ABSTRACT

Comic has been around for a long time. It became famous with the revolution of printed media. Manga is a type of comic that was emerged with Japanese culture. It is very popular around the world because it has a broad range of genre and target reader. Digital comic are also becoming more popular along with the increasing technological advances in mobile devices. Reading comic collections using mobile phone gives benefit that there is no need to bring the physical comic books. Existing mobile comic applications can only display the full comic page. It brings problem because user has to zoom and pan the image to read the comic page properly as the text are too small for mobile phone's screen. Author develops a method for extracting panels from digital comic pages automatically using connected component algorithm. The extracted panel images then can be used as sources for a new kind of mobile comic reader.

Keywords: Comic, Manga, Panel Extraction, Connected Component

1 INTRODUCTION

Comic is a form of graphic narrative. It is a blend of pictures and words to describe a story [1]. Graphic narrative form has been around for centuries. Comic became popular in early 20th century after the advances in printed mass media. It was appeared in newspapers as comic strip. Then it was gathered in booklet form known as comic book.

One of the most popular type of comic today is manga. It came from Japan and has been a part of Japanese art [2-4]. Manga has broad range of target reader, from male to female, from teenager to adults, and also with many genres. It's market in Japan was worth 5.5 Billion in 2009. It also has been very popular throughout the world [3].

Aside from printed comic such as comic book, digital comic are becoming more popular. It is influenced by the increasing popularity of mobile phone, specially smartphone. Smartphone is a mobile phone that has advanced operating system, can do multitasking, and can be installed with customized applications [4]. There was nearly 500 Million smartphones sold worldwide in 2011. It is still increasing rapidly. According to Gartner, the annual sales of smartphone will reach 1.2 Billion in 2013. That is more than twice in increase just in 2 years. More than 50% smartphones sold in 2011 are using Android operating system.

There are several problems in viewing comic pages on smartphone's screen. The diagonal of the screen usually varied from 3" to 5". It is not enough to view a full comic page because the text will not be readable. User has to zoom in the page to make the image larger so that the text are large enough to be readable. Than user has to zoom out or pan through the image to see the next panel. Panel is a visible area that contains a moment of story in the comic page [5]. The need for zooming and panning the image is not quite comfortable for reading comic.

In this research, author develops a method to extract the panels in the comic page automatically so that it can be displayed on small screen while still being readable. User can display the comic by viewing each panel without the need of zooming and panning to read the text in the panel. Author develops the methods to solve the problem of how to detect the panels contained in digital comic page extract the detected panels.

There are some recent research that is similar to the goal of this research which is to automatically extract panels from comic pages. A proceeding by Kohei Arai published in Information Technology: New Generations (ITNG) 2010. It is titled "Method
for automatic e-comic scene frame extraction for reading comic on mobile devices. Another similar research is a proceeding by Anh Khoi Ngo published in Ninth International Workshop on Graphics Recognition 2011. It is titled "Comics page structure analysis based on automatic panel extraction". Unfortunately the author can only access the abstracts of those papers and the implementation source of the research are also nowhere to be found. Hence author continues the research using connected component labeling algorithm as it is designed for blob extraction and is suitable to the goal of this research.

2 ANALYSIS, DESIGN, AND IMPLEMENTATION

The comic books and strips contain panels. A panel is a visible area that contains a moment of the story. The comic's author tell a story by presenting carefully selected moments of varying length within panels. The panels are placed side by side and has certain reading sequence. Western comic can be read from upper left to lower right. Another kind of comic such as manga has different reading sequence. Figure 1 shows the reading sequence for manga.

2.1 Method for Extracting Comic Panels

Before the comic panels can be extracted from the comic page, it needs to be identified first. To identify the panels, in this research author uses connected component labeling algorithm. Connected component analysis consist of connected component labeling of the black pixels (foreground) followed by property measurement of the component regions and decision making. The connected component labeling operation changes the unit of analysis from pixel to region or segment. All black pixel that are connected to each other by a path of black pixels are given the same identifying label [6].

There are several steps used in the method to automatically extract panels from comic pages, figure 2 shows the steps used in the method. First the comic images are read into matrix in grayscale format. The matrix dimensions are defined by the image's height and width. The number of pixel from the image's height represents the matrix's total row. The number of pixel from the image's width represents the matrix's number of column. Each cell in the matrix are filled with number from 0 to 255. The numbers represent the gray level of the relevant pixel. Black is represented by value of 0. White is represented by value of 255. Other gray level is represented with value in between.

Next step is to convert the image to binary image using thresholding. The values in the grayscale matrix are converted to binary value of 1 bit. The values can only be 0 or 1. Black is represented by value of 0. White is represented by value of 1. The specified value for thresholding are used to determine the binary value. If a cell from the grayscale matrix are less than threshold value, it is given 0 for the binary value. Otherwise, it is given 1 for the binary value.
The third step is to remove small objects from the binary image. To avoid small objects like letter in the comic panel to be labeled as individual object, it is removed from the binary image. Each pixel in the binary image that represents the object with less than specified number of pixels are reassigned with value of 1. Therefore the small objects are becoming white and blended with the background. This process is known as opening in morphology.

Next step is to use connected component labeling to label each panel. Search the binary image for objects that has black pixels. Connected black pixels are identified as one object. The application scan the object from upper right to lower left. If the first scanned pixel of an object found, the object is labeled in sequence.

The fifth step is extracting the panels. The identified object are saved into matrices. The panels are extracted from matrix of the grayscale image using the object's coordinates. Object that has coordinates inside other object are not extracted.

The last step is to sort the panels to match the reading sequence of manga comics. The extracted objects are sorted to avoid false sequence of the panels that are supposed to be in the same row. This may be occurred if the image of comic page is slightly rotated.

2.2 Implementation of the method using MATLAB

The method is implemented using Image Processing Toolbox of MATLAB. It is designed as a function. The function accept a matrix of grayscale image as input argument. It returns several matrix of grayscale image that contain the extracted comic panels. The function's outline:

```matlab
function segmentOut = segment(segmentIn)
& Code for the method
end
```

To convert the input to binary image, im2bw function can be used. It produce binary image that has values of 1 (white) for all pixels in the input image with luminance greater than specified and 0 (black) for all other pixels. The process is called thresholding. Code for converting input image to binary image:

```matlab
bwImage = ~im2bw(segmentIn,0.5);
```

bwImage is the binary image. segmentIn is the input image. The thresholding level is 0.5. It means all pixels that has gray level greater than 255*0.5 are considered white and all other pixels are considered black. The binary image needs to be complemented using ~ because the connected component labeling function in MATLAB detect white pixels as object rather than the black one.

To remove small objects, bwareaopen function can be used. It morphologically open binary image. It removes from the binary image all connected component (objects) that have fewer than specified pixel. Code for removing small objects:

```matlab
pixelToRemove = 1000;
bwImage = bwareaopen(bwImage, pixelToRemove);
```

The specified value for pixelToRemove is 1000. The bigger the number, the more object are removed from the binary image and risk to remove the comic panel. The lower the number, the fewer object are removed from the binary image and can make the comic panel extraction need longer time to complete. In testing section, author compare several number of pixel to determine the specified value to minimize needed time while no panels are missed from extraction.

To label the connected component, bwlabel function can be used. It search all connected component in binary image and label them. The default labeling order is from upper left to lower right, column first. The order for reading manga comic is from upper right to lower left, row rst. To get the desired order, the image and the label matrix needs to be transposed. Code for labeling connected component in the binary image:

```matlab
[labelComp totalComp]=bwlabel(bwImage');
labelComp=labelComp';
```

The comic panels are considered the larger objects that would be found in the image. The objects inside the panel are to be avoided in extraction. Only objects that are not inside any object is extracted. Code for extracting the comic panels from labeled component:

```matlab
totalPanel = 0;
```
for i=1:totalComp
[r,c] = find(labelComp==i);
end

for j=1:totalComp
if i ~= j
    if (min(r) >= min(r1)) & ...
        (max(r) <= max(r1)) & ...
        (min(c) >= min(c1)) & ...
        (max(c) <= max(c1))
        isInside = 1;
    end
end
end

if isInside == 0
    totalPanel = totalPanel+1;
    segmentOut{totalPanel} = segmentIn(min(r):max(r), ...
    min(c):max(c));
    segmentAnchor{totalPanel} = ...
    [min(r),min(c)];
end
end

% input file name and total iteration
fileName = input('Nama File : ', 's');
% read the file
imgIn = imread(fileName);
% call segment function, extract the % panels
imgsOut = segment(imgIn);
% write the extracted panels
for i=1:length(imgsOut)
    imwrite(imgsOut{i}, ...
    ['Test1_','num2str(i), '.jpg']);
end

If the comic page image is slightly rotated, the sequence of the panels can be wrong. The leftmost panel in the same row can be detected first if the image is rotated in certain degrees clockwise. It is because the panel's first pixel has higher position than the rightmost panel which has to be detected first. To overcome the possibility, the extracted panels needs to be sorted to ensure it has the right sequence. In the sorting section, author give 20 pixels tolerance for difference of the first pixel's row in the panels. If the difference are greater than 20 pixel's row, the panels are considered not in the same row. Code for sorting the comic panels :

for i=1:totalPanel
    for j=i:totalPanel
        if i ~= j
            if (abs(segmentAnchor{i}(1) - ...
                segmentAnchor{j}(1)) < 20) & ...
                (segmentAnchor{i}(2) < segmentAnchor ... ...
                [j](2))
                temp = segmentOut{i};
                segmentOut{i} = segmentOut{j};
                segmentOut{j} = temp;
            end
        end
    end
end

3 TESTING AND RESULTS

Before doing some testing, first here is the example to use the function. Figure 3 shows the example of comic page image and the extracted panels using the method by calling the function segment. Code for the example :
The first test is to test the function for its accuracy. This test is intended to determine how accurate the method in extracting the panels in comic pages and its sequence. Table 1 shows the results of the test. The results are classified into 2 groups such as "Correct" and "Incorrect". The term correct means that all the panels in the comic page is successfully extracted and have correct sequence. Author checks the extracted panels one by one. If all the result images are all of the panels from the comic page, no panel is missing, no result images that are not the panel, and the sequence is the same with the comic’s reading direction, then the panel extraction for the comic page is correct. Otherwise, the results are included to incorrect group. 109 pages from various comic titles are tested and 82% of them are extracted correctly.

<table>
<thead>
<tr>
<th>Comic Title</th>
<th>Total Page</th>
<th>Correct</th>
<th>Incorrect</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bleach Ch.007</td>
<td>21</td>
<td>17</td>
<td>4</td>
<td>81 %</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Comic Title</th>
<th>Total Page</th>
<th>Correct</th>
<th>Incorrect</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dragon Ball Ch.004</td>
<td>14</td>
<td>10</td>
<td>4</td>
<td>71 %</td>
</tr>
<tr>
<td>Naruto Ch.150</td>
<td>19</td>
<td>14</td>
<td>5</td>
<td>74 %</td>
</tr>
<tr>
<td>One Piece Ch.004</td>
<td>19</td>
<td>17</td>
<td>2</td>
<td>89 %</td>
</tr>
<tr>
<td>Yotsuba Ch.002</td>
<td>36</td>
<td>34</td>
<td>2</td>
<td>94 %</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>109</strong></td>
<td><strong>92</strong></td>
<td><strong>17</strong></td>
<td><strong>82 %</strong></td>
</tr>
</tbody>
</table>

The second test is to compute the time needed by the method to complete extracting panels from various amount and type of images. To do the test, the script is similar with the previous test, except now there are iterations to simulate the number of images. The variables in this test are number of iteration and the comic page file name.

Table 2 shows the results of the test. The columns represent various number of iterations. The rows represent the various images to process. The result is the time needed by the method to extract all the panels. It is shown in milliseconds. Increased number of images to process clearly makes the process need longer time to complete. But as shown in the chart in figure 4, the additional time needed to complete the process is linear with the increase of total image to process.

<table>
<thead>
<tr>
<th>Image 1</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>472</td>
<td>2305</td>
<td>4594</td>
<td>23104</td>
<td>47833</td>
<td></td>
</tr>
<tr>
<td>686</td>
<td>3408</td>
<td>6791</td>
<td>35448</td>
<td>72976</td>
<td></td>
</tr>
<tr>
<td>286</td>
<td>1461</td>
<td>2758</td>
<td>15670</td>
<td>29522</td>
<td></td>
</tr>
<tr>
<td>168</td>
<td>829</td>
<td>1662</td>
<td>8606</td>
<td>18160</td>
<td></td>
</tr>
<tr>
<td>210</td>
<td>1040</td>
<td>2049</td>
<td>10218</td>
<td>20283</td>
<td></td>
</tr>
</tbody>
</table>

Table 2. Test Results For The Method's Performance
There are several problems that are not resolved in this method. It always crops the area for the panels as rectangles. If the panels are not rectangles, some part of adjacent panels can be included. There are also some panel arrangements that cannot be extracted with correct sequence.

4 CONCLUSION

In general, the method can detect the panels and their sequence correctly. But there are several cases when it cannot detect them correctly. The method still needs further refinement because the comic page has many types and styles. But with more than 80% accuracy in extracting panels from comic pages and sorting them to the right sequence is considered a good result.

The method can be used as a base to develop a digital comic service for mobile devices with special features which is the ability to show each panel of the comic pages. It can also be improved in several aspects. There are comic styles that use black color as background. In this state, the method will recognize the comic page as only one object. The sorting section of the method can also get some improvement to correct the problem in extracted panels sequence which has been mentioned in previous chapters.

REFERENCE


