MVA'98 IAPR Workshop on Machine Vision Applications, Nov. 17-19, 1998, Makuhari, Chiba, Japan

13—15 An Algorithm for Reducing Text Line Candidates of Incorrect Orientation

Hideaki Goto *
Education Center for Information Processing,
Tohoku University

Hirotomo Aso †
Graduate School of Engineering,
Tohoku University

Abstract

Japanese documents often contain both horizontally and vertically printed text lines in the same page. It has been required for document analysis systems to detect correct orientation of text lines and to select text line candidates of correct orientation. We designed an efficient framework for the procedure and developed some algorithms which reduce text line candidates of incorrect orientation. However, our previous system could handle only text lines with slight skew ($\pm 10°$).

In order to improve the performance of our document analysis system, we recently developed an improved algorithm for text line extraction which can handle curved or wavy text lines as well as straight text lines of arbitrary orientation. We have also developed a new candidate reduction algorithm for arbitrarily oriented text lines. This paper describes mainly the detail of the candidate reduction algorithm. The overview of the system is also mentioned.

The candidate reduction algorithm is based on the *a priori* knowledge that inter–line spacing is much wider than inter–character spacing in most documents. Experimental results show that the algorithm works very well for many documents, including very complicated ones, written in both Japanese and English, and also for the documents which contain curved or wavy text lines.

1 Introduction

Documents in some languages, for example Japanese, often contain both horizontally and vertically written text lines in the same page. Japanese characters are usually printed by fixed–width fonts. One is sometimes unable to guess the right orientation of text lines only viewing the arrangement of rectangles without reading strings (Figure1). Two hypotheses on the orientation of text line, horizontal and vertical, must be carried over until character recognition or until text analysis. However, it is wasteful keeping the two hypotheses for every text region, because character recognition process is computationally expensive. It is required for document analysis systems to pick up text lines of correct orientation at as early a stage as possible.

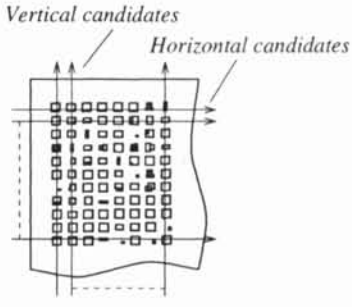


Figure 1: Two hypotheses on the orientation of text line

Several orientation detection (or skew detection) methods have been developed [1, 2]. They were designed to detect page skew or block skew, and such an orientation detection method is applied to text blocks which have been extracted during physical structure analysis. However, such a framework is based on the assumption that text blocks would obviously be defined and extracted. Recently, it has been required that document analysis systems should handle various documents, not only documents with simple layout, but also ones with very complex layout. Note that it is impossible even for us, human beings, to point out text blocks exactly in some complex documents as shown in Figure6. Text line candidates of incorrect orientation should be reduced without extracting text blocks as far as it is possible.

We designed a framework for reducing text line candidates of incorrect orientation [3]. We also developed some algorithms for this framework. In the framework, we first used a text line extraction method called "Linear Segment Linking (LSL, for short)" [4]. The method has an advantage that it is applicable to complex documents directly because it is independent of text block extraction. However, in

*Address: Kawauchi, Aoba-ku, Sendai 980–8576, Japan.
E-mail: hgot@ecip.tohoku.ac.jp
WWW: http://www.ecip.tohoku.ac.jp/~hgot/
†Address: Aramaki, Aoba-ku, Sendai 980–8578, Japan.
E-mail: aso@ecei.tohoku.ac.jp

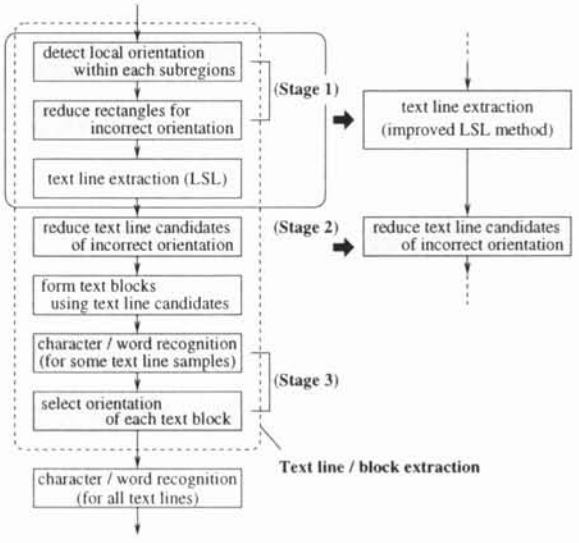


Figure 2: Overview of text line / block extraction

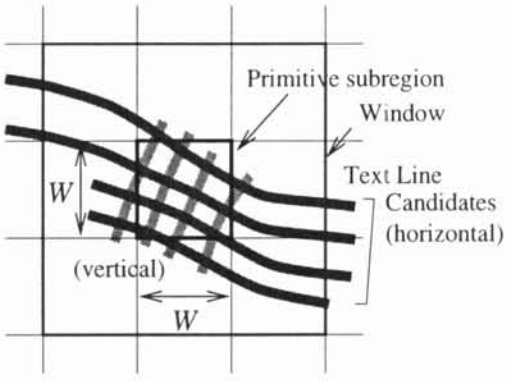


Figure 3: Primitive subregion and window

LSL and in the candidate reduction algorithms, local skew angle of text line is limited to around $\pm 10°$ to horizontal or vertical axis of image.

In order to improve our document analysis system, we recently developed an improved version of LSL [5]. Curved or wavy text lines as well as straight text lines of arbitrary orientation can be extracted by the method. Text line candidates for multiple orientation are obtained by the method. However, we did not have any candidate reduction algorithm for arbitrarily oriented text lines. In most documents, inter–line spacing is much wider than inter–character spacing. Using this property, text line candidates of incorrect orientation would be reduced before applying a character recognition process to the candidates. We have developed a new algorithm for reducing text line candidates of incorrect orientation which is able to handle curved or wavy text lines. This paper describes the detail of the algorithm and the experimental results showing the effectiveness of the algorithm.

2 Candidate reduction algorithm

2.1 Overview of text line / block extraction

The framework which we proposed previously for candidate reduction is shown in the left side of Figure2. The candidate reduction procedure consists of three stages. At Stage 1, rectangles (bounding boxes) for the text line extraction in an incorrect orientation are removed comparing the linearity of the rectangle arrangement for two orientations, horizontal and vertical. This stage was designed to reduce rectangles especially in text blocks printed with proportional fonts. This stage has been merged into the improved version of LSL method, since the text line extraction is now strongly dependent of local skew detection and we thought it was quite natural for the rectangle reduction being incorporated into the procedure of text line extraction. Readers are referred to [4] and [5] for the details of the text line extraction.

Stage 2 is the process for text line candidate reduction and Stage 3 is the final process for selecting text blocks of correct orientation. The algorithm at Stage 2 has been improved so that curved or wavy text lines can be handled. The detail of the algorithm is described in the following section.

2.2 New algorithm for candidate reduction

The basic idea of the algorithm at Stage 2 is as same as that of the algorithm we proposed before [3]. First, a creation of window is introduced. A document image is partitioned into square regions with constant width and height (W). We call the region "primitive subregion" (Figure3). We have chosen 128 (pixels) for W so that up to 5 or 6 text lines with the smallest characters (4pt) go across a primitive subregion in a 400dpi image. For each primitive subregion, the number of the text line candidates crossing the subregion is calculated. A "window" is defined as a virtual subregion of minimum size where the number of text lines is more than or equal to N_l. The window is the set of primitive subregions of size 1×1 or 3×3 surrounding a subregion (Figure3). If the number of text lines is less than N_l for the window of size 3×3, the orientation at the central primitive subregion is marked as "double–oriented". The "double–oriented" here means that the two hypotheses for both horizontal and vertical text line are remaining.

Then, a mean value of local character density for horizontal and vertical text lines is calculated at each primitive subregion (Figure4). By LSL, a text line candidate is represented by a set of rectangles. A local region is set around each rectangle. Width and height of the region are set to be $20H(i)$

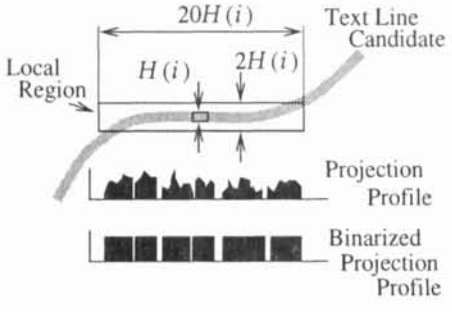


Figure 4: Local projection profile of text line

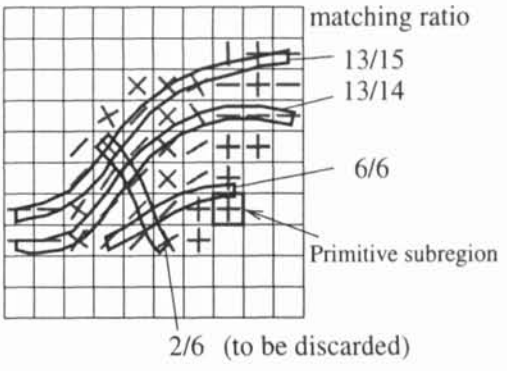


Figure 5: Matching ratio of text lines

and $2H(i)$ respectively, where $H(i)$ denotes the estimated height of text line around the i-th rectangle. The algorithm is less sensitive to the change of the coefficients, 20 and 2, in many documents. A projection profile is obtained for each local region. The character density is defined as the number of non–zero points divided by the length of the projection profile. The mean value of character density for an orientation is calculated by averaging all the character densities for the rectangles in the window around the primitive subregion.

At each primitive subregion, the difference between two mean values of the character density for horizontal and vertical orientations is calculated. If the absolute value of this difference exceeds T_{diff}, the orientation at the primitive subregion is set to the orientation with larger character density. Otherwise, the label of the primitive subregion is kept to be "double–oriented".

A matching ratio of orientation is estimated for each text line candidate. Let M_k be the number of subregions with their orientation same as the local orientation of the k-th text line candidate, and let N_k be the total number of subregions through which the k-th candidate crosses. The "matching ratio" is represented by M_k/N_k. Some examples are shown in Figure5. If the matching ratio is less than a threshold, T_{match}, the corresponding candidate is discarded. The optimum values, $N_l = 3$, $T_{diff} = 0.2$ and $T_{match} = 0.5$, were found by preliminary experiments using many document images. Note that text line candidates of correct orientation should not been discarded even in text blocks in which inter–line spacing is almost equal to or slightly narrower than inter–character spacing (Figure1). A greater value is desirable for T_{diff} in order to keep this property. However, too great a value will result in the lowering of the candidate reduction efficiency. We chose an optimum value for T_{diff} considering the above problems.

3 Experiments

The performance of the proposed method strongly depends on the property of a given document image. Due to the principle of the algorithm, the process does not work well for documents whose inter–line spacing is narrower than or almost equal to inter–character spacing.

An example of a complex document is shown in Figure6. The original image is shown in gray, and extracted text lines are represented by black lines. In Figure6(a), we have a large number of short text line candidates of incorrect orientation created by the text line extraction. Total number of the text line candidates is 1178, which is reduced to 201 by the candidate reduction. The number of candidates of correct orientation is 78, while only 20 candidates of incorrect orientation is left. We still have 103 false candidates which come from figures. However, they may be reduced by a region discrimination method.

The candidate reduction method was also applied to 170 document images out of JEIDA '93 database [6]. The documents we used consists of Japanese text, English text, or the mixed. The text lines in the images are horizontally, vertically written, or mixed. These images are of scanned journal pages, technical articles, pocketbooks, etc.

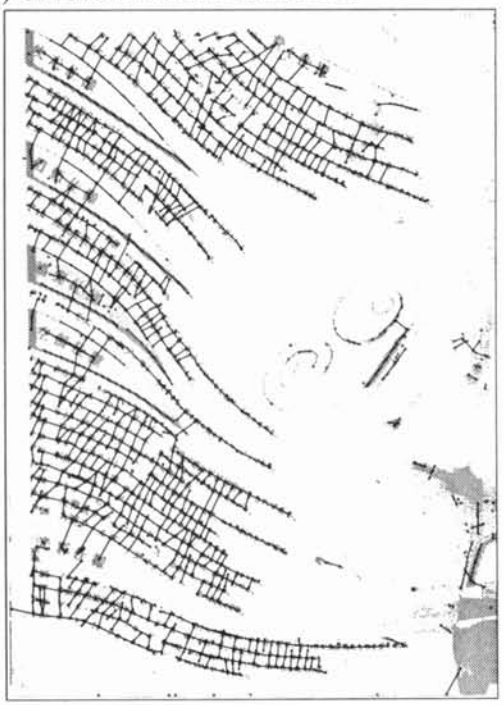
For all documents except some ones whose inter–line spacing was narrower than or almost equal to inter–character spacing, the candidate reduction worked very well. Only a few text line of incorrect orientation was left. Almost all the candidates of correct orientation were preserved, while only a few candidates for very short text lines was discarded.

4 Conclusion

We have proposed an algorithm for reducing text line candidates of incorrect orientation in this paper. We used the framework for candidate reduction which we had previously designed [3]. The new candidate reduction algorithm was implemented and incorporated into the framework as the process of Stage 2. The new algorithm can handle curved or wavy text lines. Moreover, the new algorithm is applicable to any page image directly since it is independent of text block extraction.

Experimental results show that the algorithm works very well for many documents, including very

(a) before candiate reduction



(b) after candidate reduction

Figure 6: An example of candidate reduction

complicated ones, written in both Japanese and English, and also for the documents which contain curved or wavy text lines.

We do not have any text block extraction algorithm for curved text lines yet. As mentioned in introduction, it is sometimes difficult for us to define text blocks in some complex documents such as shown in Figure6. If we regard each text line candidate as a text block, Stage 3 will work well, because the stage are expected to work well for a text block which consists of only one text line candidate. However, the total processing speed of the system will get worse in such a case. Further researches are required on the text block extraction and the candidate reduction at Stage 3.

References

[1] H.S. Baird, "The skew angle of printed documents", *Documet Image Analysis*, IEEE, pp.204–208, 1995.

[2] L. O'Gorman, "The Document Spectrum for Page Layout Analysis", IEEE Trans. on PAMI, vol.15, no.11, pp.1162–1173, 1993.

[3] H. Goto and H. Aso, "A Framework for Detecting and Selecting Text Line Candidates of Correct Orientation", Proc. 14th Int. Conf. Patt. Recogn. (ICPR'98), pp.1074–1076, 1998.

[4] H. Goto and H. Aso, "Robust and Fast Text-line Extraction Using Local Linearity of the Textline", Systems and Computers in Japan, vol.26, no.13, pp.21–31, Nov. 1995. (Translated from Trans. IEICE(D–II), vol.J78–D–II, no.3, pp.465–473, 1995.)

[5] H. Goto and H. Aso, "Extracting Arbitrarily Oriented Text Lines Using Local Linearity of Text Line", Technical Report of IEICE, PRMU98–14, pp.9–16, May 1998. (in Japanese)

[6] `http://www.etl.go.jp/etl/gazo/docidb/`