Character Grouping Technique Using 3-D Neighborhood Graphs in Raster Map

Yong-Bin Kang
third floor of sin-ou building
Poydong, Kang-Nam-Ku, Seoul, Korea
ybkang@keystone.co.kr

Se-Young Ok, Hwan-Gue Cho
Graphics Application Lab
Department of Computer Science
Pusan National University
Kum-Jung-Ku, Pusan 609-735, Korea
seok,hgcho@pearl.cs.pusan.ac.kr

Abstract

The main problem in this paper is how to find the character which is placed on a line or curves of raster map. We give one novel algorithm to group each separated characters in a map. This word grouping is difficult especially in a map, since a map has lots of different types of character and each word has its own slanting line. For this, we propose the 3-D neighborhood graph G from a give set of characters. In this graph, each vertex of G represents the separated characters and it is placed in 3-D space according to the size of character. This makes the bigger characters will be located in the upper position, the smaller characters will be placed in the bottom. And we give an edge if two vertices are nearly placed in that 3-D space. By this edge connection strategy we can easily find the words of various different size in a map.

Keywords: Document analysis, Feature extraction, Recognition, word grouping

1 Introduction

The separation of text strings from a raster image is one of the most difficult and complex problems for digitizing the image automatically in a document analysis system. In general, this problem is solved manually and this work is error-prone and is an ineffective process. Therefore the need for an automated segmentation system is desirable.

There are many previous works of automated document analysis on the mixed text and graphics in engineering drawings and map digitizing fields. This technique is based on the filtering method, interpolation scheme and adaptive thresholds using the gray level of pixels. Eikvil proposed a method which used the elliptical Fourier descriptor based on the feature of symbols[3] For grouping a word from each separated character, Burge applied an area Voronoi tessellation[4].

1.1 Text Extraction In Image

In order to resolve problems which was described above paragraph, we propose an algorithm which uses a 3-D neighborhood graph to represent the semantically meaningful objects without any specific prior knowledges. Raster map images are obtained through scanning and binarization.

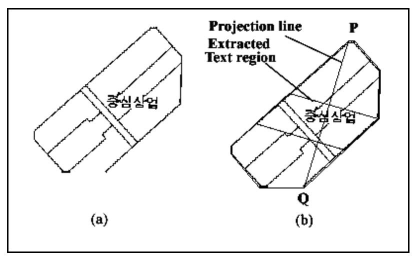


Figure 1. Extracting character string in map image. (a) image is original map and (b) shows text extracted image

Fig.1(a) shows an example of real raster map and one typical case that the previous character segmentation fails. Also the common line tracing algorithm would fail since texts are placed on the line. But our profiling algorithm can extract these overlapped characters with graphic al images. Fig.1(b) shows its extracted Korean character region, denoted as a rectangle, after running profiling algorithm.

2 Character Grouping

In this section, we present the method for grouping a word from the extracted characters. Since Korean, Japanese, Chinese letter consists of several disconnected strokes, this stroke grouping is required. However alphabet based Western language dose not have this difficulty except two characters such as i and j.

2.1 Multi-stroke Problem

Generally speaking, the distance between multi-strokes is smaller than the inter-character distance in most oriental languages. If the gap between multi-strokes is smaller that the inter-character gap, several single characters may be combined falsely into a single character. This problem can be avoided by applying our circle-based grouping technique. This procedure works in the following steps.

(1) Compute the centroid cp_i of every stroke s_i ;
(2) Compute the minimal enclosing circle C_i for each s_i.
(3) For all i, check if C_i intersects the image of $s_k, 1 \leq k \leq n$ or other enclosing circle C_j. Then we put s_a and s_b into the same stroke group for one character. We proceed this grouping step till there are no more intersecting C_i.

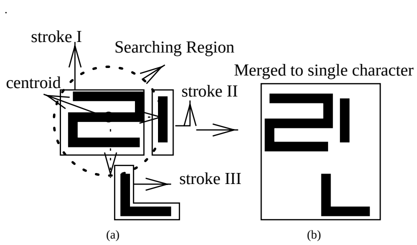


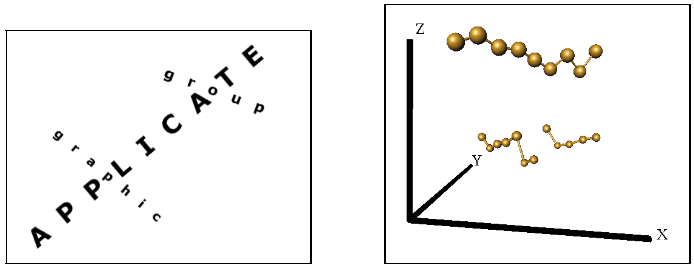
Figure 2. Multi-stroke grouping procedure. (a) shows one 3-stroke Korean character and (b) shows merged single character.

Fig.2 shows one Korean character which consists of three strokes. Dotted circles are the enclosing circle C_i. Fig.2(a), there are three strokes and stroke II and strike III intersects the enclosing circle of s_i, C_i. So we could group one character of three strokes as shown in Fig.2(b).

2.2 Word Grouping with separated letters

Now we describe a grouping algorithm to detect a word from singly identified characters in a map. By constructing the 3-D neighborhood graph, we can obtain more detailed geometrical information than 2-D representation of all character segments in a plane map.

As was illustrated in Fig.3(a), various sizes of characters are placed in a map. Some words are placed an curve rather than on a straight line. So the previous algorithm for word grouping used in a common document analysis should fails.



(a) character distributions (b) A 3-D graph of (a)after grouping

Figure 3. 3-D neighborhood graph(b) from (a).

Note the the word "graphic" cuts the bigger word "APPLICATE" in the middle in Fig.3(a).

The main idea under the 3-D neighborhood graph, $G_{3d}(V, E)$, is that we place each letter in the 3-dimensional space. Therefore we assign the coordinate of each character c_i as $c_i(x_i, y_i, z_i)$. Since x_i and y_i is the same as the location of c_i in the plane, we need to give the height value z_i for c_i. Briefly stating, if the size of c_i is big(small), then it should be placed in the higher(lower) position in (x, y, z) space. This placement strategy prevents from the interference between several sizes of letters. The detail z_i-height assigning rule is given below.

$$V = \{c_i \| whose\ position\ is\ \ (x_i, y_i, z_i)\}$$

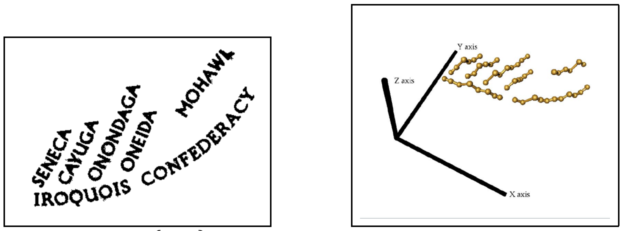
$$z_i = \frac{Area(c_i) \cdot \Gamma}{\Upsilon},\ where\ \Upsilon = \frac{1}{n}\sum_{k=1}^{n} Area(c_k),$$

$$\Gamma = 1/\left(\sqrt{\frac{1}{n}\sum_{i=1}^{n}(Area(c_i) - \frac{1}{n}\sum_{k=1}^{n} Area(c_k))^2}\right)$$

Now we need to define edge set E of $G_{3d}(V, E)$ with $\{ c_i(x_i, y_i, z_i) \}$ node set. For each $v \in V$ we give an edge u, if v is the nearest vertex from u in $G(V, E)$. Briefly saying we give an edge for every nearest neighborhood vertex of G. Therefore the degree of each node after this edge assigning should be 1 or 2. This graph guarantee that all pairs of relatively closest nodes are connected by an edge.

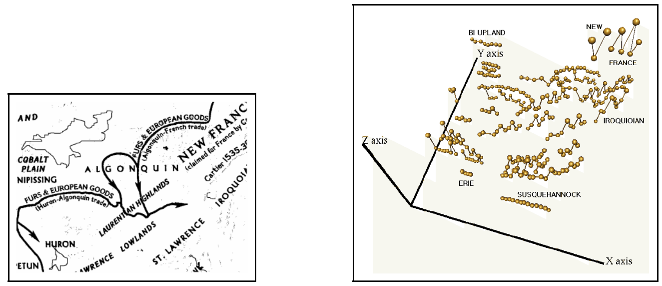
Fig.4(a) shows example characters placed in a testing raster map, Fig.4(b) shows the corresponding $G_{3d}(V, E)$ from a map character set $\{ c_i \}$.

Now we want group Words with $G_{3d}(V, E)$model. At first, we traverse the G_{3d} in Depth First Search manner so that join edges into a subgraph, depending on the connections of all nodes in G_D. Next, we have to find a word. Since it is common that characters in a word is placed linearly in maps, we believe that the centroids of characters in a word are located co-linearly as possible as. Thus we compute the degree of linearity by examining the path(=a



(a) Character set $\{ \ c_i \ \}$ in a map (b) Edge connections of (a)

Figure 4. The construction of 3-D neighborhood graph.



(a)An original map image (b) the corresponding 3-D graph of (a)

Figure 5. Word grouping result using 3-D graph approach

sequence of adjacent edges) to answer these are of a word or not.

Let θ_i be the angle between two successive edges, (v_{i-1}, v_i) and (v_i, v_{i+1}) and that is $\theta_i = \angle v_{i-1} v_i v_{i+1}$. Then for v_i, we find the maximal index of k satisfying following constraint.

$$\theta_{avg} = \sum_{j=1}^{k-1} \theta_j / n, \ \frac{1}{n} \sum_{i=1}^{k} (\theta_{avg} - \theta_i)^2 \ \leq \ \delta_0 \qquad (1)$$

where δ_0 is a threshold constant for linearity. Though characters in a word are adjusted on a base line strictly, there are different types or number of strokes. So the centroid can not be placed in a straight line exactly. Therefore δ_0 can not be 0. Finally, we traverse the each subgraph and calculate the local threshold by considering length of all nodes in a subgraph so that we remove edges if the length of an edge larger than a local threshold value. Fig.4(b) shows the 3-D neighborhood graphs which correspond to Fig.4(a). Fig.3(b) shows the result of grouping Fig.3(a).

3 Experiments and Conclusion

The proposed algorithm has been evaluated using a number of test images. These maps are raster maps with scale 1:5000 and digitized to 1300 by 1300 pixel dimension.

	Name	S_r	FRR	FAR	OC	SC
CHAR	MAP_1	98.4	1.6	5.8	24	106
	MAP_2	96.4	3.6	4	199	192
	MAP_3	97.4	2.6	9	159	151
WORD	MAP_1	98.2	1.8	1.78	24	106
	MAP_2	97.1	2.9	2.9	199	192
	MAP_3	97.3	2.7	3.6	159	151

Table 1 shows the final testing result. OC is the number of characters overlapped on a graphic symbol. SC is the number of characters which can be separable and not placed on a symbol. S_r is the successful extraction ratio which is defined as (the number of identified characters)/(OC + SC). FRR denotes the False Reject Ratio for all texts and FAR denotes the False Accept Ratio.

Our algorithm is adaptable to cope with the changes in text font styles, sizes and word orientations. The algorithm could also successfully isolate characters which were overlapped on other non-character symbols without a priori font or other domain specific information.

In the future we hope that this system would include an automatic character recognition system to make truly full-automatic geographic map analysis system or recognition of engineering drawings.

References

[1] 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.

[2] A. Nakamura and O. Shiku, *A Method for Recognizing Character Strings from Maps Using Linguistic Knowledge*, Proc. of 2th International Conf. on Document Analysis and Recognition, pp. 561-564, Oct. 20-24, 1993.

[3] L. Eikvil and K. Aas, *Tools for Interactive Map Conversion and Vectorization*, Proc. of Document Analysis and Recognition, Vol. 2, Aug. 14-16, 1995.

[4] 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, 1995.

[5] J. Ohya and A. Shio, *Recognizing Characters in Scene Images*, IEEE Transaction On PAMI, Vol. 16, No. 2, pp. 214-220, Feb. 1994.