be — A Block Extraction Program Based on the Area Voronoi Diagram MEMO

Koichi Kise
Dept. of Computer & Systems Sciences,
Osaka Prefecture University
kise@cs.osakafu-u.ac.jp

October 15, 1999

1 Changes

1. 991015
   - first version

2 Files

The archive `be.tgz` contains the following files.

`Makefile`	makefile
`const.h`	constants and default values of parameters.
`defs.h`	data types used in the program.
`extern.h`	declaration of global variables.
`function.h`	prototype declaration of some functions.
`read_image.h`	header file for read_image.c
`main.c`	main routine for the program `be`
`read_image.c`	utility functions for reading a binary image. Sun rasterfiles and TIFF files are supported.
`img_to_site.c`	extraction of generators (points) for Voronoi diagrams from an input binary image.
`label_func.c`	functions for handling a labeled image.
`voronoi.c`	construction of Voronoi diagrams.
`heap.c`, `edgelist.c` `geometry.c`, `memory.c` `sites.c`, `output.c`	utility functions for voronoi.c.
`erase.c`	extraction of blocks from the area Voronoi diagram.
`bit_func.c`	functions for handling binary images.
`cline.c`	command line analyzer.
`usage.c`	usage.
`hash.c`	functions for hash tables.
`dinfo.c`	functions for displaying information.

For further details of Sun rasterfile, see the manual page on a computer (`man rasterfile`).

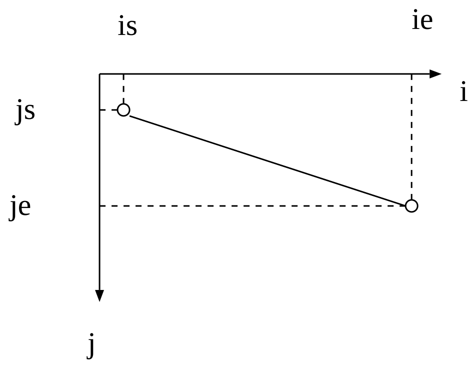

Figure 1: A line segment.

3 Installation

3.1 Requirements

For the compilation of the program, it is required that the library `libtiff` has been installed in your computer.

3.2 Installation on a Linux machine

The program is developed under the Linux OS 2.0.36 (RedHat 5.2) with gcc Ver.2.7.2.3. If you have a Linux computer, the installation is as follows:

1. Extract the files from the archive by `tar xvzf be.tgz`.
2. In the directory which contains the files, just do `make`.
3. If the compilation completes, you can find `be` in the current directory.

3.3 Installation on machines with other OS's

We are sorry for not testing the program on machines with other OS's.

4 Usage

4.1 Requirements

The program requires at least 64MB of real memory to process an A4-size document image scanned with the resolution of 300dpi. We recommend that your computer has more than 128MB memory.

4.2 Input and output

The program `be` takes two mandatory arguments: input and output.

input The input is a binary document image whose format is either sun raster or TIFF.

output The output is a list of (i_s, i_e, j_s, j_e) (see Fig. 1) each of which represents a line segment on borders of blocks.

`be` also accepts optional arguments to change the default values of parameters, etc. The list of arguments and their brief descriptions are shown in Table 1. You can find the details of these arguments in the next section.

Table 1: Optional arguments for `be`

argument	type	default	meaning
`-sr`	int.	`SAMPLE_RATE`	sampling rate of points on contours.
`-nm`	int.	`NOISE_MAX`	max. perimeter of a noise CC.
`-fr`	float	`FREQ_RATE`	threshold setting parameter for Td2.
`-ta`	float	`Ta_CONST`	threshold value of the area ratio.
`-sw`	int.	`SMWIND`	the size of the smoothing window for frequency distribution of *d*.
`-dparam`	none	`NO`	If this option is supplied, the values of parameters used to run the program are listed.

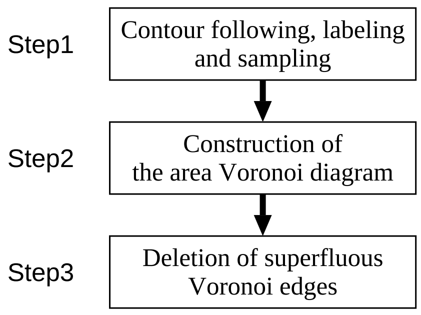
Step1
Contour following, labeling and sampling
Step2
Construction of the area Voronoi diagram
Step3
Deletion of superfluous Voronoi edges

Figure 2: Steps of processing

5 Overview of the method

The method consists of 3 steps shown in Fig. 2. The description of the method can also be found in [1].

5.1 Contour following, labeling and sampling

The first step is to extract connected components and sample points on their contours both of which are based on contour following [1]. The default value of the sampling rate is represented as `SAMPLE_RATE` in the source code: every `SAMPLING_RATE`-th pixel on a contour of a connected component is stored as a generator for the point Voronoi diagram. An example of sample points is shown in Fig. 3(b). In this step, connected components are discarded as noise if their perimeter is less than or equal to `NOISE_MAX`.

5.2 Construction of the area Voronoi diagram

The next step is to construct the area Voronoi diagram from the sample points and the connected components extracted in the previous step. The procedure is as follows:

1. Construct the point Voronoi diagram from the sample points.
2. Construct the area Voronoi diagram from the point Voronoi diagram by selecting Voronoi edges lying *between* connected components.

Examples of the point and the area Voronoi diagrams are shown in Figs. 3(c) and (d), respectively.

[1]Note that this is the most time-consuming step in the method; you can speed up the processing if you replace the program with a more sophisticated one.

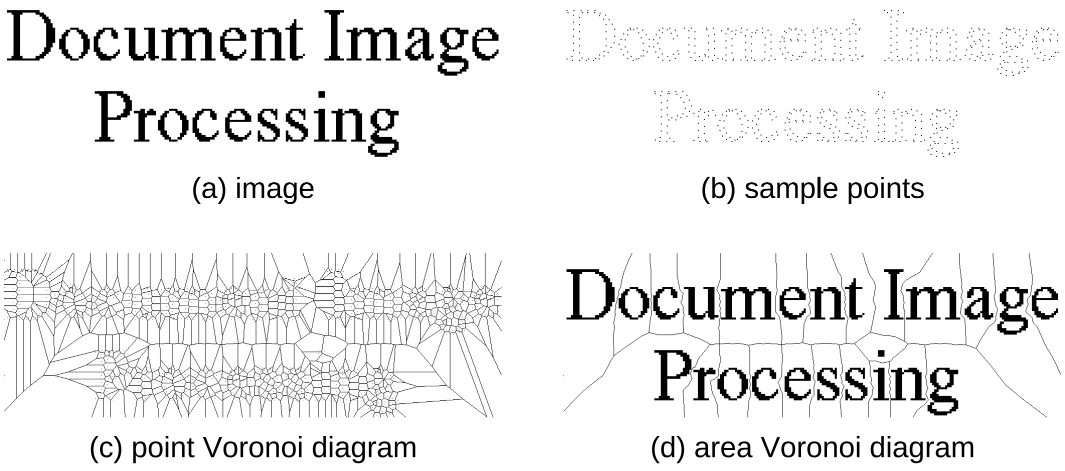
Document Image Processing
(a) image
(b) sample points
(c) point Voronoi diagram
Document Image Processing
(d) area Voronoi diagram

Figure 3: Examples of the processing results.

5.3 Deletion of superfluous Voronoi edges

As shown in Fig. 3(d), Voronoi edges lie between any adjacent connected components. Thus Voronoi edges in the area Voronoi diagram generally include correct boundaries of blocks.

In our method, appropriate Voronoi edges are selected by deleting superfluous ones. For this purpose, we employ two characteristic features: distance and area ratio of connected components.

distance the minimum distance d between adjacent connected components. Let p_i and q_j be sample points on contours of adjacent connected components p and q. Then the distance d is defined by:

$$d = \min_{i,j} d(p_i, q_j),$$

where $d(p_i, q_j)$ indicates the Euclidean distance between points p_i and q_j.

area ratio the ratio of area of two adjacent connected components defined by:

$$a_r = \frac{max(a_1, a_2)}{min(a_1, a_2)},$$

where a_i represents the area (the number of black pixels) of a connected component.

We delete a Voronoi edge as superfluous if at least one of the following conditions is satisfied:

$$\frac{d}{T_{d1}} \leq 1 \tag{1}$$

$$\frac{d}{T_{d2}} + \frac{a_r}{T_a} \leq 1 \tag{2}$$

where T_{d1}, T_{d2} ($T_{d1} < T_{d2}$), and T_a are thresholds. Figure 4 illustrates the $d - a_r$ plane in which a Voronoi edge corresponds to a point. Eqs. 1 and 2 indicate that Voronoi edges in the shaded area are deleted.

The thresholds are determined as follows:

- T_a is fixed to the value `Ta_CONST`. This value can be changed by the command line option `-ta`.

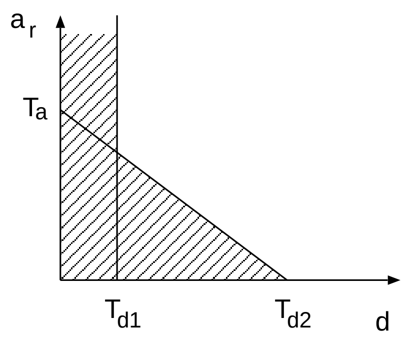

Figure 4: Criteria for deleting superfluous Voronoi edges. A Voronoi edge corresponds to a point in this d–a_r space. The Voronoi edges in the shaded area are deleted.

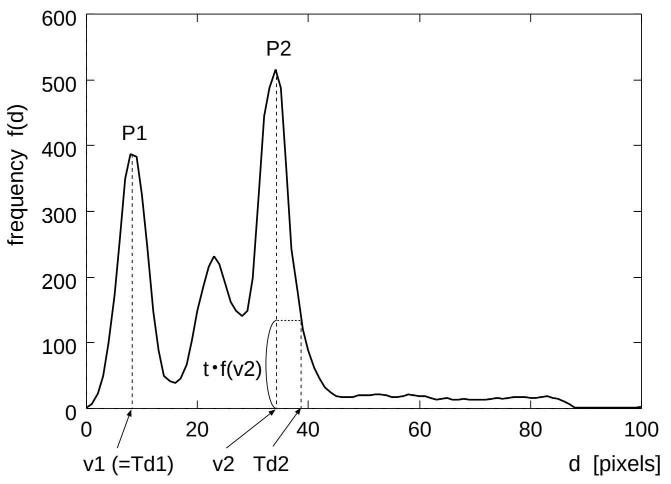

Figure 5: A frequency distribution of the distance d.

- T_{d1} and T_{d2} are calculated based on the frequency distribution of the distance d. Figure 5 shows an example of the distribution. Let P1 and P2 be the two largest peaks (P1 is at the smaller distance). Note that P1 and P2 correspond to the intercharacter and the interline gaps of body text regions, respectively. We utilize as T_{d1} the distance of the peak P1. The value of T_{d2} should be larger than the interline gap not to split text blocks. Thus we set T_{d2} as shown in Fig. 5 where t is specified by the constant `FREQ_RATE` in the program. This can also be changed by the command line option `-fr`.

6 Questions, Comments

If you have any questions or comments on the program, the algorithm and this note, please feel free to ask me. The e-mail address is:

`kise@cs.osakafu-u.ac.jp`

Bug reports are also welcomed.

References

[1] 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, Academic Press, 1998.6.

[2] K.Kise, M.Iwata and K.Matsumoto, On the Application of Voronoi Diagrams to Page Segmentation, *Proc. of Workshop on Document Layout Analysis and Its Applications (DLIA99)*, Bangalore, India, (IV-C), 1999.9.

[3] K.Kise, A.Sato and K.Matsumoto, Document Image Segmentation as Selection of Voronoi Edges, *Proc. of IEEE Workshop on Document Image Analysis*, San Juan, Puerto Rico, pp.32-39, 1997.6.