraction*", "*missed detection*" and "*false detection*". The term correctly extraction means the success frame extraction without error. The terms "missed detection" means that system cannot extract overlapped frames, and the terms "false detection" means that some non frame detected as a frame. From the experimental results, 91.48% average of success rate of comic frame extraction is achieved.

TABLE 1: Frame extraction experimental results for 634 pages from 5 comic sources.

Digital Comic Sources	Total Pages	Correctly Extraction	Missed Detection	False Detection	Success Rate (%)
Dragon Ball Vol 40	175	161	12	2	92.00
Dragon Ball Vol 42	237	218	10	9	91.98
One Piece Vol 1	191	171	20	0	89.53
Nonono Vol 55	18	18	0	0	100.00
Dragon Ball Ch 196	13	12	1	0	92.31
Total	**634**	**580**	**46**	**8**	**91.48**

An experimental result for comparison with Tanaka's [11] method has shown in Table 2. In our experiments we also include some particular pages in main volume were Tanaka exclude it. So, the total number of tested images (one image is one comic page) was different. The results were classified into 5 groups such as "*Succeeded*", "*Not succeeded*", "*Not tested*", "*Total pages tested*" and "*Total pages*". The term "*Succeeded*" means the total pages of success on frame extraction. The term "*Not succeeded*" means the total page of failure for frame extraction. The term "*Not tested*" means number of pages that not include in testing process. The term "Total Pages" means the total number of comic page of the comic.

TABLE 2: Experimental comparison results with Tanaka's [11] method for the Dragon Ball Volume 42 comic image sources

Classification of Results	Comic Page	
	Tanaka's Method	Our Method
Succeeded	195 / 82%	218 / 92%
Not succeeded	22	19 / 8%
Not tested	20	0
Total Page Tested	217	237
Total Page	237	237

Our method is better than Tanaka's method as shown on the experimental result in Table 2. By using the same comic source images, our method is 10% better than Tanaka's methods. Our methods also need less computation process because the efficiently of division line detection algorithm and blob extraction method. Once blob extraction function created, then it reused in frame extraction, balloon extraction and balloon text extraction.

5.2 Comic Balloon and Text Extraction Experimental Results

Performance evaluation of proposed methods on balloon detection and text extraction is evaluate for 13 comic pages from Dragon Ball Chapter 196 comic pages. Experimental result of frame extraction method has shown in Table 3. The results were classified into 3 groups such as "*correctly extraction*", "*missed detection*" and "*false detection*". The term *correctly extraction* means the success of balloon detection or text extraction without error. The terms *missed detection* means that system cannot detect balloon or text. The terms *false detection* means that some non balloon or non text detected. From the experimental results, 90.70% of success rate of comic balloon detection method and 93.63% of success rate of comic text extraction methods is achieved.

TABLE 3: Comic Balloon and Text Extraction experimental results

Comic Content	Total	Correctly Extraction	Missed Detection	False Detection	Success Rate (%)
Balloon Detection	121	86	8	2	90.70
Text Extraction	314	294	20	8	93.63

5.3 Evaluation of Processing Time

Time consuming in processing is main issue in real time online application. We evaluate processing time of our method in offline and online simulation. In offline simulation, processing time of each process on comic frame extraction is evaluated. In online simulation, total time for processing all process is evaluated, including: comic image parsing processing, comic content frame extraction and output generating. However, in online situation experiment, we do not counting the time consuming for access or downloading the comic image file, and also without text recognition and image reproducing process. The result of processing time evaluation is shown in Table 4, with comparison with other method. Processing time experimental results shows that our proposed method is faster that other method. Comparing with [6], processing time of our method is about 90% faster than [6]. Online situation need more processing time consuming rather than offline situation because of another processing within the systems, but still acceptable as online application.

TABLE 4: Processing time experimental results and comparison

Comic Page	Processing Time (in seconds)		
	[6]	[10]	Our Method
1 comic page offline	3	25	0.250
1 comic page online	-	-	0.513
30 comic page offline	90	750	10
30 comic page online	-	-	16

6. CONCLUSION

We implemented a system for automatically adapt e-comic content for reading comic on mobile devices. We proposed frame content extraction method and intermediary content adaptation proxy server systems with new E-Comic XML. Comic frame content extraction method is based on blob extraction method using modified of connected component labeling algorithm. The proposed method on frame extraction does work in a real time basis so that it is possible to adapt relatively large scale of existing digital comic image to comparatively small screen size of mobile terminals by displaying extracted images onto the screen by frame-by-frame. It is still rather difficult to detect balloons, images, and characters those are situated in between frames. The proposed method allows detection of these and separates the different frames even if these balloons, images, and characters are exist.

The proposed method has produced better results in frame extraction method and executes faster than other methods. From the experimental results, our comic frame extraction method has 100% accuracy for flat comic and 91.48% accuracy for non-flat comic, while balloon detection method achieves 90.7% accuracy and text extraction method achieves 93.63% accuracy. Our comic frame extraction method has 10% improvement of [11] method and about 90% processing time improvement of [6] methods. Our comic frame extraction method is an efficient and effective methods comparing to conventional method, and applicable for real time online e-comic content adaptation application.

Our system is designed to be adaptable with old and new mobile technologies, because it can create mobile comic content based on user's profile. Our system provides 2 mode of content adaptation, image *transcoding* mode for old mobile devices with limited internet connection, and information *transcoding* for new mobile devices with high speed internet connection. Also, our system creates e-comic xml files that are being able to use by third party companies to develop their own application of e-comic reader. The future direction of this research work is to provide a robust algorithm for extraction e-comic content and automatically convert it into mobile specific content. The accuracy of comic frame extraction and text extraction should be improved and needs further exploration. By utilizing the results of our study and further exploration, the real

implementation of online reading of existing e-comic on mobile phone can immediately be realized.

7. REFERENCES

1. Chen, Y., Ma, W.Y., Zhang, H.J.: "*Detecting Web Page Structure for Adaptive Viewing on Small Form Factor Devices*". In Proceedings of the International WWW Conference Budapest, Hungary, 2003

2. Wai Yip Lum, Francis C.M. Lau, "*A Context-Aware Decision Engine for Content Adaptation*". IEEE Pervasive Computing, 1(3): 41-49, 2006

3. Laakko et al., "*Adapting Web Content to Mobile User Agents*", IEEE Internet Computing, 9(2):46-53, 2005

4. Dongsong Zhang, "*Web content adaptation for mobile handheld devices",* Communications of the ACM, 50(2):75-79, February 2007

5. Hsiao J.-L., Hung H.-P., Chen M.-S., "*Versatile Transcoding Proxy for Internet Content Adaptation*", IEEE Trans. on Multimedia, 10(4):646--658, June 2008.

6. Eunjung Han, et.al. "*Automatic Mobile Content Conversion Using Semantic Image Analysis*", Human-Computer Interaction HCI Intelligent Multimodal Interaction Environments, LNCS 4552, Springer, Berlin, 2007

7. Ono Toshihiko. "*Optimizing two-dimensional guillotine cut by generic algorithms*". In Proceedings of the Ninth AJOU-FIT-NUST Joint Seminar, pages 40-47, July 1999.

8. Yamada, M., Budiarto, R. and Endoo, M., "*Comic image decomposition for Reading comics on cellular phones*". IEICE transaction on information and systems, E-87-D(6):1370-1376, June 2004.

9. D. Ishii, K. Kawamura, H. Watanabe, "*A Study on a Fast Frame Decomposition of Comic Images*," National Convention of IPSJ, 1P-2, March 2007

10. Chung HC, Howard L., T. Komura, "*Automatic Panel Extraction of Color Comic Images*", Advances in Multimedia Information Processing – PCM 2007, LNCS 4810, Springer, Berlin, 2007

11. Tanaka, T., Shoji, K., Toyama, F. And Miyamichi, J.: "*Layout Analysis of Tree-Structured Scene Frames in Comic Images"*. In Proceedings of IJCAI 2007, pp. 2885-2890, June 2007.

12. Jason McIntosh. "ComicsML", an essay in http://www.jmac.org, published 2005. Accessed March 2008.

13. Kohei, A., Tolle, H. "*Method for Automatic E-Comic Scene Frame Extraction for Reading Comic on Mobile Devices*", In Proceedings of ITNG 2010 Conference, April 2010. Confirmation accepted

14. F. Chang, C-J. Chen, and C-J. Lu. "*A Linear-Time Component-Labeling Algorithm Using Contour Tracing Technique",* Computer Vision and Image Understanding, 93(2):pp. 206-220, 2004.

15. R. Gonzalez and R. Woods. *"Digital Image Processing"*, Addison-Wesley Chap.2., Publishing Company (1992)