Method for Real Time Text Extraction of Digital Manga Comic

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

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

Abstract

Manga is one of popular item in Japan and also in the rest of the world. Hundreds of *manga* printed everyday in Japan and some of printed *manga* book was digitized into web *manga*. People then make translation of Japanese language on *manga* into other language -in conventional way- to share the pleasure of reading *manga* through the internet. In this paper, we propose an automatic method for detect and extract Japanese character within a *manga* comic page for online language translation process. Japanese character text extraction method is based on our comic frame content extraction method using blob extraction function. Experimental results from 15 comic pages show that our proposed method has 100% accuracy of flat comic frame extraction and comic balloon detection, and 93.75% accuracy of Japanese character text extraction.

Keywords: E-comic, Manga, Image Analysis, Text Extraction, Text Recognition

1. INTRODUCTION

Manga is one of popular item in Japan and also in the rest of the world. Hundreds of *manga* book is printed everyday in Japan, and some of printed *manga* book is digitized into web content for reading comic through the internet. People then make translation of Japanese language in manga into other language to share enjoy of reading *manga* for non Japanese reader. However, people make translation of the text on printed comic book (they call it *scanlation*) in manually because there is no automatic method for translate comic text image into any other language. The challenge in extracting Japanese character in *manga* is how to detect comic balloon and extract text in vertical direction as Japanese classic writing direction is top down and right to left.

Several research projects [1-4] proposed method for text extraction from images but not specific for extraction from comic image. There are two base methods for text extraction, using region based method and texture based method. In [5], propose the concept of automatic mobile content conversion using semantic image analysis that include comic text extraction, but this paper did not explain the details for text extraction. Also, Yamada [6] proposed method for comic image decomposition for reading comic on mobile phone that including comic text extraction but not details on comic text extraction. The conventional method assuming extraction process in offline way and using scanned comic image. In the internet and mobility era, we need advance method for extraction text in online way and automatically make translation using online translation feature on internet like Google language translation.

In this paper, an approach for automatically extract Japanese character from *manga* comic is presented. We propose a new method for automatically extract text inside comic balloon from digital comic (e-comic) image in online way. 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. This research work is improvement of our previous research on automatic e-comic content adaptation [7] that designed for extraction comic content from existing comic image in comic web portal and adapting it to mobile content for reading on mobile phone. Our propose method is an efficient and effective text extraction 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 comic content extraction method is given, including balloon detection method. Section 3 describes the detail process on text extraction method. Experimental results with comparing to conventional method are presented in Section 4. Finally, conclusions are drawn in Section 5.

2. COMIC FRAME CONTENT EXTRACTION

Extraction of the text from digital comic image page assuming only text situated within a comic balloon which is located in a comic frame. That is why comic frame detection and comic balloon detection has to be done clearly before text extraction as shown on flow diagram of comic text extraction method in Figure 1. We use the same process for comic content extraction [8] on frame extraction but improvement on balloon detection and text extraction. In comic frame content extraction, we extract comic frames and then checking if any overlapped frames situated on extracted frames. If there any overlapped frames are detected, then processed with overlapped frame division. After every frame extracted, all balloon inside a frame and texts inside balloons are extracted. All process is done base on modified connected component labeling (CCL) algorithm [9] as comic blob extraction function.

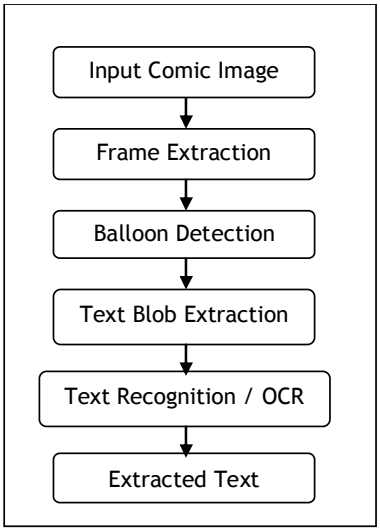

FIGURE 1: Flow diagram of digital comic text extraction method,

2.1 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. 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 [9] 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 to black and white images. Binarization with an appropriate threshold number produces 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 and produced good results. Then color inversion is done to switch color between blobs and background, because our blob extraction method assume black pixel as background color. After that, blob detection process will produce connected pixels as separate blob object. 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 [*Image.Width*/6] x [I*mage.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 Dragon Ball Chapter 195 that we use in our experiment. 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 [7], but this paper currently focus on the improvement of text extraction method.

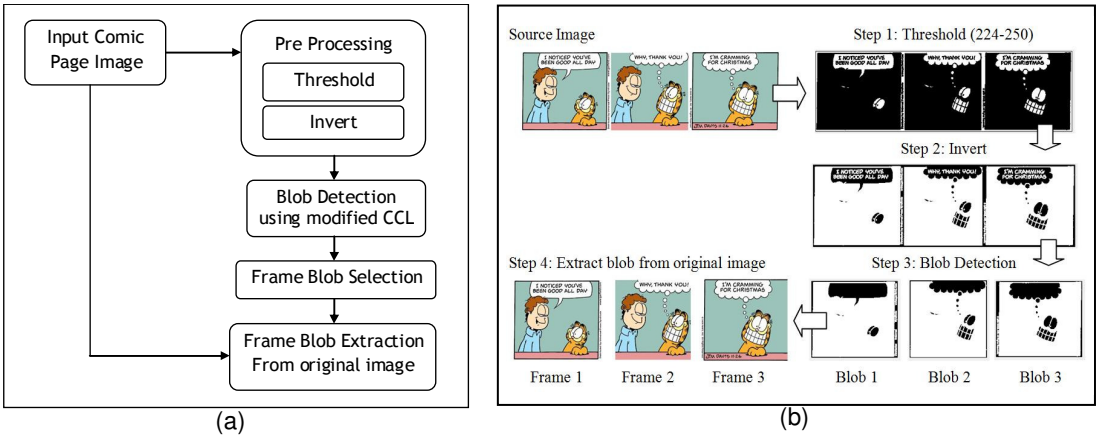

FIGURE 2: Flow diagram of comic frame extraction using comic blob extraction method, (b). Step-by-step process and result on frame extraction

2.2 Comic Balloon Detection

Comic balloon detection is one of the important processes in comic text extraction, because we only extract text inside a comic balloon. The accuracy of balloon detection is correlated with successfully of text extraction. We use same basic blob extraction function for frame extraction in balloon detection process. In typical comic images, balloon text usually has a white background. So, using base blob extraction method without inversion can detect comic balloon which has white pixel area. Flow diagram of balloon detection is shown in Figure 3.

Blob detection process will detect many blobs, then balloon blob selection is needed to classify and select only balloon candidate. Balloon blob selection is base on 4 rules for classification as follows:

1. *Blob Size (width & height)*: Minimal size of the blob is about [*Image.Width*]/10 and [*Image.Height*]/8 of frame image size.
2. *White pixel occurrence*: Minimal number of white pixels in blob is 45% of blob area.

3. *Straight Line*: At least two straight vertical white lines with 70% long of blob height size is detected. One line in the half left and one in the half right.
4. *Width to Length Ratio*: Blob width size is less than 1.5 of blob height size.

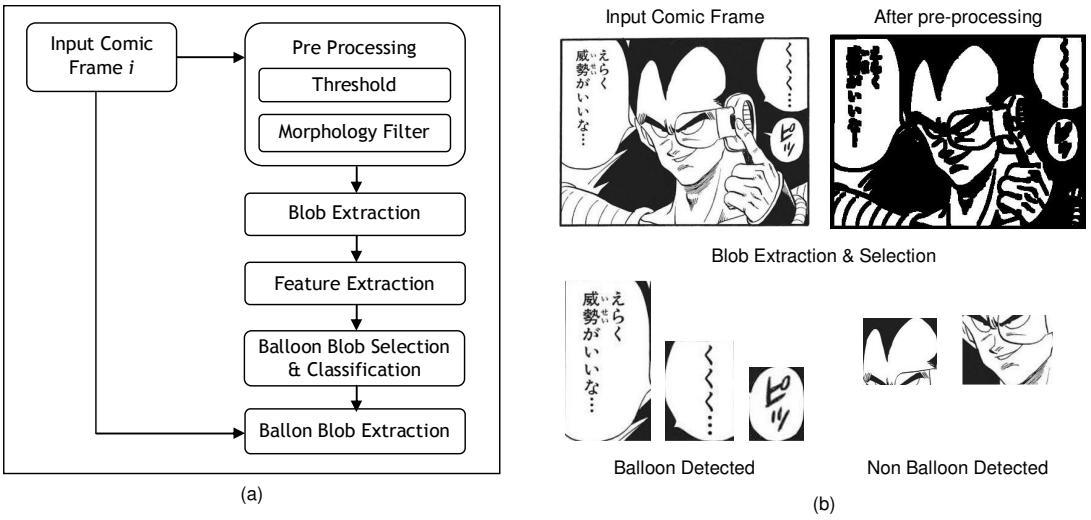

FIGURE 3: Flow diagram of comic balloon detection using comic blob extraction method, (b). Step-by-step process and result on comic balloon detection

3. BLOB EXTRACTION BASED METHODOLOGY FOR TEXT EXTRACTION

The objective of text extraction method is to extract Japanese text from comic balloon as single blob object for same vertical sentence or word. This vertical direction word will useful for character recognizing using optical character recognition (OCR) method which is out of this scope at this time. The proposed methodology is blob extraction based or region based method, and operates on digital comic image which has frame extraction and balloon detection processed previously. The text extraction method comprises of 5 phases; pre processing with binarization and morphology filter, blob extraction, blob feature extraction, text blob selection and classification, and the last process is text blob extraction. The block schematic of the text extraction methodology is given in Figure 4. The detailed description of each phase is presented in the following subsections.

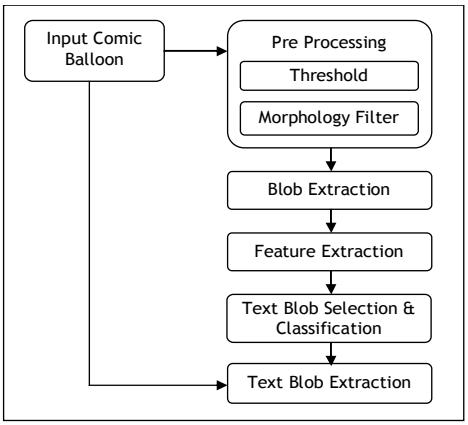

FIGURE 4: Text Extraction Algorithm

3.1 Pre-Processing

Before extracted text blob from comic balloon, pre-processing filtering has to be done for preset image into binary color. First, adaptive threshold method is applied to comic page image. Threshold value *T* is obtained from average pixel value of comic image *μ*. We set threshold value is 30% above from the average empirically.

$$\mu = \frac{1}{N}\sum_{i=0}^{N-1} I_i \quad (1)$$

$$T = \mu / 0.30 \quad (2)$$

Then we implement a morphology filter in *pre-processing* to merge neighboring character blob image as a single blob. In pre-processing, erosion filter and opening filter is applied with top and bottom side priority more than left and right side because Japanese character reading order in *Manga* is from top to bottom. The block diagram and results sample of pre processing is given in Figure 4.

3.2 Blob Extraction

After pre-processing, we detect all connected pixels as a separate blob object. Blob extraction is done using same blob extraction method with 2 previous processes to extract text blobs candidate from a comic balloon. This process produces text blobs and also non text blobs. To classify a blob as a text blob or non text blob, we extract some features from text blob candidate for classification. In text blob extraction using blob extraction function, we select only blob with minimal size that selected as candidate of text blob. The minimal size of the text candidate blob width is [*Image.Width*]/20 and the text blob height is [*Image.Height*]/40. Parameter of *[Image]* corresponds to balloon blob image size as the input of process.

3.3 Text Blob Feature Extraction and Classification

Text blob classification is needed to classify text blob from extracted blobs into vertical (column) text region. Classification is based on the features of blob size and blob position. Classification process classify text blob with the same column into same cluster as a representative of Japanese classic sentence or word in column. To classify and combine text blob in vertical direction, we extract some features from detected text blob. Features to extract are *average text blob width* and *text blob x center point* (*TB.XCenter*). Average text blob width is needed to determine distance between two columns of vertical text. Average text blob width is chosen from average width of three blobs in the *mean* of blob after sorting all blob based on blob's size. *Text blob x center point* is described in Equation 3. Text blob size of each blob is actual parameter of blob object which is containing *top* and *left* position parameter, and *width* and *height* parameter.

$$TB[i].XCenter = TB[i].Left + (TB[i].Width / 2) \quad (3)$$

Where;
TB[i] corresponds to i^{th} blob of detected text blob.
TB[i].Left corresponds to left point position parameter of detected text blob
TB[i].Width corresponds to width parameter of detected text blob

For selecting and classifying text blob into vertical text region, we use 2 rules for classification as follows:

1. Blob position is not related with balloon blob border. Assuming minimal distance is d_{min}.
2. Classify blob with same vertical center point in same cluster. Assuming difference distance between centers is BD_{min}.
   If (TB[i].XCenter +- BD_{min}) ∈ Column[j] Then TB[i] member of Column[j]

Minimal distance from border d_{min} is approximately 5% pixels from balloon edge and minimal distance BD_{min} is approximately less than half of average text blob width.

3.4 Text Blob Extraction

Last step on text extraction method is text blob extraction from original balloon image based on text blob cluster from previous process. In classification process, each blob in same cluster is combining as a new blob as representative of text in vertical. Then using the position of new blobs, text candidate is extracted from original balloon text image from input process. Figure 5 show step by step process and results of *manga* text extraction algorithm. The final result is set of text blob that represent Japanese sentence from comic text balloon in vertical direction. Those image blobs than store in database or directly proceed to OCR application for text recognition.

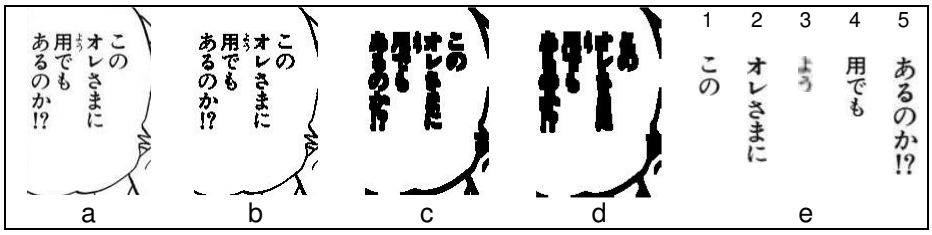

FIGURE 5: Sample of Text Extraction Process
(a) Original balloon text image; (b) Threshold; (c) Morphology-Erosion filter;
(d) Morphology-Opening filter; (e) Extracted blob from original image

4. EXPERIMENTS

The proposed methodology for extracted Japanese character from digital comic image has been evaluated using various comic image pages. We implement the proposed method using Microsoft.Net environment with C# as native language and AForge image processing framework on desktop computer with Pentium Dual Core processor and 1 mega byte of RAM. Experiment is conducting through 15 comic pages to evaluate the success rate (accuracy) of text extraction. 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.

4.1 Experimental Results

Performance evaluation of proposed methods on frame extraction, balloon detection and text extraction is evaluated for 15 comic pages from Dragon Ball Chapter 195 Japanese version. This comic contains 15 pages with 60 frames, 70 comic balloons and 160 columns of Japanese sentences. Experimental result of balloon detection and text extraction method has shown in Table 1. The results were classified into 3 groups such as "*correctly extraction*" (*T*), "*missed detection*" (*M*) and "*false detection*" (*F*). In the case of frame extraction, the term *correctly extraction* means the number of true detection of frames; *missed detection* means that there are frame that not detected by system; and the terms *false detection* means that some non frames are detected as frame. We also use the same interpretation on *T*, *M* and *F* for balloon detection and text extraction process. In Table 1, there are 2 methods, base method and modified method. Base method is our method which is explained in previous chapter, while *modified method* is base method with modification. Detail of modified method explained in *Discussion* on next sub chapter. From the experimental results, 100% of success rate of frame extraction method, 100% of success rate of comic balloon detection method and 93.75% of success rate of manga text extraction methods is achieved.

Table 1. Comic Balloon and Text Extraction experimental results

Comic Content	Total	T	M	F	%
Frame Extraction	60	60	0	0	100

Balloon Detection based method	70	70	0	53	100
Balloon Detection with modification	70	70	0	7	100
Text Extraction	160	150	10	6	93.75

Time consuming in processing is main issue in real time online application. We evaluate processing time of our method in offline simulation for each process on comic text extraction without text recognition process. The result of processing time evaluation is shown in Table 2, with comparison with other method. Processing time experimental results shows that our proposed method is faster that other method. Comparing with conventional method, processing time of our method is about 90% faster than conventional method. 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 2. Processing time experimental results and comparison

Comic Page	Processing Time (minutes)	
	Conventional Method	Our Method
15 comic page offline with 70 balloon	Approx. > 30	2.5

4.2 Discussion

Although our base method for balloon detection and text extraction has performed good results on *correctly extraction*, it still has a lot of false detection. In balloon detection method, there are a lot of non balloon detected as balloon candidates. It causes by the form of comic image that looks like balloon and pass our selection criteria. To reducing false detection, we make a simple modification for balloon detection criteria by adding one process. The new process is performing text detection on balloon candidate. If a candidate of text detected inside a balloon candidate, then classify it as a new balloon candidate. If there is no text blob candidate detected inside a balloon, then classify it as a non balloon. Implementing of this modified method, reduce about 90% of false detection. Figure 6 shows the sample of false detection of balloon candidate, which eliminated on text detection. Actually, the occurrence of false detection in text extraction process is not a serious problem because in the next process while we implement OCR for extract text from text blob image, the false detected of balloon or text produces nothing after OCR processing. In the other hand, the accuracy of text extraction method is 93.75% while 6.25% of texts are not detected or system fail to detect and extract. Most of the failure on text extraction is cause by non standard of position of the character within a balloon as shown in Figure 6.b. System also fail to extract 2 or more text column that close to another.

Implementation of our method is possible in offline or online real time basis because the simplicity of the program code and fast processing time. By using our method, vertical Japanese sentence in a comic balloon is easy to extract then recognized for language translation. We can create a mobile application for reading comic with language translation process by implementing of our method combine with OCR feature and *Google Language Translation* tool using mobile programming *framework* like *Android* or another framework.

(a) (b)

FIGURE 6: (a) Sample of non balloon detected as a balloon candidate (false detection); (b) sample of miss detection text

5. CONCLUSION & FUTURE WORK

We implemented a system for automatically extract Japanese character from digital manga comic image. Our proposed method can automatically detect comic frame and detect all balloon within a comic frame. After all balloon detected, then system will extract Japanese character text inside a balloon as comic dialog in Japanese language for further purpose especially for language translation. The proposed method on Japanese character text extraction is possible to implement in offline or online real time basis so it is possible to become an online application for language translation. From the experimental results, our comic frame extraction method has 100% accuracy for flat comic, balloon detection method achieves 100% accuracy and text extraction method achieves 93.75% accuracy.

Combining the proposed method with our e-comic content adaptation system, will create a robust system for automatically convert digital comic image from comic web portal into mobile content with language translation features. We will improve this current method with implementing character recognition function in the systems. Also for further work, we will implement an application for automatically convert Japanese comic into international comic for enjoy reading comic anywhere anytime using mobile phones.

6. REFERENCES

1. L. Fletcher and R. Kasturi, A robust algorithm for text string separation from mixed text/graphics images. IEEE Trans. Pattern Anal. Mach. Intell. 10 (1988), pp. 910–918

2. A.K. Jain and B. Yu, Automatic text location in images and video frames. Pattern Recognition 31 12 (1998), pp. 2055–2076

3. O. Iwaki, K. Kubota and H. Arakawa, A character/graphic segmentation method using neighborhood line density. IEICE Trans. Inform. Process. J68 4 (1985), pp. 821–828.

4. L.A. Fletcher and R. Kasturi, A robust algorithm for text string separation from mixed text/graphics images. IEEE Trans. Pattern Analysis Mach. Intell. 10 6 (1988), pp. 910–918

5. 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

6. 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.

7. Kohei Arai, Tolle Herman, "Method for Automatic E-Comic Scene Frame Extraction for Reading Comic on Mobile Devices," itng, pp.370-375, 2010 Seventh International Conference on Information Technology, 2010

8. Kohei, A., Tolle, H., “Automatic E-Comic Content Adaptation”, International Journal of Ubiquitous Computing IJUC Volume (1): Issue (1), May 2010.

9. 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.

10. R. Gonzalez and R. Woods. “Digital Image Processing”, Addison-Wesley Chap.2., Publishing Company (1992)