Word Grouping in Document Images Based on Voronoi Tessellation

Yue Lu[1,2], Zhe Wang[2], and Chew Lim Tan[2]

[1] Department of Computer Science and Technology
East China Normal University, Shanghai 200062, China
[2] Department of Computer Science, School of Computing
National University of Singapore, Kent Ridge, Singapore 117543

Abstract. Voronoi tessellation of image elements provides an intuitive and appealing definition of proximity, which has been suggested as an effective tool for the description of relations among the neighboring objects in a digital image. In this paper, a Voronoi tessellation based method is presented for word grouping in document images. The Voronoi neighborhoods are generated from the Voronoi tessellation, with the information about the relations and distances of neighboring connected components, based on which word grouping is carried out. The proposed method has been evaluated on a variety of document images. The experimental results show that it has achieved promising results with a high accuracy, and is robust to various font types, styles, sizes, skew angles, as well as different text orientations.

1 Introduction

Segmenting word objects from document images is an essential component for most document analysis and recognition systems. Its goal is to group a set of image elements of characters, touched characters, or character portions into a word. The accuracy of the word grouping greatly influences the performance of the systems. If word grouping is incorrectly done, serious irreparable errors could occur in the subsequent processing. This is especially true for most word-based recognition systems and word spotting systems. However, it is not trivial to develop a word segmentation method with not only high accuracy but also robustness to various documents.

Many methods for word grouping in document images have been suggested by researchers. Flethchar and Kasturi[1] described a method in which Hough transform was used to group characters into words. Hough transform is applied to the centroids of the rectangles enclosing each connected component, the collinear connected components are thus located. The positional relationships between the collinear connected components are then examined to locate the words. This method, however, is not capable of dealing with the documents with various sizes of characters, because the process depends on the average height of all connected components. Jain and Bhattacharjee[2] considered text image as textured objects and used Gabor filtering for text analysis. Obviously, this method is sensitive

S. Marinai and A. Dengel (Eds.): DAS 2004, LNCS 3163, pp. 147–157, 2004.

to font sizes and styles, and it is generally time-consuming. Ittner and Baird[3] assumed that the distribution of scaled inter-symbol distance parallel to text-line orientation is bimodal with one model for inter-symbol spaces and the other mode for inter-word spaces. Wang et al.[4] presented a statistical-based approach to text word extraction that takes a set of bounding boxes of glyphs and their associated text lines of a given document and partitions the glyphs into a set of text words. The probabilities, estimated off-line from a document image database, drived all decisions in the on-line text word extraction. An accuracy of about 97% was reported. In [5], Park introduced a 3D neighborhood graph model which can group words in inclined lines, intersecting lines, and even curved lines. Sobottka[6] proposed an approach to automatically extract text from colored books and journal covers. Tan and Ng[7] gave a method using irregular pyramid structure. The uniqueness of this algorithm is its inclusion of strategic background information in the analysis.

A crucial step for word grouping is to find the neighbors in proximity of a particular image element. A naive approach is to compare the distances of the element to all the others in the image, and the neighbors are then defined by those with shorter distances. Such definition is not always accurate, because an element with a short distance is not a real neighbor sometimes. Voronoi tessellation(also named as Voronoi diagram) provides a useful tool which is capable of generating minimal in the number but complete neighbors of an element, i.e. only those elements that are closest are obtained, but all are included. The Voronoi tessellation of a collection of geometric objects is a partition of space into cells, each of which consists of all the points closer to one particular object than to any others. It divides the continuous space into mutually disjoint subspace according to the nearest neighbor rule. In the past decades, increasing attentions have been paid to the use of Voronoi tessellation for various applications.

The most important and significant contribution of the Voronoi tessellation to image analysis is that it introduces neighboring relations into a set of elements(e.g. connected components) on a digital image. In particular, it enables us to obtain neighbors without recourse to predetermined parameters. In recent years, there are some reports in the literature about applying Voronoi diagram to document image analysis. For instance, Ittner and Baird[3] applied the Delaunary triangulation, dual of the Voronoi diagram, to detect the orientation of text lines in a block, based on the assumption that most Delaunary edges lie within rather than between text lines. Xiao and Yan[8] described a method of text region extraction using the Delaunay tessellation. In both of the above two methods, the connected components in a document image are represented by their centroids. Such simplification is inappropriate in some cases, because the centroid is a poor representation of shapes for non-round elements. Since a document image generally contains various characters of different sizes and different intercharacter gap, the approximation of each element as a single point is too imprecise, and it does not adequately represent the spatial structure of

the page image. Therefore, the point Voronoi tessellation is unsuitable for some applications.

Considering the complex shapes of image elements, the use of area Voronoi diagram has been investigated for document image analysis. For example, Wang et al.[9] applied area Voronoi tessellation for segmenting characters connected to graphics, based on the observation that area Voronoi tessellation represents the shape of connected components better than the bounding box does. Kise et al. employed area Voronoi diagram to perform page segmentation[10] and text-line extraction[11].

Word grouping would evidently benefit from the information provided by the Voronoi tessellation. However, the research on this topic has not been extensively studied so far, except Burge and Monagan's work[12] which made an attempt using the Voronoi tessellation for grouping words and multi-part symbols in a map understanding system. An obvious shortcoming of their method is that it requires the necessary information of the resolution at which the processed image was scanned. In most general sense, the image resolution is unknown for a document analysis system in most cases.

Based on the area Voronoi tessellation, a method for grouping the image elements to word objects is proposed is this paper. No priori knowledge such as character font, character size or intercharacter spacing is required for the proposed method, and no special word orientations are assumed. Experimental result on real document images shows that more than 99% of words are successfully extracted.

2 Word Grouping Based on Voronoi Neighborhoods Analysis

We suppose the area Voronoi tessellation of a processed document image has been obtained. For the details about how to constructing the Voronoi tessellation in a digital image, readers can refer to [13]. A test image, as in Fig. 1, with skewed text of different font styles and sizes, is used to demonstrate the performance of our proposed word grouping method based on the analysis of Voronoi neighborhoods. As shown in Fig. 1(a), Voronoi edges lie between any two adjacent connected components (elements). In other words, every word component is represented as a set of Voronoi tessellations which are adjacent with one another. If two elements e_i and e_j share parts of their Voronoi edges, they are said to be Voronoi neighbors each other. From Fig. 1(b), we can find that the Delaunay triangulation, the dual of the Voronoi tessellation, shows us the relations among Voronoi neighbors. Each edge of the triangles connects two Voronoi neighbors. As a result, a neighborhood graph can be constructed from the area Voronoi tessellation. In the neighborhood graph, each node represents an image element, and each edge is a connection to its neighboring element. The distance between two Voronoi neighbors is treated as the weight of the edge connecting them. Such an edge is represented using a 3-tuple:

$$\mathcal{E}_{ij} = \{e_i, e_j, d_{ij}\}$$

(a)Voronoi tessellation

(b)Delaunay triangulation

Fig. 1. Voronoi tessellation of a document image.

where e_i and e_j are two Voronoi neighbors, and the distance d_{ij} is defined as follows. Suppose $L_{ij} = \{l_1, \cdots, l_m\}$ be the Voronoi edge between the two elements e_i and e_j, where l_k is a point on L_{ij}. Note that $d(l_k, e_i) \equiv d(l_k, e_j)$ according the definition of Voronoi edge, but in digitized space $d(l_k, e_i) = d(l_k, e_j) \pm 1$ is also a possible case. We define the d_{ij} as the shortest distance summation of the distance from the Voronoi edge L_{ij} to the elements e_i and e_j, viz.

$$d_{ij} = \min_{1 \le k \le m} (d(l_k, e_i) + d(l_k, e_j)) \tag{1}$$

Table 1 lists the distances of some neighbor pairs generated from Fig. 1. Then the goal of grouping elements to words becomes a goal to search the

Table 1. Neighbor pairs and their distances.

neighbor pair	distance	neighbor pair	distance	neighbor pair	distance
(1,2)	9	(2,4)	8	(3,5)	10
(1,8)	58	(2,1)	9	(3,4)	33
(1,9)	63	(2,13)	60	(3,18)	63
(1,12)	63			(3,25)	63
				(3,15)	70
(4,2)	8	(5,6)	8	(6,5)	8
(4,3)	33	(5,3)	10	(6,10)	27
(4,17)	64	(5,20)	59	(6,29)	64
(4,15)	76	(5,27)	66	(6,24)	70
		(5,25)	84	(6,20)	73
(7,11)	8	(8,9)	1		
(7,10)	11	(8,16)	19		
(7,36)	36	(8,19)	37		
(7,33)	59	(8,1)	58		
(7,30)	71				
...	...	...	...	...	...

neighborhood graph to generate subgraphs, so that connections among elements from different words are deleted, but connections among elements belonging to same word objects remain. The process of word grouping is, therefore, considered to be the selection of the edges in the neighborhood graph, which connects two elements potentially in the same word. To this end, we need criteria for deciding which connecting lines should be deleted, and which should remain.

We can see from Fig. 1 and Table 1, that an element generally has more than three neighbors. However, in most cases, only one or two of the neighbors are within the same word that the element belongs to. For example, in the case that two neighbors belong to the same word, one is its preceding character, the other is its succeeding character. Therefore, we take into account only the two nearest neighbors. One is of the most shortest distance and the other is of the second shortest distance from the element. An example is shown in Fig. 2, where only the two nearest neighbors with the most shortest distance are selected, whereas the others are excluded from further processing, i.e. we need only consider whether the two nearest neighbors should be grouped with the element in the subsequent process. With this, most of the connections between text lines are effectively eliminated.

Now the task we are facing is how to know a remainder connection between the elements is within a word or between words. To solve this problem, we employ four features defined as follows. Suppose the element e_k has two most nearest neighbors. They are e_f with the most shortest distance d_{kf}, and e_s with the second most shortest distance d_{ks}. The characteristics of them are as follows: the heights, widths and areas of e_k, e_f, e_s are (h_k, w_k, a_k), (h_f, w_f, a_f) and (h_s, w_s, a_s), respectively. The four features are defined as:

$$f_1 = \frac{d_{kf}}{min((h_k + w_k)/2, (h_f + w_f)/2)} \tag{2}$$

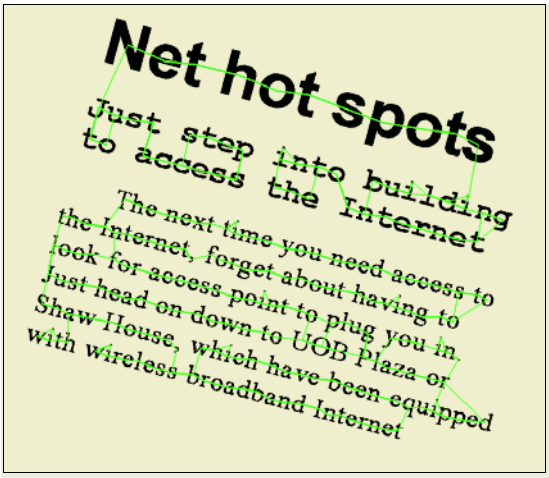

Fig. 2. Two selected nearest neighbors with most shortest distances.

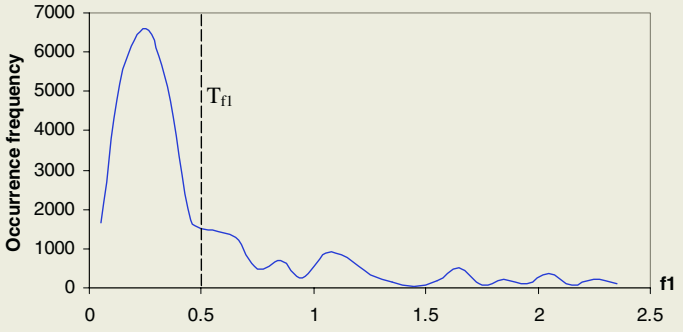

Fig. 3. Occurrence frequency vs. f_1.

$$f_2 = \frac{d_{ks}}{min((h_k + w_k)/2, (h_s + w_s)/2)} \tag{3}$$

$$f_3 = \frac{d_{ks} - d_{kf}}{d_{ks}} \tag{4}$$

$$f_4 = \frac{a_k}{a_f} \tag{5}$$

To investigate the features f_1, f_2 and f_3, 100 real document images with varying character fonts, sizes and different intercharacter, inter-word and inter-textline gaps, are utilized to obtain the statistical characteristics of them. The statistical results of occurrence frequencies are demonstrated in Fig. 3-5, respectively. Feature f_1 is the normalized distance from e_k to its most nearest neighbor e_f. We can see the fact, from Fig. 3, that the majority of the pairs (e_k, e_f) are within same words, but exceptions do exist. One exception is found

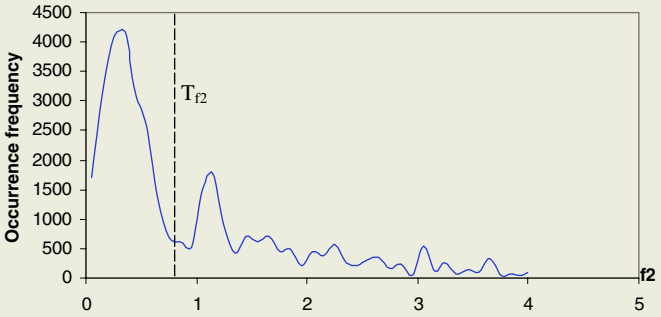

Fig. 4. Occurrence frequency vs. f_2.

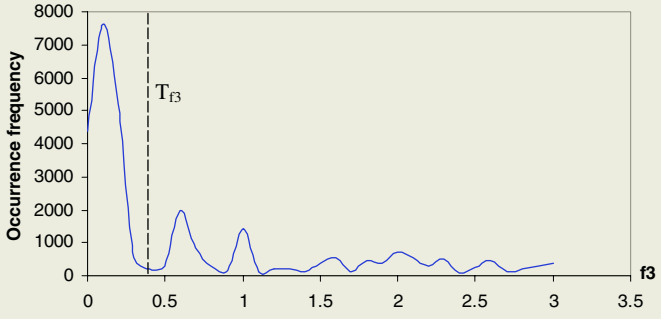

Fig. 5. Occurrence frequency vs. f_3.

in some single-character words like the indefinite article "a". Another exception is found in some elements with small areas like the dots on 'i', 'j', and the punctuation. For the former case, the "a" need not group with others. As for the latter case, we will further discuss it later. Therefore, we can select T_{f1} as shown in Fig. 3 as a threshold. If $f_1 < T_{f1}$, then e_k and e_f should be grouped.

Feature f_2 is the normalized distance from e_k to its second most nearest neighbor e_s. Figure 3 shows us that, most of the pairs (e_k, e_s) are within same words. But there are some pairs from different words. For example, the last character of a word, and the first character of its succeeding word generates such a pair. So, the decision cannot be made only using Feature f_2.

Feature f_3 is the normalized difference between the distances d_{ks} and d_{kf}, as given in Fig. 5. If an element is located in the middle of a word(i.e. neither the first character nor the last character), the value of its f_3 feature is small. Otherwise, the value is large. We combine the features f_2 and f_3 to produce another criterion. If f_2 is less than T_{f2} and f_3 is less than T_{f3}, then e_k and e_s are grouped, and of course e_k and e_f are grouped as well, where T_{f2} and T_{f3} are two thresholds as shown in Fig. 4 and 5 respectively.

Then, two criteria can be summarized as follows:

Rule 1: if $f_1 < T_{f1}$, then e_k and e_f are grouped.
Rule 2: if $f_2 < T_{f2}$ and $f_3 < T_{f3}$, then e_k, e_f and e_s are grouped.

Based on the above process, there are two problems left. One is that many dots of characters 'i' and 'j' are not grouped to the corresponding words. The other is that some punctuation marks are erroneously grouped with words. From the properties of the elements, it is difficult to distinguish the dots on the characters 'i' and 'j' from the punctuation like commas and full stops. Anyway, they have the same characteristic of relative small areas. We therefore employ Feature f_4, the area ratio, to identify them from others. If f_4 of an element is less than a predefined threshold T_{f4} (say 0.25 empirically), it undergoes a further process then.

Our observations find that the two nearest neighbors of the dots on 'i' and 'j' come from the same word in general. On the other hand, for an element of punctuation, its most nearest neighbor is generally the last character of its preceding word, whereas its second nearest neighbor is the first character of its succeeding word. As a result, its f_3 normally has a larger value. We then utilize the following criteria to estimate them:

Rule 3: if $f_3 < T_{f3}$ and $f_4 < T_{f4}$, then e_k and e_f are grouped.
Rule 4: if $f_2 > T_{f2}$, $f_3 > T_{f3}$ and $f_4 < T_{f4}$, then e_k cannot be grouped with e_f.

Generally speaking, commas and full stops are commonly used punctuation marks in documents. Rule 4 can effectively detect them. It is also worth noting that, some special symbols such as dash('-'), tilde('∼'), and various kinds of parentheses '{','}','[',']','(',')', should be detected and excluded from word grouping. For this purpose, Kim's method[14] can be applied as post-processing.

Figure 6(a) shows the processing results of Fig. 2, and Fig. 6(b) gives the corresponding Voronoi edges separating the grouped words.

3 Experiment Results

To evaluate the performance of the proposed method, the document image, from different sources, with various characters of sizes and fonts, are used for the test. They are 100 page document images selected from the UW document image database, and the images provided by the Digital Library of our university, including 328 document images of scanned books and 476 document images of scanned outdated student theses.

Figure 7 shows an example of word grouping results carried on an image of scanned books, in which the segmented word entities are bounded using rectangle boxes. It has be showed that our proposed method can deal with the words of different text orientation. The method is also insensitive to the character sizes and fonts.

The word grouping performance on the test documents is tabulated in Table 2, including the rates of correction, splitting(fragmentation) and over-merging. An encouraging accuracy of over 99% has been achieved. The errors of word extraction can be divided into two categories. One is fragmentation or splitting, in which one word is erroneously divided into two or more words. The other one

(a)Neighbors within words

(a)Voronoi edges among words

Fig. 6. Word grouping results of Fig. 1.

is over-merging, in which two or more words are merged into one word. Some examples of failed word bounding by the present algorithm are illustrated in Fig. 8.

4 Conclusions

Voronoi tessellation is an effective tool for representing the neighboring relations among elements in a digital image. A Voronoi neighborhood based algorithm for grouping word objects in document images is presented in this paper. The Voronoi neighborhoods are generated from the Voronoi tessellation for word grouping, with the information about the relations and distances of neighboring

Fig. 7. An example of word grouping.

Table 2. Performance of word grouping.

	UW image	NUS scanned books	NUS scanned theses	Average
Accuracy(%)	98.83	99.06	99.26	99.05
Splitting(%)	0.56	0.28	0.43	0.42
Over-merging(%)	0.61	0.66	0.31	0.53

Fig. 8. Examples of failed word bounding.

connected components. The experimental results on various document images have shown that the proposed approach has achieved promising results with a high accuracy, and is robust to various font types, styles, sizes, skew angles, as well as text orientations.

Acknowledgements

This research is jointly supported by the Agency for Science, Technology and Research, and Ministry of Education of Singapore under research grant R-252-000-071-112/303.

References

1. L.A. Fletcher and R. Kasturi, A Robust Algorithm for Text String Separation from Mixed Text/Graphics Images, IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 10, No. 6, pp. 910-918, 1988.
2. A. Jain, S. Bhattacharjee, Text segmentation using Gabor filters for automatic document processing, Machine Vison Applications, Vol.5, pp.169-184, 1992.
3. D. J. Ittner and H. S. Baird, Language-free layout analysis, Proceedings of Second International Conference on Document Analysis and Recognition, Tsukuba, pp.336-340, 1993.
4. Y. Wang, I. T. Phillips, R. Haralick, Statistical-based approach to word segmentation, Proceedings of 15th International Conference on Pattern Recognition, vol.4, Barcelona, Spain, pp.555-558, September 2000.
5. H.C. Park, S.Y. Ok, Y.J. Yu and H.G. Cho, A word extraction algorithm for machine-printed documents using a 3D neighborhood graph model, Int. J. Doc. Anal. Recognition 4: 115-130, 2001.
6. K. Sobottka, H. Kronenberg, T. Perroud and H. Bunke, Text extraction from colored book and journal covers, Int. J. Doc. Anal. Recognition 2: 163 - 176, 2000.
7. C.L. Tan, P.O. Ng: Text extraction using pyramid. Proc. Pattern Recognition 31(1):63-72, 1997.
8. Y. Xiao and H. Yan, Text Region Extraction in a Document Image Based on the Delaunay Tessellation, Pattern Recognition, Vol. 36 (2003), No. 3, pp. 799-809, 2003.
9. Y. Wang, I. T. Phillips, and R. Haralick, Using Area Voronoi Tessellation to Segment Characters Connected to Graphics, Proceedings of Fourth IAPR International Workshop on Graphics Recognition (GREC2001), Kingston, Ontario, Canada, September, 2001, pp.147-153.
10. K. Kise, A. Sato, and M. Iwata, Segmentation of Page Images Using the Area Voronoi Diagram, Computer Vision and Image Understanding, vol. 70, no. 3, pp. 370-382, June 1998.
11. K. Kise, M. Iwata, A. Dengel and K. Matsumoto, Text-Line Extraction as Selection of Paths in the Neighbor Graph, Document Analysis Systems, pp.225-239, 1998.
12. M. Burge, G. Monagan, Using the Voronoi tessellation for grouping words and multipart symbols in documents, Proceedings of SPIE International Symposium on Optics, Imaging and Instrumentation, Vol.2573, San Diego, California, pp.116-124, July 1995.
13. A. Okabe, B. Boots, K. Sugihara, S. N. Chiu, Spatial tessellations: Concepts and applications of Voronoi disgrams(Second Edition), Chichester: John Wiley, 2000.
14. S.H. Kim, C.B. Jeong, H.K. Kwag and C.Y. Suen, Word segmentation of printed text lines based on gap clustering and special symbol detection, Proceesings of International Conference on Pattern Recognition, Quebec, Canada, 2002, vol.2, pp. 320-323.