IJDAR (2001) 4: 115–130

International Journal on Document Analysis and Recognition

A word extraction algorithm for machine-printed documents using a 3D neighborhood graph model

Hwan-Chul Park[1], **Se-Young Ok**[2], **Young-Jung Yu**[3], **Hwan-Gue Cho**[3]

[1] R&D Center, PAXVR, Seocho Jeil B/D, 1624-2, Seocho-Dong, Seocho-Ku, Seoul 137-878, Korea
[2] LG Innotek, Yongin-shi, Kyunggi-do, Korea
[3] Graphics Application Lab., Department of Computer Science, Pusan National University, Kum-Jung-Ku, Pusan 609-735, Korea

Received August 3, 2001 / Accepted August 8, 2001

Abstract. Automatic character recognition and image understanding of a given paper document are the main objectives of the computer vision field. For these problems, a basic step is to isolate characters and group words from these isolated characters. In this paper, we propose a new method for extracting characters from a mixed text/graphic machine-printed document and an algorithm for distinguishing words from the isolated characters. For extracting characters, we exploit several features (size, elongation, and density) of characters and propose a characteristic value for classification using the run-length frequency of the image component. In the context of word grouping, previous works have largely been concerned with words which are placed on a horizontal or vertical line. Our word grouping algorithm can group words which are on inclined lines, intersecting lines, and even curved lines. To do this, we introduce the 3D neighborhood graph model which is very useful and efficient for character classification and word grouping. In the 3D neighborhood graph model, each connected component of a text image segment is mapped onto 3D space according to the area of the bounding box and positional information from the document. We conducted tests with more than 20 English documents and more than ten oriental documents scanned from books, brochures, and magazines. Experimental results show that more than 95% of words are successfully extracted from general documents, even in very complicated oriental documents.

Key words: Document analysis – Text extraction – 3D Neighborhood graph – Word grouping

1 Introduction

Recently, the need for a document analysis system that automatically extracts and recognizes text parts from documents with a complex layout has arisen. Figure 1 shows the flow of a general document analysis system.

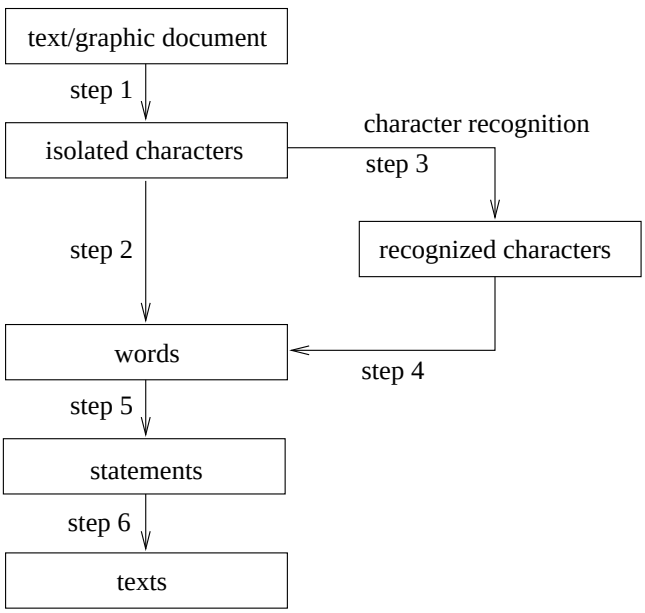


Fig. 1. The flow of a general document analysis system

In this paper, algorithms for character isolation (step 1) and word grouping (step 2) as shown in Fig. 1 are proposed.

There are many previous works that have proposed methods of automated text extraction from mixed text-graphic documents. Dori[2,3] explained a vector-based method for segmentation and recognition of dimensioning text from engineering drawings. In this work, the dimensioning text could be successfully extracted from engineering drawings. To do this, a vector-based segmentation method was used. However, the method is specialized to formatted documents such as engineering drawings. In our work, documents without format, such as brochures or magazines, are considered as input. Sobottka[11] proposed an approach to automatically extract text from colored books and journal covers. Fletcher[4] described a method that uses information from each connected component in a mixed text-graphic document. The Hough transformation was used

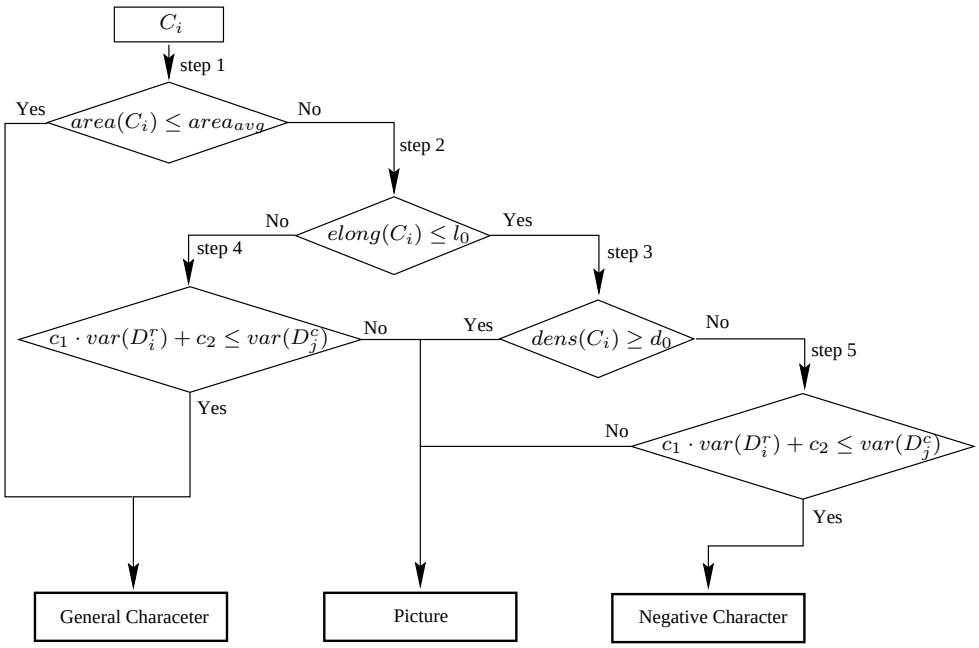


Fig. 2. The flow of a character classification algorithm

to group characters into words. For grouping characters into words, Burge[1] applied a new technique based on Voronoi tessellation. Jain and Bhattacharjee[6] considered text images as textured objects and used Gabor filtering, a well-known method for text analysis, but this method is sensitive to font sizes and styles, and generally it is a time-consuming process. Tan[12] proposed the pyramid model which efficiently and quickly identifies words or phrases placed in an image. However, accuracy is decreased when characters are required to be grouped into words. Kamel[7] proposed a method that separates text areas using grey-scale information in document images with mixed background images, shadows, and highlighted images. This technique uses a filtering method, an interpolation scheme, and adaptive thresholds. Liang[8] presented a technique for classifying and extracting layout structures from document pages. For this, they exploited the spatial configuration of the bounding boxes of different entities in a given image. Messelodi and Modena[9] proposed a method for the automatic localization of text embedded in complex images. They extract lines of text which have arbitrary orientations by using a hierarchical divisive procedure and external features (closeness, alignment, and comparable heights). However, it cannot be used to process lines of text on a curve or that intersect. Hideaki and Hirotomo[5] presented the ELSL (Extended Linear Segment Linking) algorithm to extract curved text lines from documents. However, documents with intersecting text lines were not considered.

In this paper, documents written in Asian and Western languages were considered. For these documents, an algorithm that automatically separates character and non-character parts and a new word grouping technique were proposed. To isolate characters, several character features (size, elongation, and density) were used with a run-length analysis method. The proposed word grouping algorithm deals with words that are arbitrarily placed on straight, curved, and intersecting lines. To do this, the 3D neighborhood graph model was developed. Two constraints, angle and distance, were used, which allows words to be extracted from curved or arbitrary straight lines. The 3D neighborhood graph model enables words to be extracted from intersecting text lines. This algorithm has been tested on English and Korean documents obtained from books, brochures, and magazines.

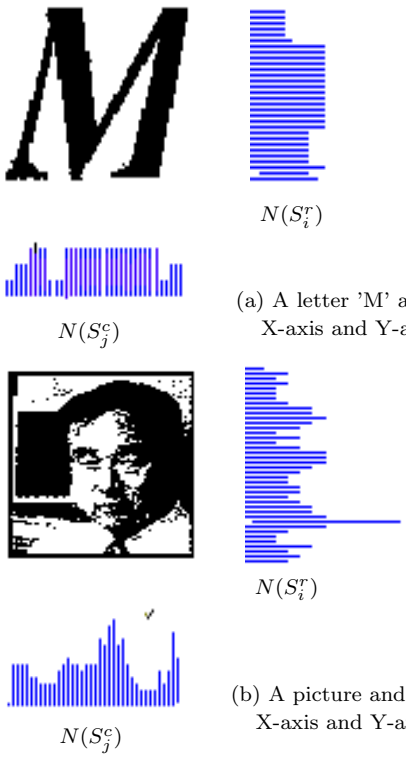
The rest of this paper is organized as follows. In the next section, an algorithm for extracting characters from a general mixed text-graphic image is presented. In Sect. 3, the word grouping algorithm, which uses the isolated characters acquired in the previous section as input, is described. The 3D neighborhood graph model and the angle and distance constraints used to extract words are also described. Section 4 shows some experimental results. In Sect. 5, the conclusion is presented.

2 Character extraction

In this part, a method for isolating characters from a mixed text-graphic document is described. A run-length encoding of the connected components is used to isolate the characters. The character grouping phase is processed after the isolation phase, since Asian letters consist of several disconnected strokes[10].

2.1 Run-length encoding

In order to isolate a character, first a set of connected components is found using an 8-connected region generating algorithm. After finding all connected components, background noise is identified by the number of pixels and removed. If the number of pixels in a component is less than a certain threshold value, it is regarded as salt



(a) A letter 'M' and corresponding X-axis and Y-axis run profile

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

Fig. 3. The variance of runs in a character and a picture

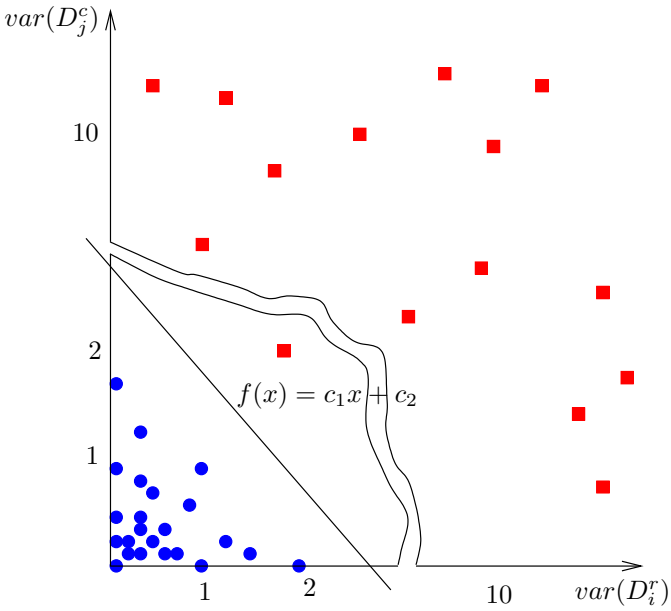


Fig. 4. The classification for characters and pictures: a circle denotes a character and a rectangle denotes a picture

and pepper noise. A threshold value of 6 pixels was determined through experimentation. Then all connected components are classified into three types: general characters, negative characters, and pictures.

In order to determine the various types of connected components, a classification technique was developed. Let C_i denote a connected component. Four metric functions were used to classify each connected component.

- $\texttt{box}(C_i) = \langle w_i, h_i \rangle$. The bounding box of a connected component C_i, where w_i is the width, and h_i is the height of the bounding box.
- $\texttt{area}(C_i) = w_i \cdot h_i$. The area of $\texttt{box}(C_i)$.
- $\texttt{dens}(C_i) = \texttt{pixel}(C_i)$ / $\texttt{area}(C_i)$. The density of a component in terms of the bounding box area and $\texttt{pixel}(C_i)$, which is the number of pixels in a component C_i.
- $\texttt{elong}(C_i) = \min\{w_i, h_i\}/\max\{w_i, h_i\}$. The elongation of a component C_i.

The classification procedure consisted of five steps, shown in Fig. 2. Let $area_{avg}$ denote the average area of bounding boxes for all connected components. In step 1, connected components are divided into two classes, small components and large ones, according to $\texttt{area}(C_i)$. Each connected component is considered a general character if $\texttt{area}(C_i)$ is smaller than $area_{avg}$. Otherwise, it is considered to be a large component.

Large components are divided into two classes according to their elongation value, $\texttt{elong}(C_i)$. In step 2, the elongation threshold was set at $l_0 = 0.8$ after many tests were conducted on real documents.

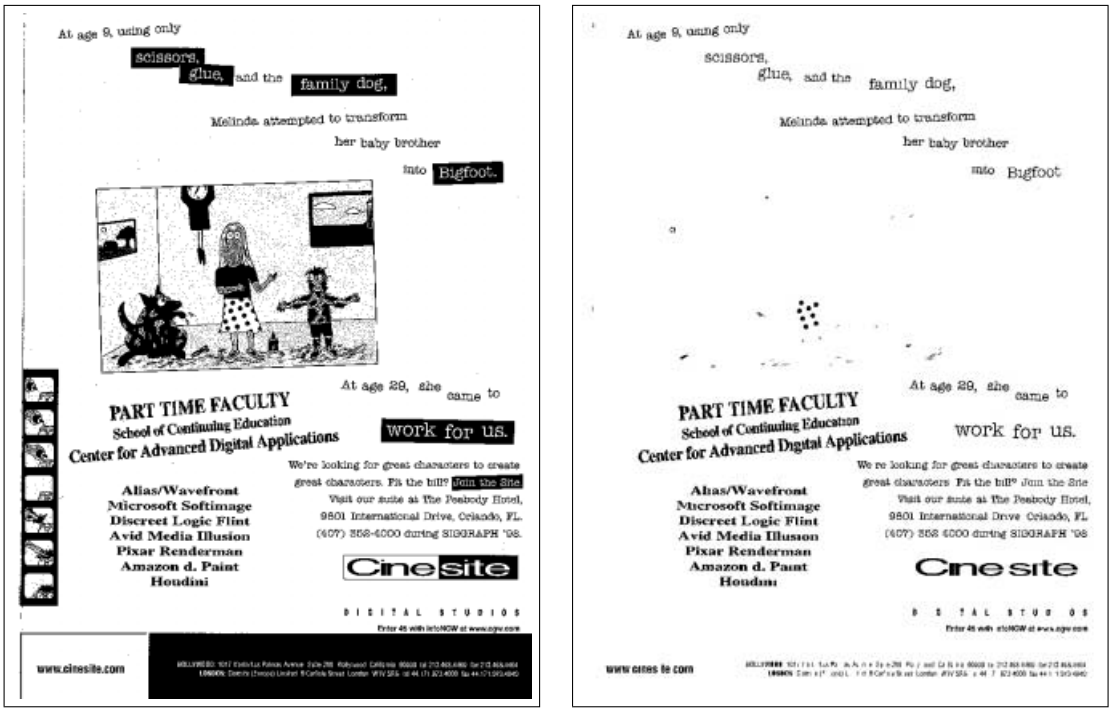
The algorithm assumes that negative characters with solid backgrounds or pictures might be longer components. Experiments have shown that the density, $\texttt{dens}(C_i)$, of a picture is larger than that of a negative character, so $\texttt{dens}(C_i)$ is used to refine these types of longer components in step 3. If $\texttt{dens}(C_i)$ of a longer component is larger than the density threshold, d_0, then it is considered to be a picture. Otherwise, step 5 is executed to derive the variance of run-length encoding, which is the crucial part of our algorithm. Through several experiments, the density threshold, d_0, was set to 0.7 so that pictures could be isolated.

Each component should now be classified as either a general character, negative character or picture. In steps 4 and 5, each component is classified by examining a characteristic value (variance) of a component. Some measuring notations for a component C_k are given. Let $S_i^r(S_j^c)$ denote a binary sequence for ith row(jth column) in C_k and $N(S_i^r)(N(S_j^c))$ denote the number of alternating runs of $S_i^r(S_j^c)$. For example, if S_i^r is "111011 01011", $N(S_i^r)$ is 7. Similarly, if S_j^c is "11100001111", $N(S_j^c)$ is 3. Let D_i^r and D_j^c denote the difference in the alternating numbers of run for each direction, i.e., $D_i^r = |N(S_{i+1}^r) - N(S_i^r)|$ and $D_j^c = |N(S_{j+1}^c) - N(S_j^c)|$. Figure 3 shows horizontally and vertically projected profiles of a character and a picture. $N(S_i^r)(N(S_j^c))$ of Fig. 3a is a regular form, but that of Fig. 3b is irregular. Then it is reasonable to assume that the variance of $D_i^r(D_j^c)$, $var(D_i^r)(var(D_j^c))$, of a picture might be larger than that of a character.

In the following, we give the notation of the variance for the component C_k.

$$var(D_i^r) = \frac{\sum_{a=1}^{h_k-1}(D_a^r - D_{avg}^r)^2}{h_k - 1}$$

$$var(D_j^c) = \frac{\sum_{b=1}^{w_k-1}(D_b^c - D_{avg}^c)^2}{w_k - 1}$$



a Original document **b** The result of character extraction from **a**

Fig. 5. Original document **a** and the result of character extraction **b**

$$D^r_{avg} = \frac{\sum_{a=1}^{h_k-1} |N(S^r_{a+1}) - N(S^r_a)|}{h_k - 1}$$

$$D^c_{avg} = \frac{\sum_{b=1}^{w_k-1} |N(S^c_{b+1}) - N(S^c_b)|}{w_k - 1}$$

In the next part, the method of classifying each component using a linear classifier is described.

2.2 A linear classifier for characters and pictures

The basic parameter used to distinguish between characters and pictures is the variance of D^r_i and D^c_j, $var(D^r_i)$ and $var(D^c_j)$. The linear classifier distinguishes between characters and pictures by comparing the variance of each component. Figure 4 shows a distribution graph for the variances, $var(D^r_i)$ and $var(D^c_j)$, of each connected component. In Fig. 4, each solid circle denotes the variance coordinate($var(D^r_i)$,$var(D^c_j)$) of a character, and each solid rectangle denotes that of a picture.

In the preliminary experiments, it was observed that the variance coordinate of each character was concentrated at the origin point, while that of each picture was placed away from the origin point. Thus, using this interesting property, connected components were classified into two types (characters and pictures) using a straight line, a linear classifier $f(x)$. The linear classifier, $f(x)$, in Fig. 4 determines the type of a connected component by carefully choosing the constants c_1 and c_2.

In steps 4 and 5, if $c_1 \cdot var(D^r_i) + c_2 \leq var(D^c_j)$, the component is classified as a character. Otherwise, it is classified as a picture. The optimal values for c_1 and c_2 were determined to be -1.09 and 2.8 through experiments on real documents. In this paper, the linear classifier coefficients are determined manually, rather than automatically. Therefore, if the resolution of experimental documents is changed, the parameters should be recalculated. Further study is needed to develop a method to automatically determine the value of the parameters according to the change of input.

Figure 5a is a mixed text-graphic document and, Fig. 5b shows the result of our algorithm for character extraction. In Fig. 5a, the phrases "work for us", "Bigfoot", "site" and "HOLLYWOOD..." were successfully recognized as negative characters by our algorithm. If negative characters are recognized, then they are easily converted into regular characters. Note that the complicated pictures in Fig. 5a were successfully removed by the linear classifier $f(x)$.

3 Word grouping by 3D graph model

Previous methods for word extraction have mainly been concentrated on extracting words from a horizontal or vertical text line. Messelodi[9] considered external features (closeness, alignment, and comparable height) between characters for text line selection, so they could extract characters of the word only on a straight alignment. However, some characters could be located on a curve (especially in commercial brochures). Hideaki and Hirotomo[5] extracted curved text lines. However, they did not consider word grouping and intersected text lines. The main focus of this section is how to deal with this problem.

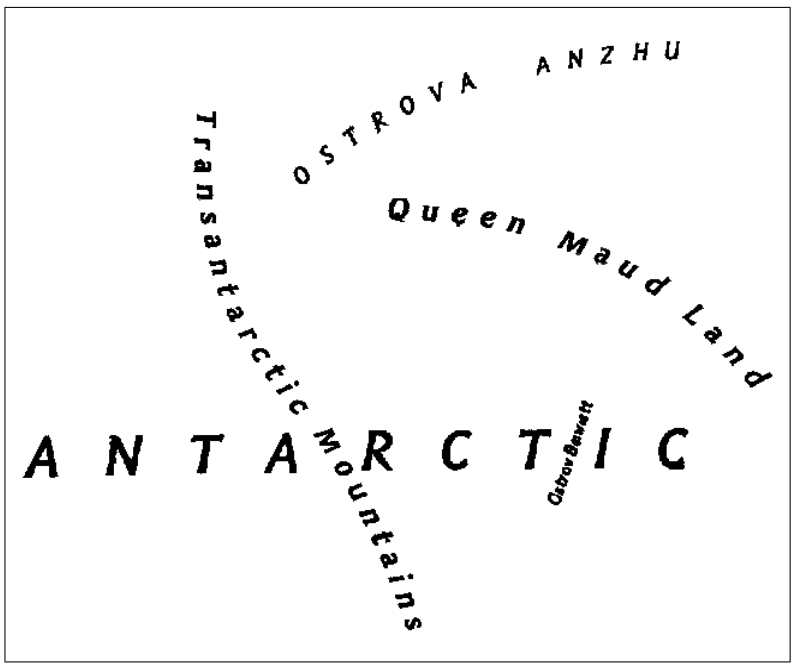


Fig. 6. An example image with intersecting words and characters that are different in sizes, fonts, and orientations

A new method to extract words from characters on a general curve and/or intersecting lines as well as on a straight line with an arbitrary direction is proposed. For isolating intersecting words (see Fig. 6), the 3D neighborhood graph model easily separates each word. The angle and distance constraints are used to trace words on a curve.

3.1 3D neighborhood graph

In general, words in a document are placed on a horizontal or vertical text line. In these cases, the distance between characters within a word is less than the distance between characters in different words. Extracting words from a document image by investigating the distance between adjacent characters is easily done. There are some cases where word orientation is arbitrary or words intersect. When two words intersect, it is difficult to automatically extract a word, since the distance between each character is not uniform. Figure 6 shows a document image (text layer of a map) with various sizes and arbitrary character orientation. In Fig. 6, we can see that the phrase "Transantarctic Mountains" intersects the larger word "ANTARCTIC". Some words(e.g., Queen Maud Land) are on an arc. In Fig. 6, the two adjacent characters 'A' and 'R' must be identified as adjacent letters of "ANTARCTIC". However, since the nearest character of 'R' is 'o' of the word "Mountains," the previous technique cannot be applied.

Here, a special technique is used to separate words from such intersections. The technique relies on the basic assumption that the size of the bounding boxes of adjacent characters is quite similar. When two words intersect each other, characters in each intersecting word have different sizes, especially in a geographical map. Therefore, two important parameters in separating intersecting words are the font size and word orientation

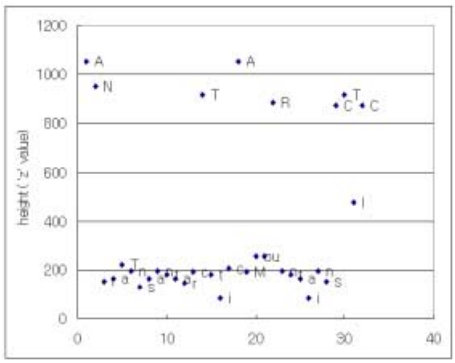


Fig. 7. The graph of the height value for each character of the intersecting words "ANTARCTIC" and "Transantarctic Mountains" in Fig. 6

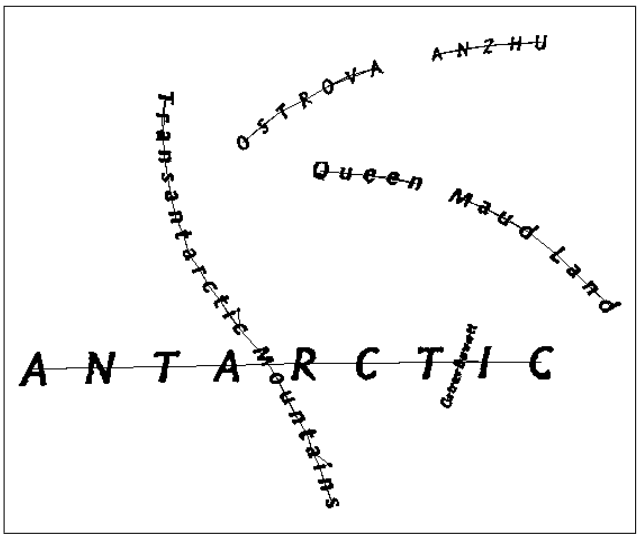
of each character. In order to apply this technique, the 3D neighborhood graph model, which maps each character in 2D space into 3D space according to its bounding box size, is used.

The character isolation process has already been described in Sect. 2; thus, the 2D placement of each character $C_i(x_i, y_i)$ and the location (x_i, y_i) of the center point of the bounding box, $\texttt{box}(C_i)$, are already known. Now the 3D neighborhood graph, $G_{3D}(V, E)$, is constructed from the set of $C_i(x_i, y_i)$. For $G_{3D}(V, E)$, first each character $C_i(x_i, y_i)$ is mapped to a vertex, v_i, of $G_{3D}(V, E)$. Then edges are created by considering the distance between two mapped characters. A vertex, v_i, has the 3D coordinate $v_i(x_i, y_i, z_i)$. The center point (x_i, y_i) of C_i is assigned as two of the 3D coordinates of v_i. The height value, z_i, is determined by the $\texttt{area}(C_i)$, and thus is mapped $v_i(x_i, y_i, z_i)$ from C_i.

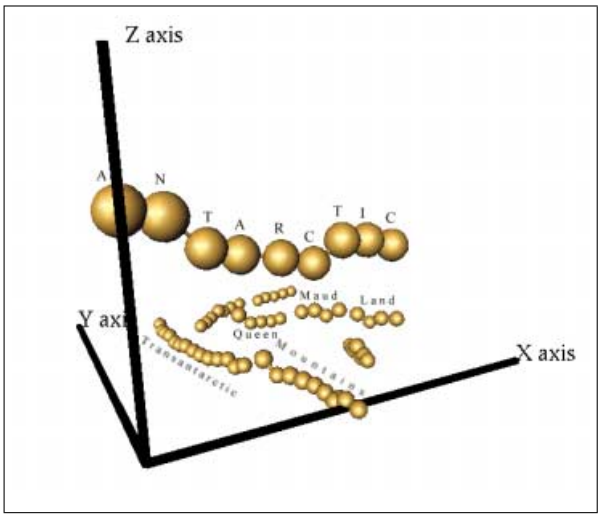
Table 1 shows the z values ($\texttt{area}(C_i)$ of each bounding box) for each character of the intersecting words "ANTARCTIC" and "Transantarctic Mountains" of

Table 1. area(C_i) values for each character of the intersecting words "ANTARCTIC" and "Transantarctic Mountains" in Fig. 6

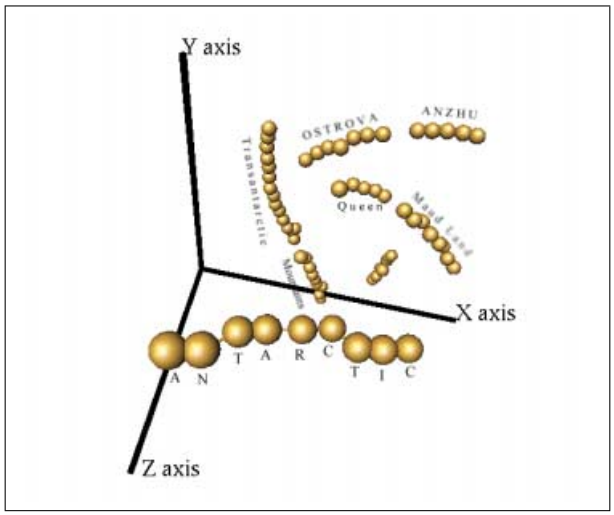
Text	A	N	T	A	R	C	T	I	C	T	r	a	n	s	a	n
$S(c_i)$	1054	952	918	1054	884	875	918	476	875	224	154	165	196	130	165	196
Text	t	a	r	c	t	i	c	M	o	u	n	t	a	i	n	s
$S(c_i)$	180	165	154	192	180	84	192	256	192	195	196	180	165	84	196	130



a $G_{3D}(V, E)$ of Fig. 6, each character corresponds to each vertex and each edge denotes the edge set



b 3D view of **a**



c Another view of **a**

Fig. 8. **a** $G_{3D}(V, E)$ of Fig. 6 and **b**, **c** the corresponding 3D view. Bigger nodes are placed on the upper z-axis

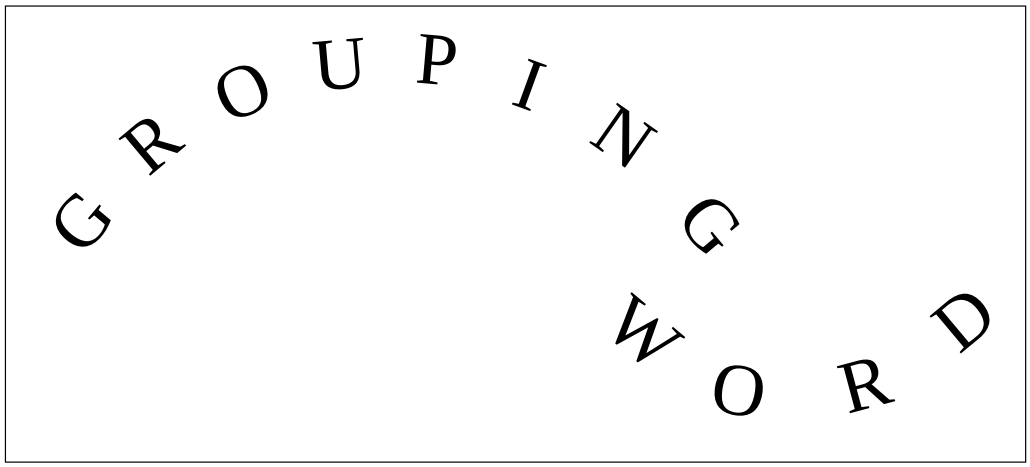
Fig. 6. Figure 7 shows the graph which corresponds to Table 1. In Fig. 7, extracted characters are located on a different plane in the 3D z-axis according to the area(C_i) of each component. When a character is a larger size, it will be placed in a higher position on the z-axis. By this mapping policy, characters with similar bounding box sizes will have similar z values, which means that they are located on a flat z plane.

After computing z_i for each $v_i(x_i, y_i, z_i)$, we need to generate the edge set, E, of the $G_{3D}(V, E)$. Let (v_i, v_j) denote an edge connecting v_i and v_j as in the following.

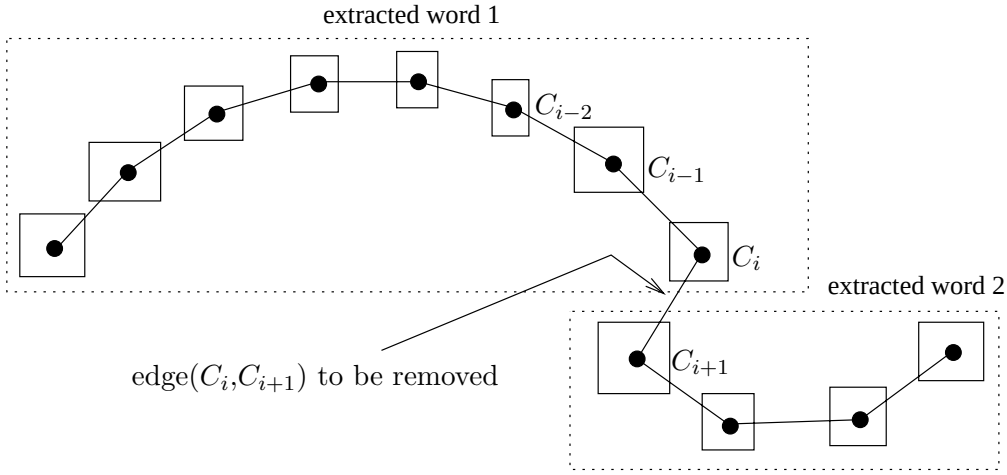
$$E = \left\{ (v_i, v_j) \mid d_\alpha(v_i, v_j) = \min_{p,q}\{d_\alpha(v_p, v_q)\},\ p \neq q \right\} \quad (1)$$

$$d_\alpha(v_p, v_q) = \alpha \cdot d2(C_p, C_q) + (1 - \alpha) \cdot d3(v_p, v_q),$$

where $d2(C_p, C_q)$ denotes a 2D Euclidean distance between C_p and C_q, $d3(v_p, v_q)$ denotes a 3D Euclidean distance between v_p and v_q, and α is a control constant. If the value of α approaches 1, the 2D distance is preferred in order to measure the edge distance. The optimal value of α depends on the type of document. In addition, we have not considered automatic determination regarding



a Two words "GROUPING" and "WORD"



b Word separation within a connected component

Fig. 9. Word grouping process using linearity constraint: two words, "GROUPING" and "WORD," are divided by linearity constraint

the optimal value of α for each document style in this paper. For the purposes of this paper, α has been assigned the value of 0.8, which was determined after several experiments. When $\alpha = 0.8$, the algorithm gives the best average performance. Through this process, many components are connected and $G_{3D}(V, E)$ is constructed.

Figure 8 shows the resulting image after the construction of $G_{3D}(V, E)$ from characters obtained in Fig. 6 and the corresponding 3D graph. Figure 8 also shows seven components connected by solid lines. The phrase "Transantarctic Mountains" and the word "ANTARCTIC" are successfully separated into two different components. However, words "Maud Land" and "Transantarctic Mountains" are not extracted. In the next section, the method used to extract words from each component in $G_{3D}(V, E)$ is described.

3.2 Word grouping with $G_{3D}(V, E)$

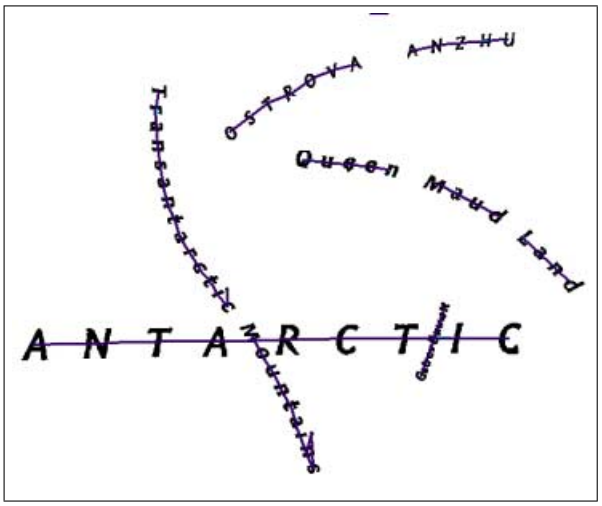
Each component in $G_{3D}(V, E)$ may contain more than one word. In this part, the method used to divide each component of $G_{3D}(V, E)$ into individual words is presented. This word grouping method has two steps. In the first step, the linearity constraint between edges is used to divide each component into candidate parts for words. Then the distance between characters becomes the main characteristic parameter in word extraction.

At first, each component is divided into several candidate parts by investigating the angle between two adjacent edges. We assume that centroids of characters in a word are nearly collinear, since it is common that characters in a word are placed linearly and uniformly in a document. Thus, words are located by computing the degree of linearity of adjacent edges. Let θ_i be the angle between two successive edges (C_{i-1}, C_i) and (C_i, C_{i+1}), i.e., $\theta_i = \angle C_{i-1}C_iC_{i+1}$. We find the maximum index, k, satisfying the following linearity constraint.

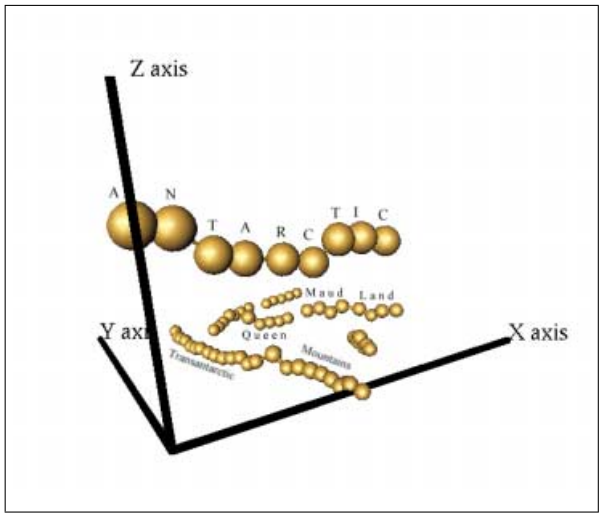
$$\theta^* = \max_k \left\{ \frac{1}{k} \sum_{i=1}^{k} (\theta_{avg} - \theta_i)^2 \right\} \leq \theta_0 \tag{2}$$

$$\theta_{avg} = \frac{1}{k} \sum_{j=1}^{k} \theta_j,$$

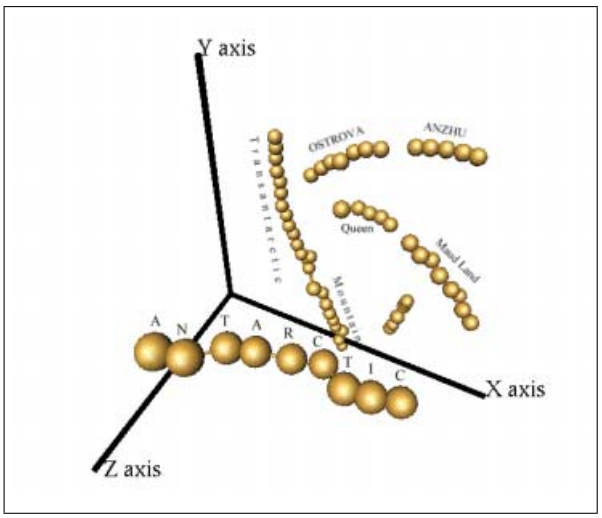
where θ_0 denotes the threshold value for linearity. If the value of θ_0 approaches 0°, only words which are placed on



a The result of word grouping after applying Eq. (2) and (3) to Fig. 8



b 3D view of **a**



c Another view of **a**

Fig. 10. a The result after applying two constraints(linearity and distance) and **b**, **c** the corresponding 3D view: the two phrases "Transantarctic Mountains" and "Maud Land" were successfully divided into four words by the two constraints(linearity and distance)

a nearly straight line are extracted. Otherwise, a word on a general curve is extracted. The optimal value of θ_0 depends on the type of document. It is difficult to determine automatically the optimal value of θ_0 for all kinds of documents. In this paper, θ_0 was assigned the value of 26° only after several experiments.

Figure 9 shows an example of word grouping using linearity constraint. Two words in Fig. 9a are selected as word candidates by 3D graph model. In Fig. 9b, the solid rectangle denotes the bounding box of a character, the solid circle denotes the center of the bounding box, the solid line denotes the edge, which is acquired from Eq. (1), and the dotted rectangle denotes a word that was extracted using the linearity constraint. Note that the linearity constraint is satisfied up to the angle $\theta_{i-1} = \angle C_{i-2}C_{i-1}C_i$. However, because the linearity constraint is not satisfied when $\theta_i = \angle C_{i-1}C_iC_{i+1}$, the edge$(C_i, C_{i+1})$ was removed from the connected component. Thus two words are extracted from a connected component. This example shows that we could successfully separate words from a connected component using the linearity constraint.

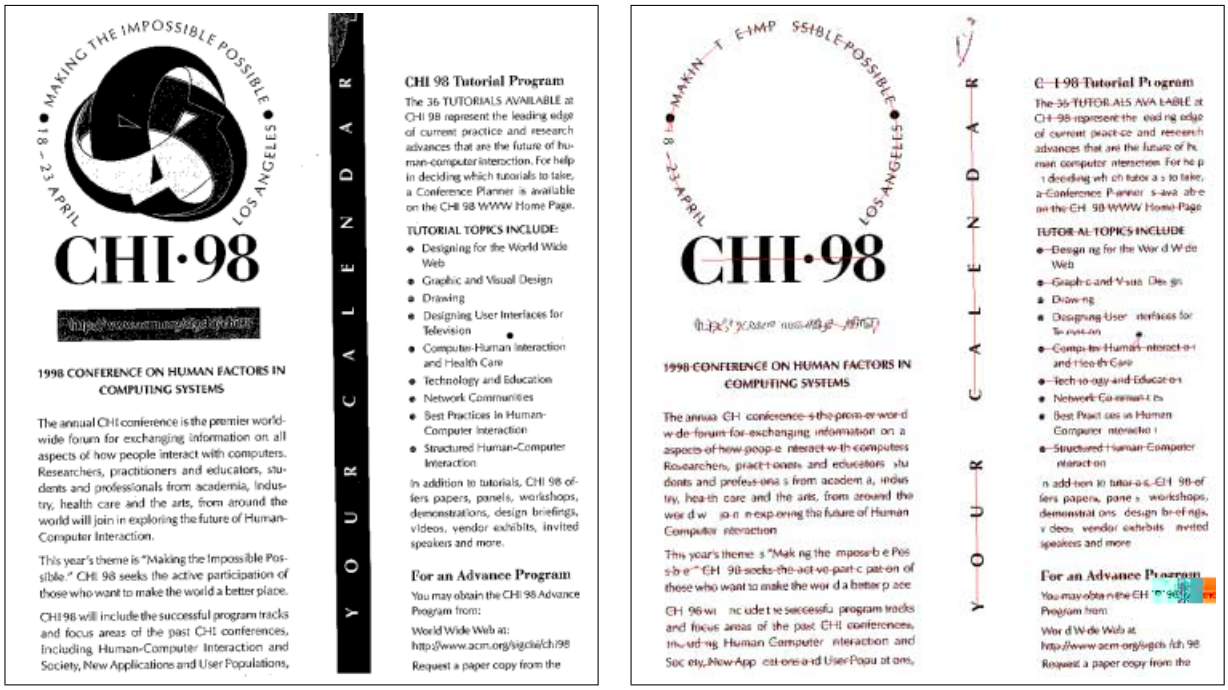
Another word extraction case should be considered. In Fig. 8, a component "Transantarctic Mountains" cannot be divided by applying only the linearity constraint (Eq. (2)), since the angle between the two words is nearly collinear. In this case, we consider the distance between characters to extract words. Usually the distance between words is greater than the distance between characters in a word. Thus, a character sequence which satisfies the following distance constraint (maximal regular sequence) is found:

$$l^* = \left(\max_{p,q} \{d2(C_p, C_q)\} - l_{avg} \right) \leq l_0 \qquad (3)$$

$$l_{avg} = \frac{1}{k-1} \sum_{i=1}^{k-1} d2(C_i, C_{i+1}),$$

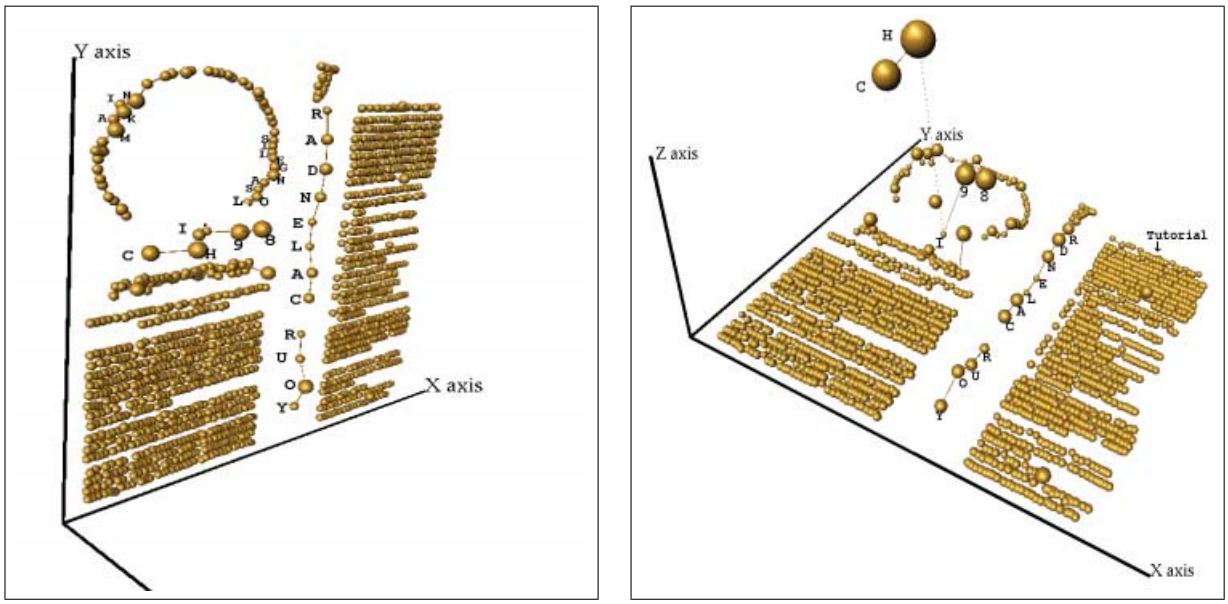
where k is the number of characters in a component and l_0 denotes the threshold value for distance.

Figure 10 shows the resulting image and the 3D view after applying Eqs. (2) and (3). In Fig. 10, the two components "Maud Land" and "Transantarctic Mountains"



a An original document image **b** Grouped words from **a**: words are marked by dotted lines

Fig. 11. An original document image **a** and the word grouping result **b**



a The corresponding 3D view of Fig. 11b **b** Another 3D view

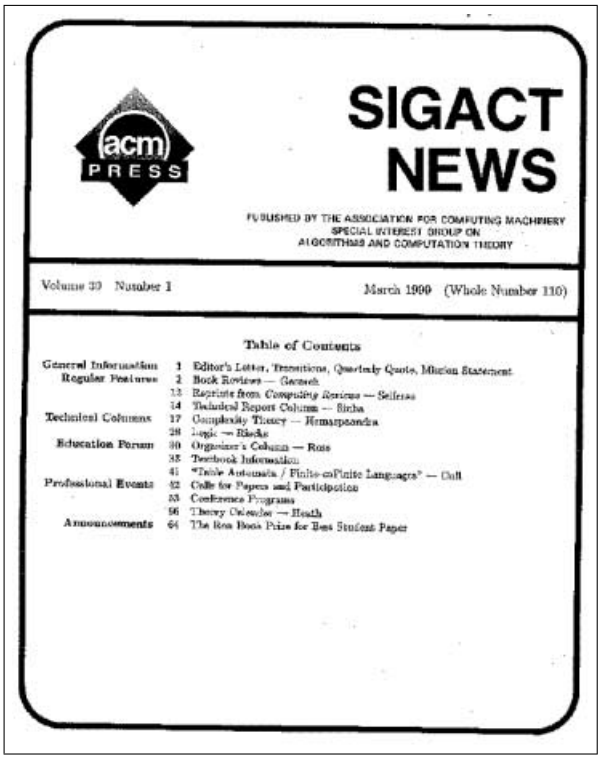
Fig. 12. Two 3D views corresponding Fig. 11b

are successfully classified into four words by the two constraints, linearity and distance.

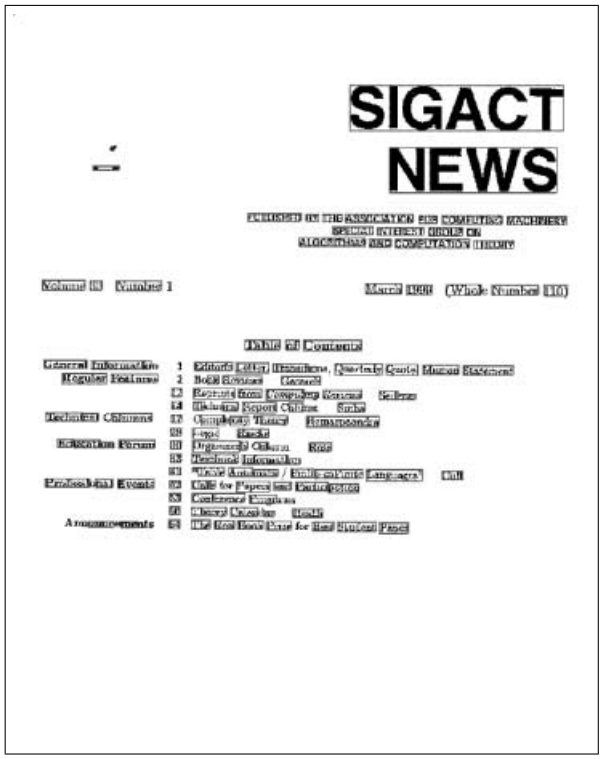
4 Experiments

The algorithm described in this paper has been tested on more than 20 machine printed documents (containing Roman characters and Korean letters) that were obtained from books, brochures, and magazines. Each document image has more than two hundred characters with various fonts, sizes, and orientations. Each document is scanned as 512×512 resolution and black and white image. We have measured the performance of our code on a PentiumIII 600/MHz computer. The processing time for each document is about 15/s.

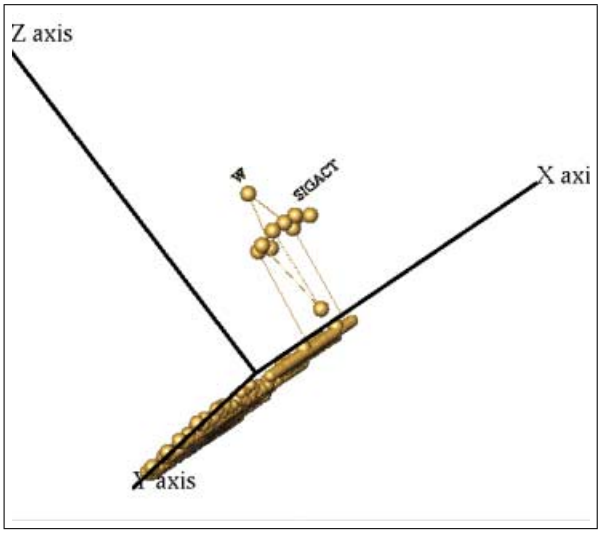
Figure 11a shows a typical document image (Call for Paper) and Fig. 11b is the final result of our algorithm. The negative characters (vertical center) and pictures are successfully processed. After the character extraction procedure, pictures are removed and negative char-



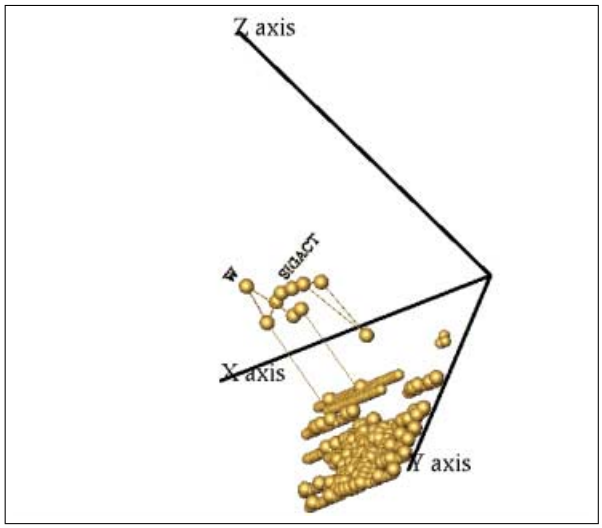
a An original document image



b The result of word grouping



c 3D neighborhood graph



d 3D neighborhood graph with different viewpoint

Fig. 13. The result of word grouping and 3D neighborhood graph

acters are recovered completely. It is worthwhile to note that characters (LOS...) on a circle were successfully extracted. Each isolated word is threaded with a dotted line in Fig. 11b. The vertical phrase "YOUR CALENDAR" was successfully reversed and isolated into two words: "YOUR" and "CALENDAR."

The perspective view of $G_{3D}(V, E)$ of Fig. 11b is shown in Fig. 12a, which is seen from top to bottom. Figure 12b is another perspective view from a different direction. Figure 12 shows that large characters (denoted as a big sphere (CHI 98)) are placed on a higher z-axis. Unfortunately, the embossed characters "http://..." were not processed, since the algorithm does not consider double negative characters.

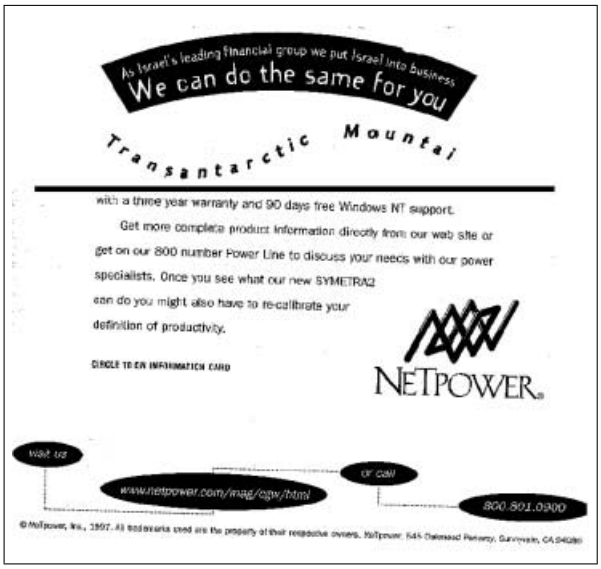
Experimental results with five typical English documents are shown in Figs.13, 14, and 15. Figure 13 shows a an original document, b grouped words, and c and d 3D neighborhood graphs from different viewpoints. Two larger words, "SIGACT" and "NEWS", are placed in a higher position on the z-axis through the use of our 3D neighborhood graph model. Thus, we could easily separate them from the smaller characters that occupy a lower position. Figure 14 shows another document (commercial page) and the corresponding word grouping re-

Table 2. The result of the character extraction and word grouping with 20 English documents: SIZE is the pixel dimension, $R_{char}(R_{word})$ denotes the number of characters(words) in a document, $D_{char}(D_{word})$ denotes the number of characters(words) detected by the algorithm, and $F_{char}(F_{word})$ denotes the number of false characters(words)(detected as characters by the algorithm, but which are not characters in the original document). $E_{SRR}(= D_{char}/R_{char})$ is the ratio of successfully isolated characters and $E_{FRR}(= F_{char}/R_{char})$ is the ratio of falsely isolated characters. $W_{SRR}(= D_{word}/R_{word})$ is the ratio of successfully grouped words and $W_{FRR}(= F_{word}/R_{word})$ is the ratio of falsely grouped words

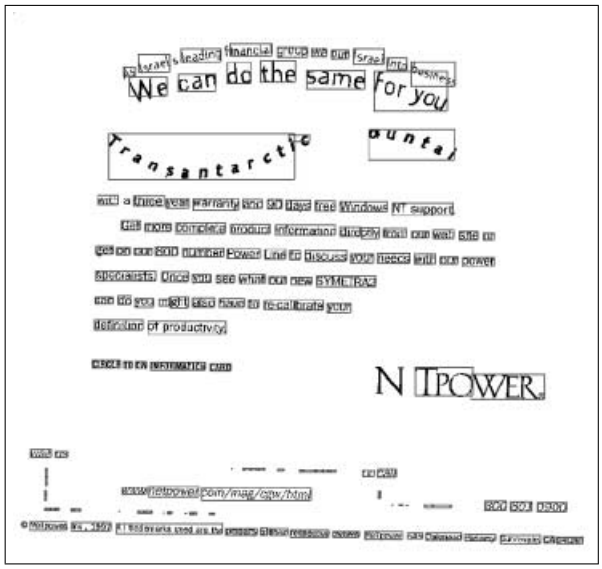
Data	Size	R_{char}	D_{char}	F_{char}	E_{SRR}	E_{FRR}	R_{word}	D_{word}	$Fword$	W_{SRR}	W_{FRR}
T1	747×992	554	549	8	99.1	1.4	130	119	2	91.5	3.4
T2	885×614	397	397	8	100.0	2.0	75	68	2	90.7	2.9
T3	842×976	864	856	13	99.1	1.5	260	252	2	96.9	0.8
T4	847×800	536	534	0	99.6	0.0	108	106	0	98.1	0.0
T5	800×1150	1094	1091	8	99.7	0.7	233	215	5	92.3	2.3
T6	798×554	587	554	2	94.4	0.3	116	115	1	99.1	0.9
T7	536×763	391	390	10	99.7	2.6	78	75	2	96.2	2.7
T8	1109×776	248	248	3	100.0	1.2	48	47	1	97.9	2.1
T9	541×755	665	660	3	99.2	0.5	142	134	2	94.4	1.5
T10	717×1134	192	191	2	99.5	1.0	68	65	1	95.6	1.5
T11	834×1251	834	831	9	99.6	1.1	390	375	6	96.2	1.6
T12	834×1220	579	570	5	98.4	0.9	219	204	2	93.2	1.0
T13	834×1196	426	402	3	94.4	0.7	179	170	3	95.0	1.8
T14	622×1000	255	251	4	98.4	1.6	76	71	2	93.4	2.8
T15	700×1000	629	625	5	99.4	0.8	96	92	2	95.8	2.2
T16	901×1283	1323	1322	13	99.9	1.0	246	241	4	98.0	1.7
T17	1100×778	586	586	20	100.0	3.4	118	117	6	99.2	5.1
T18	800×1045	237	235	0	99.2	0.0	42	40	2	95.2	5.0
T19	1083×153	1682	1669	3	99.2	0.2	316	309	5	97.8	1.6
T20	144×153	1461	1439	29	98.5	2.0	248	240	11	96.8	4.6
	TOTAL	13540	13400	148	99.0	1.1	188	3055	63	95.8	2.1

Table 3. The result of the character extraction and word grouping in the 10 Korean documents: SIZE is the pixel dimension, $R_{char}(R_{word})$ denotes the number of characters(words) in a document. $D_{char}(D_{word})$ denotes the number of characters(words) detected by the algorithm, and $F_{char}(F_{word})$ denotes the number of false characters(words)(detected as characters by the algorithm, but which were not characters in the original document). $E_{SRR}(= D_{char}/R_{char})$ is the ratio of successfully isolated characters and $E_{FRR}(= F_{char}/R_{char})$ is the ratio of falsely isolated characters. $W_{SRR}(= D_{word}/R_{word})$ is the ratio of successfully grouped words and $W_{FRR}(= F_{word}/R_{word})$ is the ratio of falsely grouped words

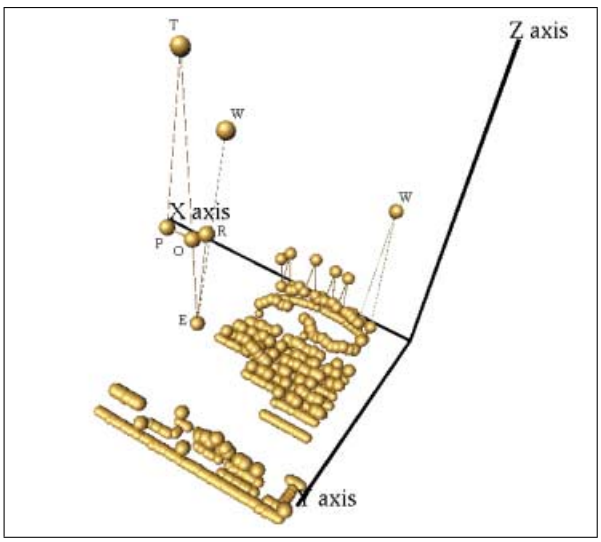
Data	Size	R_{char}	D_{char}	F_{char}	E_{SRR}	E_{FRR}	R_{word}	D_{word}	F_{word}	W_{SRR}	W_{FRR}
T21	800×1000	221	209	2	94.4	0.9	45	42	1	93.3	2.2
T22	760×988	110	109	2	99.1	1.8	40	38	1	95.0	2.5
T23	672×985	300	229	2	99.7	0.7	87	80	2	92.0	2.3
T24	700×962	454	447	4	98.5	0.9	136	130	2	95.6	1.5
T25	700×1116	170	160	3	94.1	1.8	57	52	1	91.2	1.8
T26	i700×1180	434	421	1	97.0	0.2	111	103	4	92.8	3.6
T27	893×1067	532	511	8	96.1	1.5	259	259	6	97.3	2.3
T28	842×1218	683	673	2	98.5	0.3	166	166	5	94.0	3.0
T29	842×1190	350	339	3	96.9	0.9	112	112	3	96.4	2.7
T30	847×1173	134	128	1	95.5	0.7	64	64	2	93.8	3.1
	TOTAL	3388	3296	28	97.3	0.8	1077	1077	27	94.8	2.5



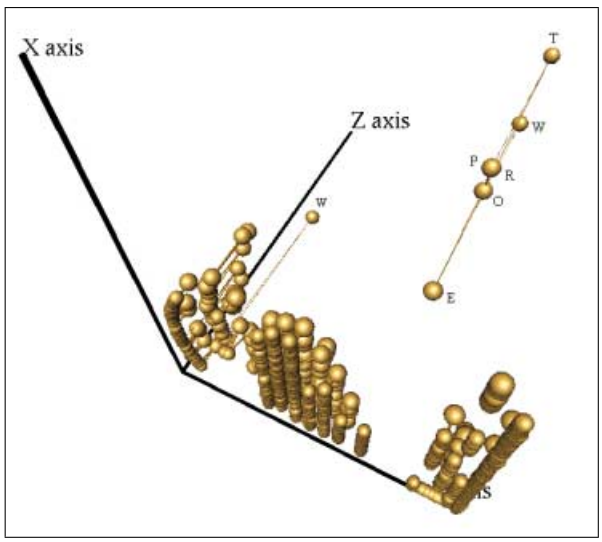
a An original document image



b The result of word grouping



c corresponding 3D view of **a**



d Another view with different viewpoint

Fig. 14. Another result of word grouping

sults. Figure 15 shows three complicated test documents. Many negative characters, pictures, and words on arcs are represented. All three documents were successfully processed.

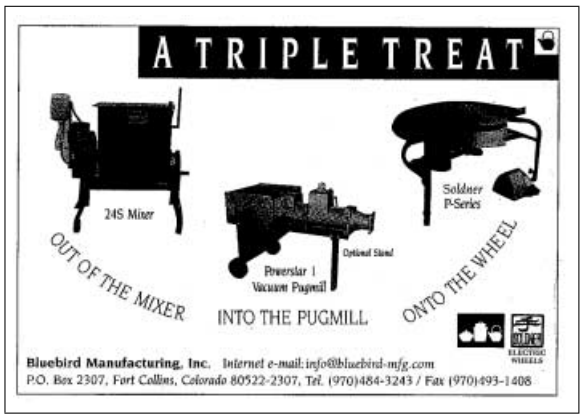
Word grouping in Korean documents (commercial page) is shown in Fig. 16. Figure 16a and c are original documents, which have lots of intersecting characters and characters on a circular form. Note that intersecting phrases in Fig. 16a are successfully grouped into words such as Fig. 16b. Figure 16c has some handwritten fonts in a diagonal direction in the upper part. Figure 16 shows that the algorithm successfully grouped the handwritten characters.

Now the quantitative results of this experiment with 20 English and ten Korean documents are given. In Tables 2 and 3, R_{char} (R_{word}) denotes the number of characters (words) in a document. D_{char} (D_{word}) denotes the number of characters (words) detected by the algorithm and F_{char} (F_{word}) denotes the number of false characters (words) (characters identified by the algorithm, but that are not characters in the original document). $E_{SRR}(= D_{char}/R_{char})$ is the ratio of successfully isolated characters and $E_{FRR}(= F_{char}/R_{char})$ is the ratio of falsely isolated characters (for example, a picture segment is recognized as a character). $W_{SRR}(= D_{word}/R_{word})$ is the ratio of successfully grouped words and $W_{FRR}(= F_{word}/R_{word})$ is the ratio of falsely grouped words (for example, a picture segment is included in a word). Our experiment shows that more than 95% of the words from a mixed text-graphic document were successfully grouped.

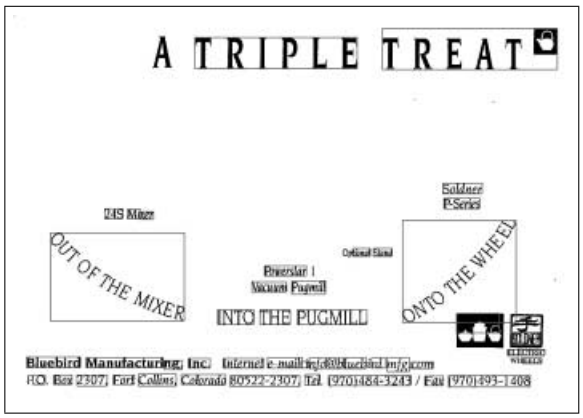
This experiment shows that it is easier to group characters into words in an English document than in an Asian document, since each character is one connected component, and the size of the character is more regular than that of Korean letters. Some pictures (at most

a

b



c



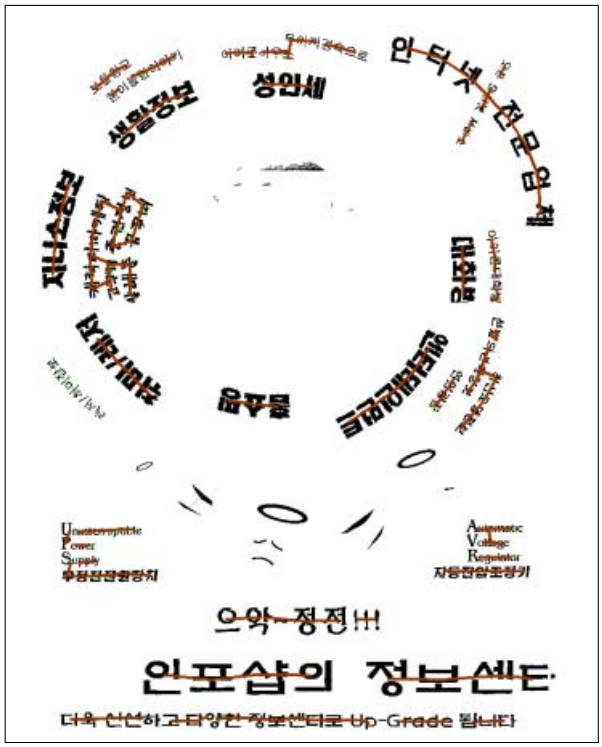
d

e

f

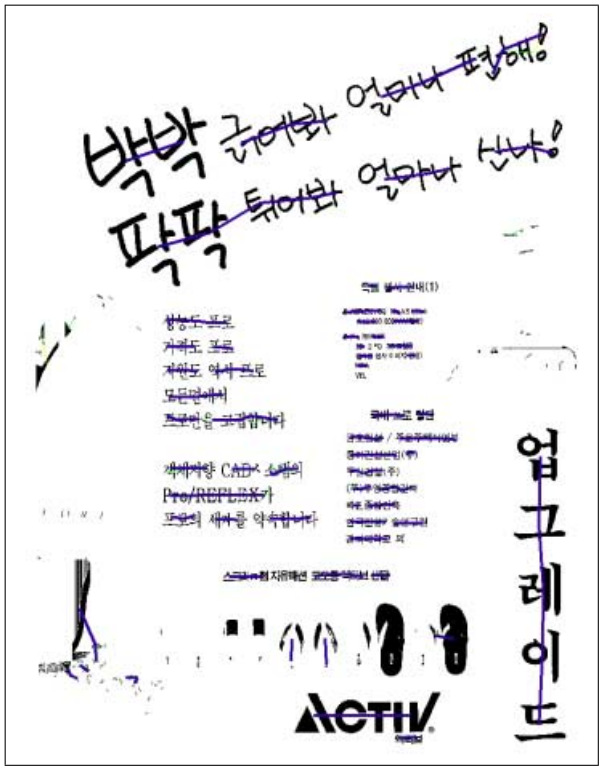
Fig. 15. Results of word grouping for English documents

a



b

c



d

Fig. 16. Two original Korean documents(a,c) and their corresponding word grouping results(b,d)

0.8%) could not be removed from the original documents (see Fig. 15), because they have similar run-length encoding with characters. Thus, some characters were not correctly grouped into words (see upper part in Fig. 15d).

5 Conclusion

In this paper, a new character extraction and word grouping algorithm for a mixed text-graphic document for Asian and Western languages is proposed. In the character extraction procedure, character and non-character parts were distinguished by investigating the variance of run-length of each connected component. The word grouping algorithm successfully extracted words, even if the words in a document were placed on a curve or intersected each other. This word grouping algorithm used a new 3D neighborhood graph model, which is very efficient and effective for word grouping. For these experiments, 20 English documents and ten Korean documents were taken from books, brochures, and magazines. Experiments have shown that the algorithm successfully extracts words in very complicated documents with more than 95% accuracy. The experiment results were quite satisfactory. In summary:

- A new character extraction algorithm was used to separate character and non-character parts from a mixed text-graphic document by investigating the variance of run-length of each connected component.
- The 3D neighborhood graph model was used to analyze the structural information of documents, which contain several fonts, sizes, orientations, and pictures.
- An angle and distance constraint equation was also used. By using this constraint, word components on a curve and/or an arbitrary straight line were identified.

As was shown in Fig.11, the system does not successfully deal with some decorative fonts (e.g., embossing effect). It seems possible to solve this problem by examining the general characteristics of decorative fonts. In the future, it is hoped that this system could be improved by adopting an automatic character recognition system to make it a truly fully-automatic document processing system.

References

1. M. Burge, G. Monagan: Using the Voronoi tessllation for grouping words and multi-part symbols in documents. Proc. Vision Geometry IV, SPIE's International Symposium on Optics, Imaging and Instrumentation (1995)
2. D. Dori, L. Wenyin: Vector-based segmentation of text connected to graphics in engeering drawing. Lecture Notes in Computer Science, vol. 1121. Springer, Berlin Heidelberg New York, 1996, pp. 322–331
3. D. Dori, Y. Velkovitch: Segmentation and recognition of dimensioning text in engineering drawings. Comput. Vision Image Understanding 69(2):196–201 (1998)
4. L.A. Fletcher, R. Kasturi: A robust algorithm for text string separation from mixed text/graphics images. IEEE Trans. PAMI 10(6):910–918 (1998)
5. H. Goto, H. Aso: Extracting curved text lines using local linearity of the text line. Int. J. Doc. Anal. Recognition pp. 111–119 (1999)
6. A. Jain, S. Bhattacharjee: Text segmentation using Gabor filters for automatic document processing. Mach. Vision Appl. 5:169–184 (1992)
7. M. Kamel, A. Zhao: Extraction of binary character/graphics images from grayscale document images. Graph. Models Image Process. 55(3):203–217 (1993)
8. J. Liang, R.M. Haralick, I.T. Phillips: Document layout structure extraction using bounding boxes of different entities. Proc. IEEE '96, pp. 278–283 (1996)
9. S. Messelodi, C.M. Modena: Automatic identification and skew estimation of text lines in real scene images. Pattern Recognition 32:791–810 (1999)
10. H.C. Park, S.Y. Ok, Y.J. Yu, H.G. Cho: Word extraction in text/graphic mixed image using 3-dimensional graph model. ICCPOL'99, pp. 171–176 (1999)
11. K. Sobottka, H. Kronenberg, T. Perroud, H. Bunke. Text extraction from colored book and journal covers. Int. J. Doc. Anal. Recognition 2:163–176 (2000)
12. C.L. Tan, P.O. Ng: Text extraction using pyramid. Proc. Pattern Recognition 31(1):63–72 (1997)

Hwan-Chul Park received his B.Sc degree in Computer Engineering from Dong-Seo University in 1996. He also received the M.Sc degree in computer science from Pusan National University in 2000. Since 2000, he has been with R&D center in the PAXVR. His research interests include virtual reality and interactive graphics.

Se-Young Ok received her B.Sc and M.Sc degrees in computer science from Pusan National University in 1997 and 1999, respectively. Since 1999, she has been working as an associate engineer in LG Innotek in Korea(ROK). Currently, she is researching in the area of multi-target multi-sensor image fusion such as multi-imagery-target detection, multi-imagery-target tracking. Her research interests include image processing and image-related works

Young-Jung Yu is a Ph.D. student at the Department of Computer Science at Pusan National University, Pusan, South Korea. He received his M.Sc and B.Sc degrees in computer science from Pusan National University in 1998 and 1996, respectively. His research interests include NPR and computer animation.

Hwan-Gue Cho received his B.Sc. degree in computer science and statistics from Seoul National University, Seoul, South Korea, in 1984. He also received the M.Sc. and Ph.D. degrees in computer science from KAIST (Korea Advanced Institute of Science and Technology), in 1986 and 1990, respectively. Since 1990, he has been with the Department of Computer Science, Pusan National University. In 1994, he was a visiting scholar at Max-Planck-Institute for Informatics in Saarbrücken. His main research topics are algorithm design and analysis, and computational geometry. Currently he works for flexible object animation and bioinformatics, especially phylogenetics.