Locating Text in Web Images Using Image Based Approaches

Seongah Chin
Division of Multimedia, Sungkyul University
(solideo@sungkyul.edu)
Moonwon Choo
Division of Multimedia, Sungkyul University
(mchoo@songkyol.edu)

A locating text technique capable of locating and extracting text blocks in various Web images is presented here. Until now this area of work has been ignored by researchers even if this sort of text may be meaningful for internet users. The algorithms associated with the technique work without prior knowledge of the text orientation, size or font. In the work presented in this research, our text extraction algorithm utilizes useful edge detection followed by histogram analysis on the genuine characteristics of letters defined by text clustering region, to properly perform extraction of the text region that does not depend on font styles and sizes. By a number of experiments we have showed impressively acceptable results.

Key words: Web images, text segmentation, font recognition, document image analysis.

1. Introduction

As graphical information plays an increasingly important role in various areas, techniques for extracting text information from mixed textural/graphical images are being explored by computer vision researchers. Moreover very large amount of images on the Web are increasingly provided, conveying quite meaningful textual contents. Unfortunately, most existing search engines merely detect plan text information without seeking into Web images. Web images carry vast amounts of information and many images contain embedded or overlaid text. The text in images may be as meaningful as the image itself. For coding or understanding document images it is essential to identify and possibly separate text, images and graphic regions into physically distinct segments of the page in order to be able to process them properly [5]. In a typical document decomposition process, the first step is to perform a physical segmentation of the document [4]. Page segmentation plays an important role in the domain since the performance of the document understanding system as a whole greatly depends on the processing modules that precede OCR (Optical Character Recognition). Therefore the task of text image segmentation is one of the primary stages of most document processing [5].

Recent text segmentation schemes have

approached this problem in several ways such as: "bottom up": using connected component grouping from characters, words, and text lines to paragraphs or run-length smoothing algorithms [8],[10],[18],[17] or "top down":, by first splitting large blocks, and then subdividing each block to identify the specific regions [1],[2],[3],[13]. The complexity of character segmentation stems from the wide variety of fonts, the varying size of fonts, the poor quantity of binary images and multiple character sets such as different languages, and special symbols [11],[18]. Recent text segmentation schemes from Web images are very few since interest on this topic has been driven lately [14],[15],[16].

In the work presented in this research, our text localization algorithm for Web images utilizes useful edge detection followed by histogram analysis on the genuine characteristics of letters, to properly perform extraction of the text region that does not depend on any font styles, sizes, or language as well as special symbols. The basic idea in our approach is to use knowledge of various characteristics of fonts, such as bitmap fonts and scalable fonts, also referred to as object oriented fonts or vector fonts. What font detail can be utilized for text segmentation? A font is a collection of characters of a particular typeface that defines the shape of each character. Therefore, the representation of the font defines the typeface but not the size. Regardless of the font system there exists a somewhat fixed width of stroke to height ratio, independent of the character including special symbols and different languages. Since the distance of a combination of curves or lines determines the segment width, we endeavor to calculate segment widths and their number of appearance in order to identify font density function. Each font system has its own font density function, which makes it possible to identify fonts by the fact that mainly occupied segment widths of each font are accompanied with high probability in the density function.

We note text can be aligned any orientation with skewed angles. Highly accumulated angles in Hough space might have angle distance 90o, orthogonal angle view made by vertical and horizontal stroke in text image. Orthogonal angle view sufficiently provides peak detection after mapping image space into Hough space through O'Gorman and Clowes's methods [6]. We are able to rotate and translate edge images in terms of new sizes and mainly skewed angle. . In our case we are not necessarily dealing with documents. We are concentrating on the Web images, such as photographs, that happen to contain text.

Our proposed scheme is concerned with textual processing in general Web images (which may or may not be documents), to extract text blocks by removing the visual or graphical regions. In our work we start with gray images as our inputs not via binarization so as to reduce loss of genuine information. In the long run, we propose the image text segmentation, which extracts text region through a Segment Mode Selection Scheme using edge detection, histogram and character feature detection.

2. Previous Related Work

As the volume of image databases grows

rapidly in all areas, the incapability of capturing required textual information might result in loosing key messages for electronic documentation such as indexing, image database retrieval, cataloging and OCR. Most current segmentation techniques have been developed for binary text images and emphasized on document images. However, useful information for character segmentation may be lost by the binarization of gray scale images. Embedded text cannot be extracted properly. In recent years, several researchers have addressed the issue of gray scale images for text segmentation by [8], [11]. Some of the approaches propose utilizing topographic features and the variation of intensities in gray level images. Anil K. Jain [8] proposes the traditional bottom-up method based on connected components. His method contributes to efficient document representation. C.L.Tan [18] proposes text extraction using a pyramid scheme which extracted text, based on connected components, is grouped into words by a multiresolution pyramid indicating the location of the words. But this method will not operate properly when text regions touch graphic regions (that considered as a part of graphic region) and subsequently will not be extracted. But both methods still work on binary images. S. Lee et al. [11] suggests a new methodology for utilization of gray scale images for character segmentation. They use a projection and topographic recognition method to achieve this segmentation. They do not, however, apply their method to general gray scale images that contain both text and background image. While their method may be useful with gray-scale images, it is not directly applicable to the problem we are studying. Another scheme, proposed by C.L.Tan et al. [18], allows for the extraction of text from maps. They use a multi-resolution method (also known as a pyramid) to optimize performance of their algorithm. Their algorithm is successful in extracting text in all orientations from maps. However, it is not clear whether their method can be extended to general images containing backgrounds with various gray level intensities. Z. Lu [12] has proposed a method for extracting text from digital engineering drawings. The method deals successfully with difficult situations where text and graphics touch. This method also works with foreign languages and symbols such as Chinese characters. It appears that the method is useful in separating text from graphics but may not be useful for separating text from background visual data.

We are motivated to apply the image segmentation technique to image scenes, which include pure images, text scenes only or mixed ones; so called text segmentation in documentation image analysis. In the field of document image analysis, text segmentation (text extraction, character extraction) has been mainly emphasized as an interest operator that performs text image analysis on two regions; text area and non-text area prior to undertaking the recognition tasks. What characteristics make text extraction difficult? First, we note that noise and degradation caused by scanning, copying, binarization and aging. Secondly, text regions touching or overlapping with graphic regions will be incorrectly classified using C Tan's method. Thirdly, text orientation can be at any skewed angle. Fourthly, the font size and type are not necessarily to be consistent all the time. Finally graphic image components usually are variant, which means it contains any kind of information such as lines, polygons in any shape, text information (scene text or graphical text), drawings and pictures. These issues always hinder us from extracting properly. Moreover, previous researches have been focused on document images rather than on complex Web images. The aim of this research is to propose general schemes, which make it possible to obtain textual information in Web images regardless of font, style, size, picture, languages, and not affected by background image content, skewed angle and noise as well.

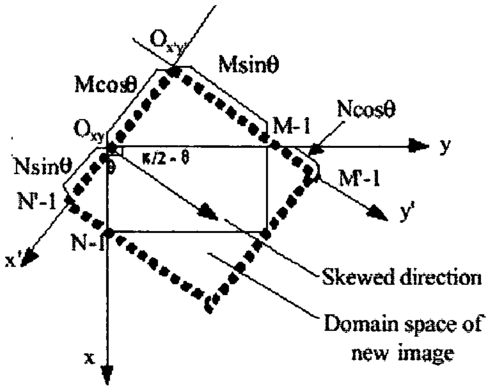
3. Methodology

The Text Extracting Filter (TEF) starts by operating on $f(x, y)$, an input image matrix where $0 \le x \le N$-1 and $0 \le y \le M$-1. Text areas in an image consist of narrow scans that contain sequences of several segments of equal width. Segments in this context are image strips of uniform grey level and a scan is linear progression through the image along some orientation. The ratio of the height, of an aggregation of segments of equal width, to the width itself is an important feature for text segmentation. The first step is to create a derivative image $\dot{f}_\theta(x, y)$ so that image regions that have significant changes in gray intensity are highlighted. We expected the boundary of text characters to be identified with this operation. The next step is to mark the highlighted regions and create a 3- level image, $e_{\theta,\tau e}(x, y)$ (the directional edge image), such that image regions with gradual changes in gray intensity are set to zero, those regions with positive slopes are set to +1 and those with negative slopes to -1. This labeling allows us to recognize the start and end of a character segment. We may use the gradient image, $\dot{f}_\theta(x, y)$ to identify these edges. We define useful edges as occurring at regions where the gradient magnitude exceeds some threshold τ_{e}, τ_e is selected based upon the average image intensity using global threshold technique [9],[6]. It is important to distinguish between rising and falling edges because, in most cases, a letter segment will be between (a rising edge and falling edge) or between (a falling edge and rising edge).

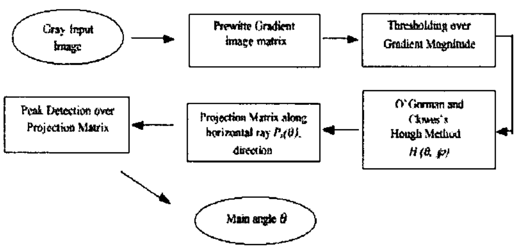
$$e_{\theta,\tau e}(x, y) = \begin{cases} 1 & \text{if } \dot{f}_\theta(x, y) > \tau_e \\ -1 & \text{if } \dot{f}_\theta(x, y) < -\tau_e \\ 0 & \text{otherwise} \end{cases} \tag{3-1}$$

If the text is skewed (i.e is not horizontally aligned) then the directional edge image is rotated so that the text is aligned correctly $e'_{\theta,\tau e}(x', y')$). This images discriminate increasing intensity edges from decreasing intensity edges rising and falling edges respectively.

Upon the selection of an orientation angle θ as shown in Figure 1, we will apply affine Transformations, i.e. a translation along the X axis and a rotation counter clockwise by the angle (π/2-θ). Now the edge matrix will be mapped into another matrix coordinate system, which is defined with respect to the text orientation angle θ. We explore the method, which adjusts for a skewed text image using transformations.



<Figure 1> Relationship of the new image axes (x', y') to the original axes(x, y) and skew angle θ



<Figure 2> Flow diagram of angle detection using O'Gorman and Clowes's version

Detection of text orientation is obtained by calculation of the O'Gorman and Clowes's version [6] of the image and a projection of the Hough Transform image matrix on the angle axis [6],[9],[16] as shown in Figure 2.

Since we have to transform all points in the edge matrix, we can proceed in two steps. At first, we calculate the combined transformation matrix, $T_{(M\cos\theta,\,0)}R_{(\pi/2-\theta)}$ then we use this combined matrix on each point. This combined matrix saves calculation time by using a single matrix multiplication at each point. We rotate the image matrix by $(\pi/2-\theta)$ radians counter clockwise and translate $M\cos\theta$ units in the X direction. Now, we realize that new transformed image matrix size has been changed from N to $N\sin\theta + M\cos\theta$ and from M to $M\sin\theta + N\cos\theta$ respectively as shown in Figure 1.

The following derivation explains the procedure performs transformation based on homogeneous coordinate system and calculation of new image size.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} x\sin\theta - y\cos\theta + M\cos\theta \\ x\cos\theta + y\sin\theta \\ 1 \end{bmatrix} \qquad (3\text{-}2)$$

First of all, we rotate with angle $(\pi/2-\theta)$ and translate to X axis with $M\cos\theta$. But we use the combined matrix for reduce calculation time. We denote new N and M as $N' = N\sin\theta + M\cos\theta$ and $M' = M\sin\theta + N\cos\theta$ or $M' = \sqrt{N^2+M^2}\ \cos(\tan^{-1}\frac{M}{N} - \theta)$ respectively.

In the case of $\pi/2 < \theta \le \pi$, we should rotate the edge matrix by the counter clock wise by $\theta - \pi/2$ (as same as $R_{(\pi/2-\theta)}$) and translate to Y axis with $N\cos(\pi-\theta)$. We also have to change a new image size as similar as Case 1. At this point, we change some notations as N to N', M to M', x to x', y to y', $e_{\theta,\tau e}(x, y)$ to $e'_{\theta,\tau e}(x', y')$.

3.1 Edge Matrices Index

By examining the edge matrix, $e_{\theta,\tau e}(x, y)$ we introduce two new edge index matrices, which store the column locations of individual edges. Edge index matrices, $K_s(x, c)$ and $K_e(x, c)$ hold the column indices of certain non zero elements of the edge matrix. In order to specify these matrices we need to define valid and invalid segments. Valid segments are scan line segment in the image that either

(a) starts with a +1 element in the edge matrix and ends with a -1 element or
(b) starts with a -1 element in the edge matrix and ends with a +1 element.

Invalid segment starts and ends with +1 and +1 or -1 and -1 respectively.

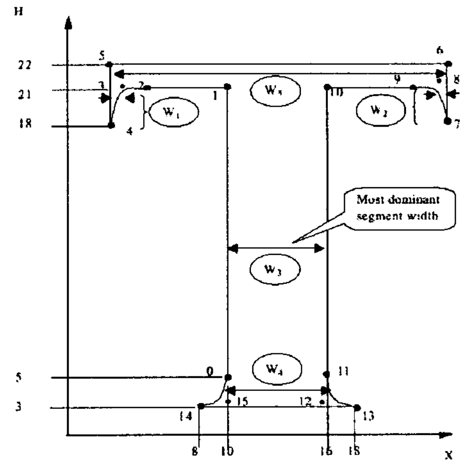
Now, $K_s(x, c)$ stores the image column index of the start of the c^{th} valid segment in the X^{th} scan line. In other words, if $K_s(13, 4)=89$ then the fourth valid segment in the 13^{th} image scan line starts at column location 89 in the image. Similarly, $K_e(x, c)$ starts the image column index of the end of the c^{th} valid segment. Let c_x where $x=0$ *to* N-1, hold the maximum valid segment count values in each scan line and let $\hat{c}$ be largest of these counts.

3.2 Segment Width Matrix

Now, we define $S(x, c)$ as the c^{th} valid segment width in x^{th} horizontal scan. Generally, segment width is the distance between the column locations of edges of which transition occurs either -1 to 1 or 1 to -1. $S(x, c) = K_e(x, c) - K_s(x, c)$. Note that we expect $e_{\theta, \tau e}(x, K_s(x, c)) = -e_{\theta, \tau e}(x, K_e(x, c))$. The following description details steps to build the edge index matrices, starting column location of edge detected $K_s(x, c)$, ending column location of edge detected $K_e(x, c)$ and segment width matrix, $S(x, c)$.

3.3 Segment Mode and the Histogram Matrix

When the segment matrix contains many segment widths with the same or similar value then there exists a possibility that these segments are textual segments based on the TEF principle. Of course regular patterns, grids or fence type structures in an image may also contribute to valid segments that produce many identical segment widths. In Figure 3 we have seen various types do segment width of letter 'T' with dominant segment width W_3. Further processing will be required to eliminate these non-textual patterns. We will construct histograms of the scan lines and select as candidate typeface widths the modes of these histograms. The expression for the histogram may be simplified by introducing the discrete delta function.



<Figure 3> Various types of segment width and dominant segment width W_3

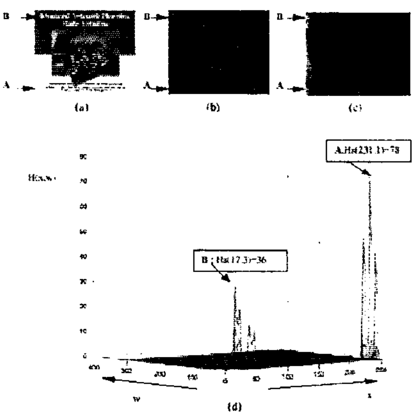
The scan line histogram matrix on the segment width matrix is :

$$H_s(x,w) = \sum_{i=1}^{c_x} \delta\,(s(x,i) - w) \tag{3-3}$$

$$\forall w = 1 \;\; to \;\; \hat{s} \text{ and } x = 0 \;\; to \;\; N-1$$

An example of this computation is shown in Figure 4. In Figure 4 (d) peak A corresponds to $H_s(231,1)=78$ implying that original image in scan line 231 there are 78 segment of width 1.

On examining the original image in 12 (a). One can see that this A refers to the scan line cutting the area of the text at the bottom "before deploying new applications, adding users", which contains 41 letters resulting in a high segment count. Peak B in Figure 4 (d) corresponds to "Advanced Network Planning"



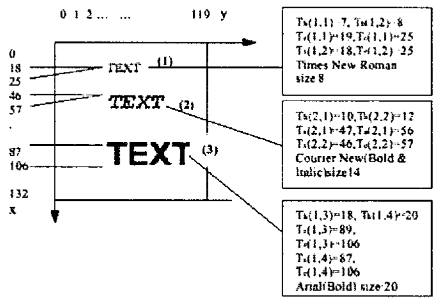
<Figure 4> High peak segment count at text areas

3.4 Text Clustering Region

The ratio of width to height of a typeface is useful for identifying regions of the histogram that contains text. We calculate the height of text characters from the histogram matrix by counting the number of consecutive occurrence in which its segment count is larger than equal to a threshold. The threshold is obtained by computing an average segment width histogram, $h_s(w)$:

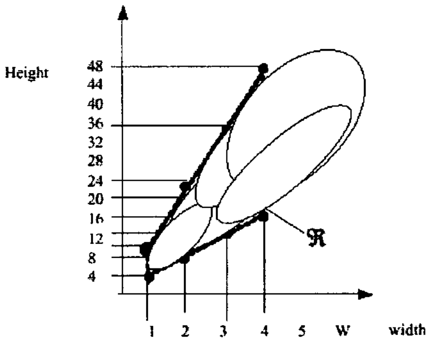
$$h_s(w) = \frac{1}{N}\sum_{x=0}^{N} H_s(x, w) \qquad (3\text{-}4)$$

Note that, $h_s(w_1)=h_1$ implies that in the entire image there are Nh_1 valid segments of width w_1. Clearly a large value of h_1 implies that there are many characters whose typeface stroke width is w_1. Those elements of the histogram matrix which exceed the column average ($h_s(w)$) are marked in a Binary Height Matrix. A sequence of continuous ones in the columns of the matrix identifies image rows that possibly contain text. We introduce matrices to catalog the locations and heights of the column text lines in the image. $T_h(r,w)$ - height of r^{th} column segment with w width $T_s(r,w)$ - starting row location of r^{th} column segment $T_e(r,w)$ - ending row location of r^{th} column segment as shown in Figure 5 given several typefaces.



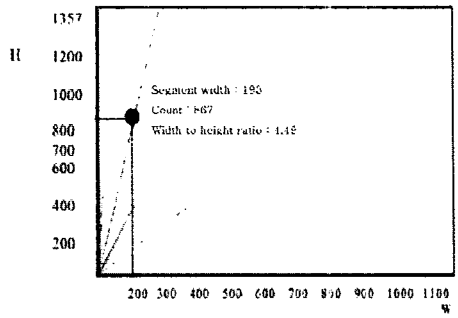
<Figure 5> $T_h(r,w)$ - height of r^{th} column segment with w width $T_s(r,w)$ - starting row location of r^{th} column segment $T_e(r,w)$ - ending row location of r^{th} column

Figure 6 shows a clustering region and a mapping from input image space into a height/width space. The region ____ in the height/width space represents valid text. This region is obtained by examining and measuring valid typeface. The region shape will change with different typefaces. The height/width space therefore constitutes our feature space in this problem. Therefore, if $T_h(r, w) \in \mathfrak{R}$ then the corresponding row location $T_s(r, w)$ and $T_e(r, w)$ identify rows in the image space that contain text line. Based on the binary height matrix $B_h(x, w)$, we develop an algorithm which allows us to make reasonable decisions concerning whether each text height falls into the the region $\mathfrak{R}$ or not as shown in Figure 6.



<Figure 6> Clustering Region $\mathfrak{R}$

Given the text height matrix, $T_h(r, w)$, associated with each width w starting row location, $T_s(r, w)$, and an ending row location, $T_e(r, w)$. We can investigate the ratio between width of each stroke and height and see whether it is suitable for the text character ratio satisfying $T_h(r, w) \in \mathfrak{R}$. We mark the binary height matrix $B_h(x, w)$ as 1 and modify the text height matrix $T_h(r, w)$ if and only if the height/width ratio is member of the clustering region. In other words, the text height, which is not be a member of the clustering region $\mathfrak{R}$ will be filled with 0s. For the experiments we simplified the region $\mathfrak{R}$ so that 4 $\le$ height/width ratio $\le$ 12.



<Figure 6> Clustering Region $\mathfrak{R}$ for Times Roman 'T' computed by glyph data

In Figure 6 most dominant segment width appears 867 times at segment width 193 generating width to height ratio 4.49.

3.5 Text Block Segmented Image

Since we have modified text height matrix $T_h(r, w)$, it helps us to recognize which rows are starting and ending locations associated with the text area possibly. We only need to look over those row of which segment width $S(x, c)$ is same as its width associated at $T_h(r, w)$. Then apply backtracking to edge index matrices $K_s(x, c)$ and $K_e(x, c)$ which are starting and ending location that segment width $S(x, c)$ with specific text height row starting and ending location $T_s(r, w)$ and $T_e(r, w)$ respectively.

Our alternate goal is to find text block so that we just extract the minimum of starting column location Bs(r, w) = min{bs(x)} and maximum of ending column location Be(r, w)=max{ be(x)} in a specific text block where bs(x)=Ks(x, c) and be(x)=Ke(x, c) associated with S(x,c)=w given x, a specific text height location, i.e Ts(r, w) $\le$ x $\le$ Te(r, w). For efficiency, we investigate individual segment width two directions. For searching starting location, we do sequential search whereas backward when doing ending location. In order to extract text block, minimum starting location and maximum ending location will be discovered in a specific text height. Now we pay attention to segment width matrix S(x, c) (a) which will specify with c, where the given segment width starts and ends via edge index matrices Ks(x, c) (b) and Ke(x, c) (c) only when that segment width is exactly same as w given by $T_h(r, w)$. In this point, we trace the edge index matrix with r=1,w=1, and $T_h(1,1)=6$ where vertical arrow indicates height range and each bold numeric data will be referenced through the procedure. Finally we figure out the minimum of starting location will be figured out and maximum of ending location will be found after traversal.

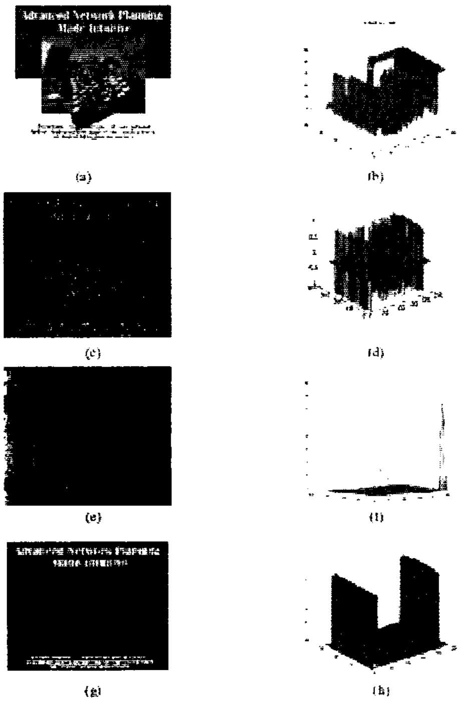
4. Experimental Results

We carried out experiments on a number of image files classified five groups such as book cover, commercial bill board from magazines, web images, document images and other languages in order to evaluate correctness of the proposed schemes under Pentium4 Compaq Desktop workstation operated by Window 2000 professional and Matlab5.3 environments. Image files are captured from the scanner or digital camera and executed for experiments whereas Web images goes into the algorithm without capturing stage. In our experiments we found that 10 to 12 point size of text mostly used in documents should be captured with a minimum of 300dpi (dot per inch) resolution since lower resolution cause significant data loss especially with small type fonts. Text in all other types of images is good enough to be extracted with 96dpi resolution.

Images can be roughly classified text in monotonous background images and text embedded in sophisticated images. Even if our work is intentionally developed to make it suitable for complex images, it also works monotonous background images as well. The results imply that this method is superior to existing dominant text segmentations, which only regard to extract text from document images. We still have the optimization solution to figure out the efficient threshold (τ_e) of the gradient images and the segment counts when selecting proper segment mode candidates. By experiments, optimal threshold is heuristically selected until expected text blocks are detected. We solved skew angle estimations for images that contain text that is not aligned horizontally. In most images text areas are aligned along the horizontal axis, however there are many occasions when the text may be at a skewed angle (denoted by $0 < \theta \leq \pi/2$ as illustrated in Figure 1). The result of this method is a primary text skewed angle, which allows us to rotate the original input image into an image with horizontally aligned text.

Since we have already introduced rotation and translation in section 3 in detail, so that we abbreviate those schemes in details.

Skewed angles associated text images, is solved using orthogonal angle view, O'Gorman and Clowes's version [6] and their projection methods in terms of technical examples as well as theoretical view as shown in Figure 2. Most importantly, we should discuss the topic to concrete text clustering region in which text should be determined by the reasonable fact.



<Figure 7> Experimental Results

(a) Input image, (b) Mesh graph of (a), (c) Edge image, (d) Mesh graph of (c), (e) Histogram image, (f) Mesh graph of (e), (g) Text block location image, (h) Mesh graph of (g)

As we have mentioned in section 3 approximate height/width ratio is to be 4 to 12 . Capital letters tends to associate with higher ratio than small letters. By the way, each font is defined by its own glyph data, which consists of straight lines or Bezier curves.

(a) (b)

<Figure 8> Experimental Results

(a) Input image with salt and pepper noise added (b) Text extracted images

Thus, we can derive theoretical clustering region by segment width density of each character in the future research. We have conducted selected trials with various parameters in terms of image size, resolution, threshold, primary height/width ratio, skew angle, processing time, correction ration and extraction rate(%) in sequence. However, we still happen to see the limitation on separating text areas, which have similar gray level to back ground. We have evaluated the algorithm properly works with noise added images as shown in Figure 8. Manually we have tried to adjust threshold in order to separate those regions. Also, we can not escape shortcomings such as extracted areas, which are non textual caused by identical segment height/width ratio to fonts.

5. Conclusion and Future Work

We have proposed an image text localization technique, which works without prior knowledge of the text orientation, size or font. Images convey vast amounts of information and many images contain embedded or overlaid text. The text in this image may be as meaningful as the image itself. Most current segmentation techniques have been developed for binary text images and emphasized on document images. However, useful information for character segmentation may be lost by the binarization of gray scale images. Embedded text can not be extracted properly.

In the work presented in this research, our text extraction algorithm for the Web images utilizes useful edge detection followed by histogram analysis on the genuine characteristics of letters, to properly perform extraction of the text region that does not depend of any font styles, sizes, or languages. The basic idea in our approach is to use knowledge of various characteristics of fonts, such as bitmap fonts and scalable fonts, also referred to as object oriented fonts or vector fonts. What font detail can be utilized for text segmentation? A font is a collection of characters of a particular typeface that defines the shape of each character. Therefore, the representation of the font defines the typeface but not the size. Regardless of the font system there exists a somewhat fixed width of stroke to height ratio, independent of the character including special symbols and different languages. Since the segment width is determined by the distance of a combination of curves or lines, we endeavor to calculate segment widths and their number of appearance in order to identify font density function.

We have carried out experiments on the huge number of Web image files. In addition, the algorithm properly works with five groups such as book cover, commercial bill board from magazines, document images and other languages. In our experiments, the extraction rate calculated by the ratio, extracted text blocks over total text blocks has shown the proposed schemes are reasonably accepted with high accuracy. Upon realizing that this technique is designed for font, size and multilingual independent characters. This algorithm can also work for with skewed images by performing the rotation and translation. As for the future work, we can extend this to text font detection by building font density functions for different fonts. An extension of this work can lead to image database retrievable system by automatically reading of text in Web images. In addition, we are able to identify each font using font density function of segment widths, which combines mainly occupied segment width its probability as future work.

REFERENCES

[1] Amamoto, N., Torigoe, S., Hirogaki, Yoshitaka, "Block Segmentation and Text Area Extraction

of Vertically/Horizontally Written Document", IEEE Proceedings, pp. 739-742, July 1993

[2] Antonacopoulos, Apostolos, "Page Segmentation Using the Description of the Background" Computer Vision and Image Understanding, Vol. 70. No. 3, pp.350-369, June 1998

[3] Chen, Francine R. and Bloomberg , Dan S., "Summarization of Imaged Documents without OCR" Computer Vision and Image Understanding, Vol. 70. No. 3, pp. 307-320, June 1998

[4] Doermann, David, "Indexing and Retrieval of Document Images": A Survey Computer Vision and Image Understanding, Vol. 70. No. 3, pp. 287-298, June 1998

[5] Etemad, K., Doermann, David and Chellappa , Rama "Multiscale Segmentation of unstructured Document Pages Using Soft Decision Integration", IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 19, No 1, pp. 92-96, Jan. 1997

[6] Haralick, Robert M., Shapiro, Linda G., Computer and Robot Visioin, Volume1, Addison Wesley,1992

[7] Haralick Robert M., Shapiro, Linda G., Computer and Robot Visioin, Volume2, Addison Wesley, 1993

[8] Jain, Anil K., Yu, Bin, "Document representation and its Application to Page Decomposition" IEEE Transaction on Pattern Analysis and Machine Intelligence,Vol. 20 No.3, pp. 294-308, March 1998

[9] Jain, Ramesh, Kasturi, Rangachar, and Schunck, Brian G., Machine Vision, McGraw-Hill, Inc,1995

[10] Kise, K. Sato, Akinori, and Iwata, Motoi "Segmentation of Page Images Using the Area Voronoi Diagram" Computer Vision and Image Understanding, Vol. 70. No. 3, pp. 370-382, June 1998

[11] Lee, Seong-Whan, Lee, Dong-June, Park, Hee-Seon, "A New Methodology for Gray-Scale Character Segmentation and Recognition", IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 18, No. 10, pp. 1045-1050, October 1996

[12] Lu, Zhaoyang, "Detection of Text Regions from Digital Engineering Drawing", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20, No. 4, pp. 431-439, April 1998

[13] O'Gorman, Lawrence and Kasturi, Rangachar, "Document Image Analysis", IEEE Computer Society Press,1995

[14] Okun, O. and Pietikainen, Matti., "Text Localization in WWW images", Proc. of the 5th Multiconference on Systemics, Cybernetics and Informatics, Vol XIV, July, pp301-306 2001

[15] Park, T., Kim, D. and Chung, "Orientation and Scale Invariant Text Region Extraction in WWW images", Proc. of IAPR Workshop on Machine Vision Applications, Makuhari, pp 290-293, 1998

[16] Schmidt, M., Kreinovich, V. and Longpre, L., "Lolmogorov Complexity-Based Ideas for Locating Text in Web Images, Proc of the 42nd Midwest Symposium on Circuits and systems, pp 543-546, 1999

[17] Slimon, A., Pret, Jean-Christophe, and Johnson, A Peter, "A Fast Algorithm for Bottom-Up Document Layout Analysis", IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 19, No. 3, pp. 273-277, March 1997

[18] TAN, C.L. and NG, P.O., "Text Extraction Using Pyramid", Pattern Recognition. Vol. 31. No 1 pp. 63-72, 1998

요약

웹 이미지로부터 이미지기반 문자추출

진 성 아*
주 문 원*

본 논문은 다양한 웹 이미지로부터 문자영역(text block)의 위치를 알아내고 문자영역을 추출하는 방법을 제안한다. 인터넷 사용자관점에서 볼 때, 웹 이미지에 포함되어 있는 문자정보는 중요한 정보이지만 최근까지 이 분야의 연구는 그리 활발하지 못했다. 본 연구에서 제안된 알고리즘은 문자의 경사방향(skew)과 문자의 크기나 폰트에 관한 사전 정보 없이 수행되어 질 수 있도록 제안되었다. 폰트 스타일과 크기에 제약되지 않고 문자영역을 적합하게 추출하기 위해 유용한 에지 검출, 문자 클러스터링 영역으로 정의되는 문자의 고유한 특성을 위한 히스토그램을 사용하였다. 다수의 실험을 통하여 제안된 방법을 테스트하고 수용할 만한 결과를 도출했다.

핵심어: 웹이미지

* 성결대학교 멀티미디어학부