Word Location Techniques for Text/Graphic Mixed Images Using 3-Dimensional Graph Model

Se-Young Ock, Hwan-Chul Park, Hwan-Gue Cho

Graphics Application Lab.,
Department of Computer Science,
Pusan National University, Kum-Jung-Ku, Pusan 609-735, Korea.
E-mail: {seok,hcpark,hgcho}@pearl.cs.pusan.ac.kr

ABSTRACT

This paper presents a new extracting method for several types of texts from a text/graphic mixed document image. We also propose a new word grouping method when intersected words are placed on a circular arc or any line segment with an arbitrary orientation. The basic strategy of our algorithm is based on the analysis of the run-length of the document image. The average and variance of the number of runs in a run-length encoding provide a nice structural property for symbols and texts. We propose 3-dimensional neighborhood graph for grouping word from a set of isolated characters, which are obtained from the first character-isolating phase. This graph maps each letter to a vertex in 3-dimensional space according to the "volume" of the character. Experimental results show that more than 97% of words were successfully extracted from a text/graphic mixed document.

Keywords: document analysis, pattern recognition, text extraction, image processing

1 INTRODUCTION

There is an ever-growing demand for electronic and computerized document management, storage, and calculation. Document structure analysis and understanding are the main processes in reaching these goals. In general, when documents such as newspapers or magazines are converted into a machine-readable form, this procedure is solved manually. Since manual work is an inefficient and error-prone process, there is a need for an automated text string extraction system. In this paper, we propose a novel algorithm that automatically extracts text and non-text elements in a document image file, and we group the extracted characters into words using 3D neighborhood graph.

1.1 Related works

Some approaches have been proposed for automatically extracting text strings in text/graphic mixed document images such as engineering drawings, newspapers and magazines [FLE98-EIK95]. Fletcher described one method that uses the information on each connected component in a text/graphic mixed document image[FLE88]. The Hough transform was applied to these separated texts in order to group characters together into text strings [FLE88]. For grouping a word from each separated character, Burge applied new technique based on area voronoi tessellation[BUR95]. Other methods regard texts information as textured objects and uses well-known methods of text analysis such as Gabor filtering, but this method is sensitive to font size and style, and as a result is generally time-consuming[AJA92]. Tan proposed a model, namely a pyramid model, to extract text strings based on connected component generation. The pyramid model efficiently and quickly identify and locate words or phrases placed in an image file. However, the efficiency is decreased when characters are required to be grouped into words[TAN97]. Recently, Kamel proposed a method that separates text areas using grey scale information in document images with a mixed background images, shadows, and highlight images. This technique is based on the filtering method, interpolation scheme, and adaptive thresholds using the grey level of pixels[KAM93]. Liang exploits the spatial configuration of the bounding boxes of different entities on a given document image. This algorithm segments an image into a list of homogeneous zones[LIA96]. At present, applications of such text-extraction or graphics-extraction systems include 1)converting paper-based drawings for integration into a CAD/CAM environment, 2)automatic processing of postal addresses[DOW98], 3)pictorial databases, 4)document analysis system, 5)archiving of past copies of newsletters, and 6)business card readers[SAI82].

1.2 Main problems

Generally, the automatic text separation is difficult since each document varies in style and contents. This text separation procedure becomes very problematic if document has various fonts and sizes of text, and the text orientation is allowed to be arbitrary. There has been little research that extracts negative text strings (white plain characters on black plain background) such as newspaper headlines. When, negative texts and mixed text/graphical components are exist, the separation will be more difficult. There are two major technical difficulties in extracting text strings from text/graphic mixed document image.

- There are various sizes of characters, and the gaps between characters or between words are also various. Moreover, text strings can be arbitrarily placed in any direction, so we do not know in advance the orientation of each text. Fig.1 is an example of a text/graphic mixed document image that has various sizes and orientations of text strings.
- Fig.1 shows a document with a negative text, text, and graphical component.

2 CHARACTER EXTRACTION

The text in brochure images and newspapers is allowed to have a variety of fonts, sizes, styles, and spacings.

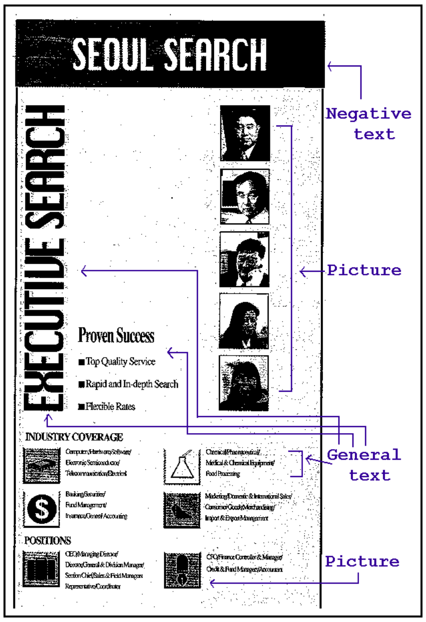


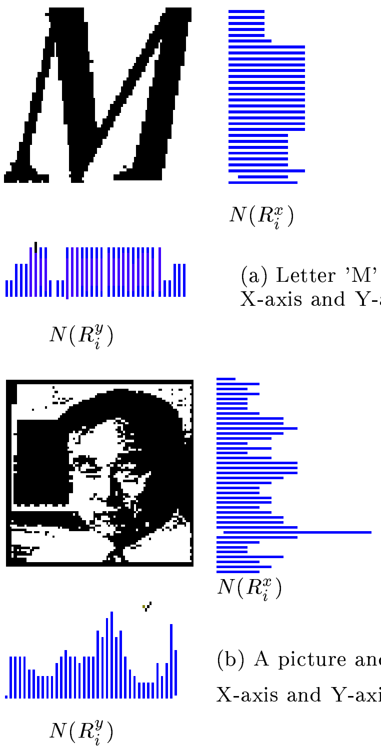
Figure 1: A document image including negative texts and mixed text/graphic

2.1 Connected components extraction

In order to isolate a character set, we have to find a connected component in document images. A connected component is computed using an 8-connected region generation algorithm[FLE88]. Let C_i denote a connected component(=pixel cluster). After finding all connected components, we should prepare the following attributes:

- $B(C_i) = \langle W_i, H_i \rangle$, Bounding box for a connected component C_i, where the width is W_i, and the height is H_i
- $Pr_i = Px(C_i)/B(C_i)$, The ratio of the number of pixels to the bounding box area, where $Px(C_i)$, The number of pixels in the connected component C_i.
- Wr_i, Aspect ratio of bounding box($Wr_i = W_i/H_i$)

If the number of pixels in a connected component is less than a threshold value, c_0, we regard it as a pepper and salt noise, and it will be removed from the document image. Threshold constant $c_0 = 6$ was determined by several experimental works. We classify all connected components into three classes (text, negative text, and graphic symbol or picture).



(a) Letter 'M' and its X-axis and Y-axis run profile

(b) A picture and its X-axis and Y-axis run profile

Figure 2: comparison for the number of runs in text and picture

2.2 Classification of a connected component

General documents such as brochures and newspapers have many different types of texts, lines, and figures. In this section, we describe an extraction technique that determines types of connected components. It is not easy to distinguish characters from non-characters by the size of connected components. There are two technical difficulties: one is that some graphic symbols and characters are very similar in terms of bounding-box size; the other is that we cannot tell which is a graphic symbol and which is a negative character, since the size and the number of pixels in these two components is nearly equal. Generally, the connected component of graphical symbol is larger than that of text in a document image. However, in case of negative texts, the connected component is similar to that of a picture.

The spatial configuration of bounding boxes of connected components can be obtained by projecting them onto a straight line. The projection onto a vertically(horizontally) straight line is called horizontal(vertical) projection.

Now we encode the connected component image by the run-length method. We define $R = \langle R^x, R^y \rangle$ as a binary sequence. $R^x(R^y)$ is the run of horizontal(vertical) projection. Let $N(R)$ denote the change number of runs in R. So, if R^x is "11101101011," $N(R^x)$ should be 6. Similarly, if R^y is "1111000011111," $N(R^y)$ should be 2. Fig.2 shows the horizontally and vertically projected sample profiles of a character and a picture. $N(R)$ of Fig.2(a) seems regular, but $N(R)$ of Fig.2(b) is irregular. So, we can see that the variance of $N(R)$ of graphic symbols is larger than that of texts. We label each connected component as a general text if its bounding box size is smaller than an average bounding box size of connected components; otherwise, it is labeled as a large component. In order to separate large components, we can equally apply the following algorithm. First, we check the Wr_i and the Pr_i of a connected component. If Wr_i is larger than threshold, l_0, the C_i is regarded as a negative or picture. Because general letters have normal $Wr_i(1 \leq Wr_i \leq 2 \; or \; Wr_i \leq 0.5)$. For that reason, we determined that l_0 is 2.5 by experiments. Next, we can accurately classify a negative text by $N(R)$, $\overline{N}(R)$(average of difference of two adjacent $N(R)$s), and $\widehat{N}(R)$(variance of difference of adjacent $N(R)$s). $\widehat{N}(R)$ of a negative text is smaller than that of graphic symbol. Fig.3 describes the classification procedure to determine if a given image is a positive character, a negative character or a picture. $\overline{B}(C)$ in Fig.3 is the average size of bounding box for all connected components. Our average and variance of $N(R)$

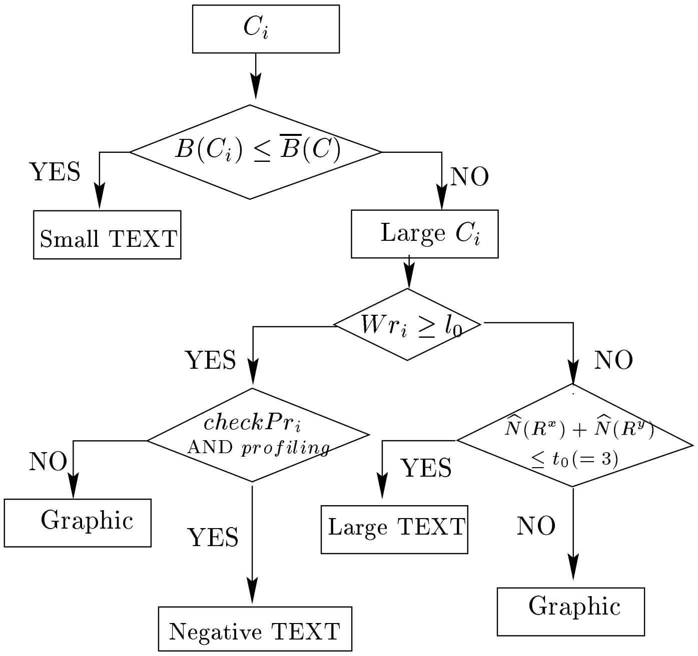


Figure 3: Flow of classification algorithm

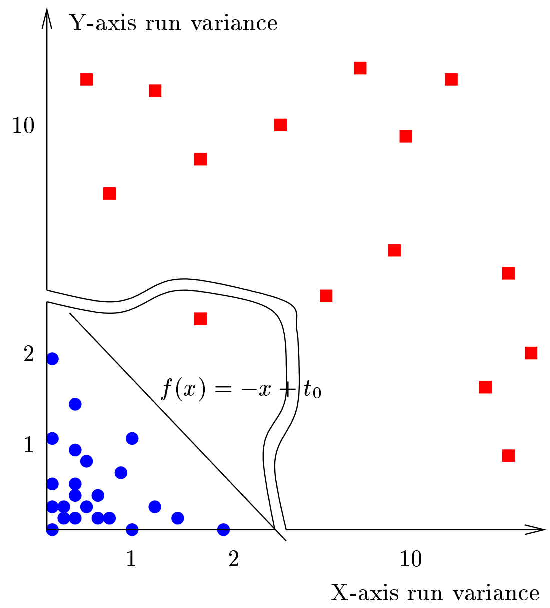
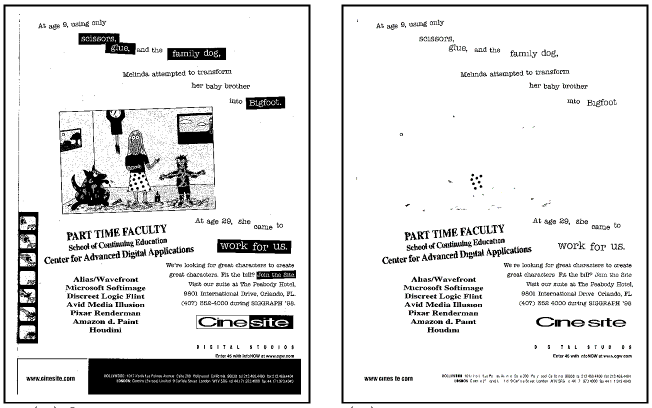


Figure 4: Classification for symbol and letter:circle dots represent letters, rectangles represent several symbols(=picture icons)

are newly defined in the following :

$$
\begin{aligned}
\overline{N}(R_i^x) &= \frac{\sum_{a=1}^{H_i} |N(R_a^x) - N(R_{a+1}^x)|}{H_i} \\
\overline{N}(R_i^y) &= \frac{\sum_{b=1}^{W_i} |N(R_b^y) - N(R_{b+1}^y)|}{W_i} \\
\widehat{N}(R_i^x) &= \frac{\sum_{a=1}^{H_i} (|N(R_a^x) - N(R_{a+1}^x)| - \overline{N}(R_i^x))^2}{H_i} \\
\widehat{N}(R_i^y) &= \frac{\sum_{b=1}^{W_i} (|N(R_b^y) - N(R_{b+1}^y)| - \overline{N}(R_i^y))^2}{W_i} \\
& \quad where\ 1 \leq a \leq H_i\ ,\ 1 \leq b \leq W_i
\end{aligned}
$$

Note that we define a new characteristic parameter, $\widehat{N}(R_i)$, which is the variance of every two adjacent runs in R_i. If $\widehat{N}(R_i)$ is larger than a given threshold t_0, the connected component is regarded as a graphic component or a pictures. For example, both $\widehat{N}(R_i^x)$ and $\widehat{N}(R_i^y)$ are close to 0 in Fig.2(a). However, in Fig.2(b), the $\widehat{N}(R_i^x)$ is 18, and the $\widehat{N}(R_i^y)$ is 6. Therefore, we can find the value which discriminates between text and graphic. Fig.4 shows a distribution graph for the variance $\widehat{N}(R_i)$ of each connected component. As you see in Table 1, each direction variance of text is small, so it is located near the origin. We can classify large connected component into two classes(text and graphic component) with a straight line, a linear classifier $f(x)$. The linear classifier $f(x)$ in Fig.4 determines the type of a connected component using a constant t_0. If $\widehat{N}(R^x) + \widehat{N}(R^y)$ of the connected component is larger or equal to t_0, the component is classified as a picture. Otherwise, the component is classified as a text. t_0 for classification of text and graphic was determined as a 3 by experiments. $\widehat{N}(R)$ of a negative text is also relatively small so it can be extracted text. As the result of profiling, we can successfully classify components into texts, negative texts, and graphic symbols by $N(R)$, $\overline{N}(R)$, and $\widehat{N}(R)$.



(a)Original test image (b) Text strings extracted

Figure 5: The result of Extracted texts using our algorithm

Name	*CLASS*	*Num*	$\widehat{N}(R^x)$	$\widehat{N}(R^y)$
DATA1	*PICTURE*	4	3	1.8
	TEXT	10	0.5	0.6
DATA2	*PICTURE*	1	7	2
	TEXT	10	0	0.4
DATA3	*PICTURE*	3	12.3	7.7
	TEXT	1	0	0
DATA4	*PICTURE*	11	14.7	18.5
	TEXT	1	1	1
DATA5	*PICTURE*	0	0	0
	TEXT	4	0	0.3
DATA6	*PICTURE*	1	37	1
	TEXT	1	0	0
DATA7	*PICTURE*	0	0	0
	TEXT	8	0.8	0.6
DATA8	*PICTURE*	0	0	0
	TEXT	6	0.2	0.5

Table 1: Test results of projection profile

Fig.5(a) is a text/graphic mixed document image and, Fig.5(b) shows the set of characters by our algorithm. In Fig.5(a), "work for us", "Bigfoot," "site," and "HOLLYWOOD..." are classified as negative texts. These negative texts are then recovered into general texts (positive texts) by our algorithm in Fig.5(b).

3 WORD GROUPING

We present a new algorithm for word grouping from extracted characters in this chapter. This algorithm introduces 3D neighborhood graph which is based on the area and location information of each connected component.

3.1 3D neighborhood graph

In this section, we propose a new 3D neighborhood graph model to group several characters, which are different in sizes, fonts, and orientations. By constructing a 3D neighborhood graph, we can obtain more contextual and geometrical information about characters. The basic idea of our word grouping procedure is that we map each character to a three dimensional box. We try to arrange a larger character to the upper space to make the distance to the smaller character large. So, larger characters will emerge to the upper plane, and the smaller will sink to the bottom. For that reason, this mapping philosophy forces the set of characters of nearly the same size to be placed in a horizontal plane.

3.2 3D-graph construction

As illustrated in Fig.6, characters of various sizes are placed in a document, and their placing orientation also differs from each other. Some words are placed on a curve rather than a straight line. Thus, the previous algorithms for word grouping used in a common document analysis cannot easily handle this kind of problem. In Fig.6, we can see that a statement of two words, "Transantarctic Mountains" cuts the relatively smaller statement of one word "ANTARCTIC." The main idea under the 3D neighborhood graph modeling, $G_{3D}(V,E)$, is that we map each character of several sizes onto 3-dimensional space and create edges, considering the distance between two mapped characters(=vertices). Let us define this formally. Let c_i denote a single character in a document. Let us give a formal $G_{3D}(V,E)$, where V is a vertex set and E is a edge set. We map c_i to a vertex v_i of $G_{3D}(V,E)$. And, v_i has a 3-dimensional coordinate, as $V_i(x_i,y_i,z_i)$. Since (x_i,y_i) is the centroid of c_i in a plane, we only need to define the height value z_i for c_i. Briefly, if the size of c_i is big(small), then it will be placed higher(lower) in (x,y,z) space. This placement strategy prevents from interfering between several sizes of the intermixed characters. Let $Area(c_i)$ denote the area of minimal enclosing circle of c_i. Equation (3) shows that height z_i is assigned by function of $Area(c_i)$. k_0 is an adjusting constant to control z_i-height in 3D space, which is set by experiments.

$$V = \{(x_i,y_i,z_i)\} \quad E = \{(v_i,v_j)|v_i,v_j \in V\}$$
$$\textit{where } (x_i,y_i) \textit{ is the centriod of } c_i \textit{ and}$$
$$v_j \textit{ is the nearest vertex from } v_i \textit{ in 3D space.}$$

$$\mu^* = \frac{1}{n}\sum_{k=1}^{n} Area(c_k) \tag{1}$$

$$\delta^2 = \frac{1}{n}\sum_{i=1}^{n}(Area(c_i) - \mu^*)^2 \tag{2}$$

$$z_i = \frac{k_0 \cdot Area(c_i) \cdot \delta}{\mu^*} \tag{3}$$

Every vertex has at least one edge connection

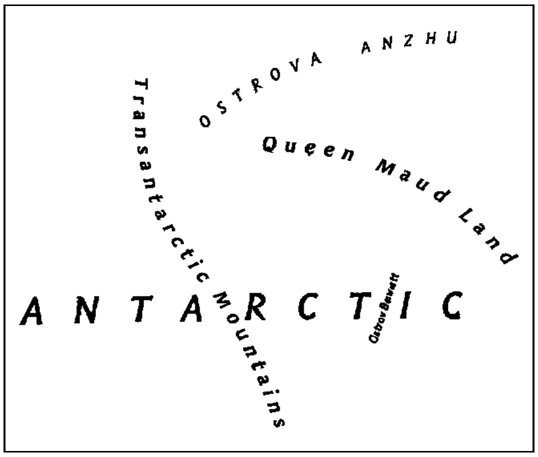


Figure 6: An example with intersecting words

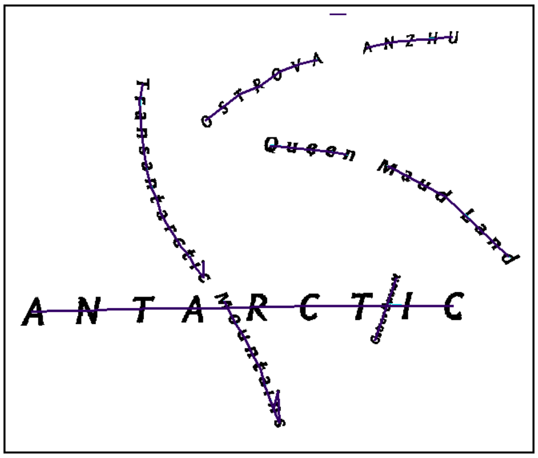


Figure 7: Grouped words of Fig.6

to its nearest vertex. Therefore, the degree of each vertex after this edge assignment procedure should be 1 or 2. In Fig.8, the higher the z_i value of c_i, the bigger the shape of vertex in $G_{3D}(V,E)$.

3.3 Word grouping in $G_{3D}(V, E)$

The goal of this procedure is to investigate G_{3D} to compute the *closely related subgraph(=word)* so that the members of the set of characters can be combined into meaningful character strings(=words). First, we must traverse the G_{3D} in depth first search method in order to cluster subgraphs, depending on the degree of connections of all nodes in G_{3D}. And then, we have to find a word. Since it is common that characters in a word are placed linearly and uniformly in documents, we assume that the centroids of characters in a word are placed nearly colinearly. Thus, we compute the degree of linearity by examining along a path(=a sequence of adjacent edges). Let θ_i be the angle between two successive edges, namely $E(v_{i-1}, v_i)$ and $E(v_i, v_{i+1})$, i.e., $\theta_i = \angle v_{i-1}v_i v_{i+1}$. Then, for v_i, we find the maximal index of k satisfying the following constraint.

$$\alpha^* = max_k \{ \frac{1}{n} \sum_{i=1}^{k} (\theta_{avg} - \theta_i)^2 \} \ \leq \ \alpha_0,$$

$$where \quad \theta_{avg} = \frac{1}{n} \sum_{j=1}^{k-1} \theta_j$$

where α_0 is a threshold constant for linearity. Though characters in a word are adjusted strictly on a base line, there are different types or number of strokes. The centroid cannot exactly be placed in a straight line. Therefore, α_0 cannot be 0. Finally, we traverse each subgraph and calculate the local threshold by considering the lengths of all edges in a subgraph in order to remove longer edges than a local threshold value. Fig.7 shows an example of word grouping for Fig.6. Fig.8 shows the 3D neighborhood graphs that correspond to Fig.7. Fig.9 shows 3D neighborhood graphs of Fig.8, which are viewed from a different direction.

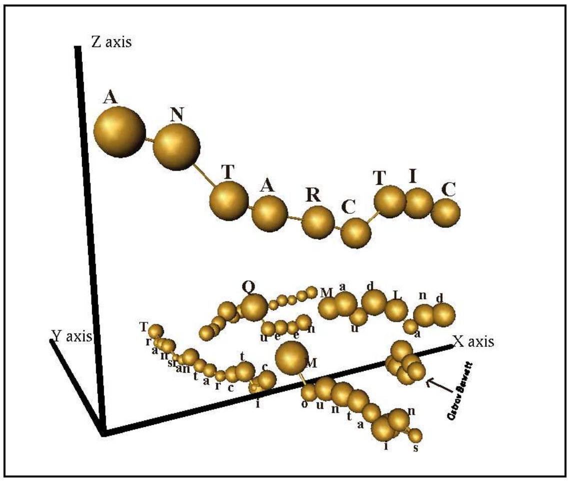


Figure 8: 3D graph of Fig.7

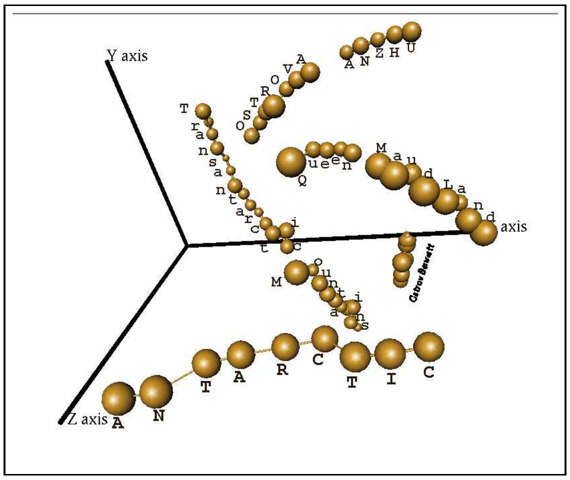


Figure 9: 3D graph of Fig.8 which is viewed from different direction

4 EXPERIMENTS

Experiments on layout structure classification and word grouping were carried on several document images obtained from books, brochures, and magazines. These text/graphic mixed document images were scanned to the raster images with 1000 by 1000 pixel resolution. Every testing image have various text fonts, sizes, and orientations. These documents also allow negative texts, i.e., the colors of text and background were reversed, compared to normal text images.

Fig.10 shows an original document image. Some texts are placed in a circular form, and negative texts and symbols are also included. The spatial resolution of Fig.10 is 892 $\times$ 994. The results after our character extraction procedure is shown in Fig.11. The different connected components(symbols, negative texts, and normal texts) are correctly identified in fig.11. Now, we need to group each characters into meaningful words. Fig.11 shows the result of word grouping procedure from Fig.10. In this Fig.11, all words are denoted as a stream of characters threaded with a dotted line. The vertical statement in the middle of Fig.11 was clearly grouped into a meaningful word. See a grouped statement "YOUR" and "CALENDAR" in the center of Fig.11. Fig.12 shows the corresponding 3D neighborhood graph of Fig.11, which was viewed from top to bottom(x-y plane). Fig.13 is another 3D graph that is viewed from different direction(z-axis). Fig.12 shows that a large letters(denoted as a big ball) are placed in a relatively higher position.

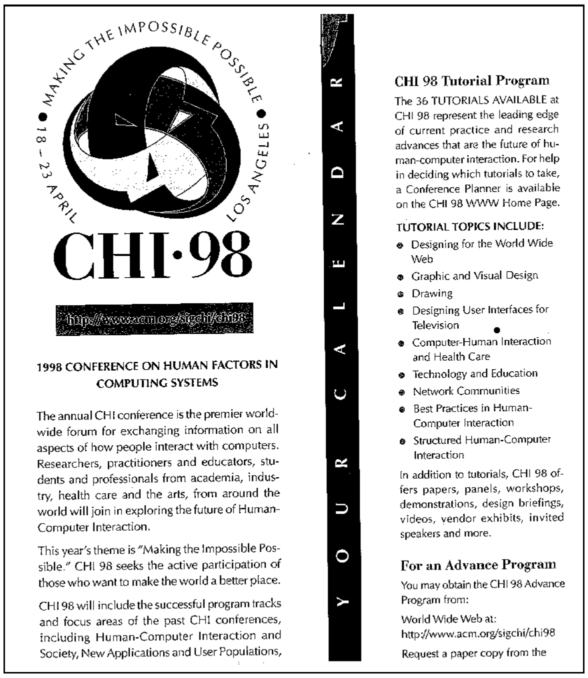


Figure 10: An original document image

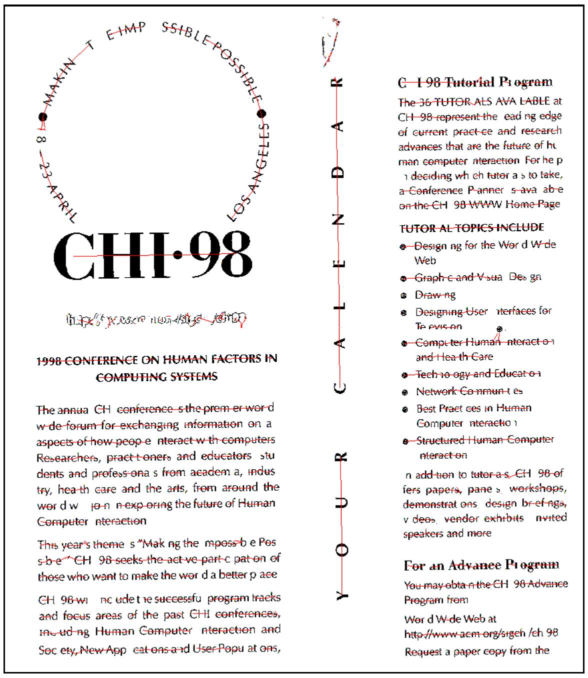


Figure 11: Grouped words from Fig.10: grouped words are marked by dotted lines

Name	*SIZE*	E_{SRR}	E_{FRR}	W_{SWG}	W_{FWG}
$DATA_1$	480 × 890	95.2%	2.3%	97.6%	1.8%
$DATA_2$	885 × 614	99.5%	0.3%	98.2%	0.4%
$DATA_3$	747 × 992	96.9%	1.4%	98.9%	1.09%
$DATA_4$	847 × 800	97.5%	1.7%	97.3%	2.2%
$DATA_5$	842 × 976	99.5%	2.1%	99.2%	0.2%

Table 2: Experimental results with five sample document images

Table 2 shows the testing results with 5 sample documents. In this table, E_{SRR} is the Successful Recognition Ratio, which is calculated as $E_{SRR} =$ (the number of identified characters)/(the number of characters in a document). E_{FRR} denotes the False Recognition Ratio for all characters in a document such as $E_{FRR} =$ (the number of image components that was identified as a character) / (the number of characters in a document). Equally, W_{SWG} is the Successful Word Grouping Ratio, and W_{FWG} is the False Grouped Word Ratio. Some graphic symbols were removed. All negative texts were returned to a positive form.

5 CONCLUSION

In this paper, we proposed a new text extraction algorithm from a text/graphic mixed document image and a grouping technique with a 3D neighborhood graph modeling scheme. The basic idea of separating texts from graphic symbols is to exploit the characteristics in the number of runs in the run-length encoded file of a given document. If the variance of runs encoded for a component is high in vertical and horizontal directions, then it is believed to be a symbol rather than a text, since text has smaller variance than symbols. The extracted character in the previous phase is mapped to a location in 3-dimensional space. We control the height of each mapped letter, according to the size of its enclosing circle, so characters with similar size will be placed in the nearly same X-Y plane in 3D space. In order to determine whether some characters are placed in a line or a curve, we need to compute the linearity of these letters. Experiments show that our algorithm can successfully extract text strings from a quite complicated document image with greater than 97% accuracy.

In the future, we will adopt an automatic character recognition techniques to extended this system for fully-automated document processing system. The extended system will be used for digital library to process papers, magazines, and engineering drawings.

REFERENCES

[FLE88] L.A. Fletcher and R. Kasturi: *A Robust Algorithm for Text String Separation from Mixed Text/Graphics Images*, IEEE Transaction On PAMI, Vol. 10, No. 6, pp. 910–918, Nov., 1988.

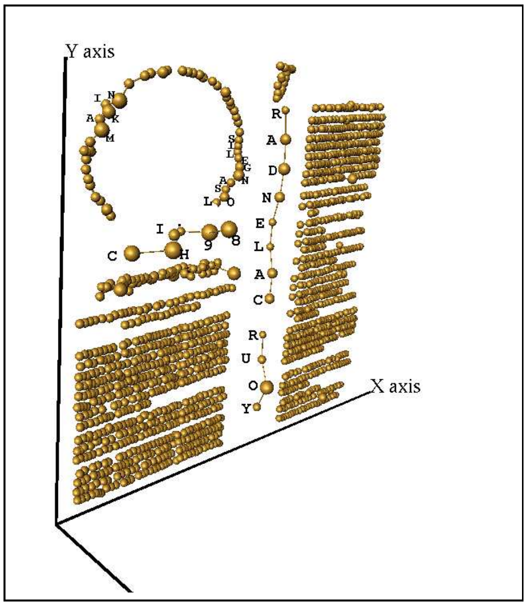


Figure 12: 3D Neighborhood Graph corresponding to Fig.11

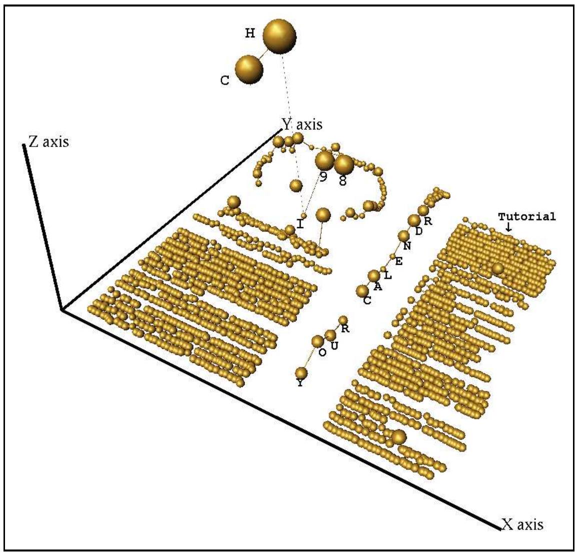


Figure 13: 3D graph of Fig.12 which is viewed from z-axis

[NAK93] A. Nakamura and O. Shiku: *A Method for Recognizing Character Strings from Maps Using Linguistic Knowledge*, Proc. of 2nd International Conf. on Document Analysis and Recognition, pp. 561–564, Oct., 1993.

[KAS90] R. Kasturi and S. Bow: *A System for Interpretation of Line Drawings*, IEEE Transaction On PAMI, Vol. 12, No. 10, pp. 978–991, Oct., 1990.

[EIK95] L. Eikvil and K. Aas: *Tools for Interactive Map Conversion and Vectorization*, Proc. Inf.conf.on Document Analysis and Recognition, Vol. 2, Aug., pp. 14–16, 1995.

[BUR95] M. Burge and G. Monagan: *Using the Voronoi tessllation for grouping words and multi-part symbols in documents*, Proc. of VISION GEOMETRY IV, SPIE's International Symposium on Optics, Imaging and Instrumentation, Vol. 2573, July, pp. 116–123, 1995.

[KAM93] M. Kamel and A. Zhao: *Extraction of Binary Character/Graphics Images from Grayscale Document Images*, Graphical Models and Image Processing, Vol. 55, No. 3, pp. 203–217, May, 1993.

[TAN97] C.L.Tan and P.O.Ng: *Text Extraction Using Pyramid*, Proc. of Pattern Recognition '97, Vol. 31, No. 1, pp. 63–72, Feb., 1997.

[AJA92] A.Jain and S. Bhattacharjee: *Text Segmentation using Gabor filters for automatic document pro*, Mach. Vision Applic, Vol. 5, pp. 169–184, 1992.

[LIA96] J.Liang and R.M.Haralick and I.T.Phillips: *Document Layout Structure Extraction Using Bounding Boxes of Different Entities*, Proc. of IEEE '96, pp. 278–283, 1996.

[DOW98] A.C.Downton and C.G.Leedham: *Pre-processing of envelope images for optical character recognition*, 9th Int. Conf. on Patter Recognition, pp. 27–31, 1998.

[SAI82] H.Saiga and Y.Nakamura and Y.Kitamura and T.Morita: *An OCR system for business cards*, 2nd Int. Conf. on Document Analysis and Recognition, pp. 642–656, 1982.