

May 10, 2000

RT0361

Human-Computer Interaction; Multimedia 12 pages

# Research Report

## Document Reader for the Visually Disabled

Kazuhide Sugawara

IBM Research, Tokyo Research Laboratory

IBM Japan, Ltd.

1623-14 Shimotsuruma, Yamato

Kanagawa 242-8502, Japan



**Research Division**

**Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich**

### **Limited Distribution Notice**

This report has been submitted for publication outside of IBM and will be probably copyrighted if accepted. It has been issued as a Research Report for early dissemination of its contents. In view of the expected transfer of copyright to an outside publisher, its distribution outside IBM prior to publication should be limited to peer communications and specific requests. After outside publication, requests should be filled only by reprints or copies of the article legally obtained (for example, by payment of royalties).

# Document Reader for the Visually Disabled

Kazuhide Sugawara

Tokyo Research Laboratory, IBM Japan Ltd.

1623-14, Shimotsuruma, Yamato, Kanagawa 242, Japan

Phone/Fax: (81)46-215-4602/(81)46-215-4282

E-mail: sugawara@jp.ibm.com

## Abstract

*In this paper, we present a system for reading documents, which can be used by the visually disabled by themselves. Our system uses speech to transmit the information to the users. To cope with the slowness of the information transmission speed caused by the one-dimensionality of speech, the logical structure of the scanned documents is extracted and used to accelerate access to the information. A tree-traversing command is provided for navigating through the logical tree generated by the system. A layered speech menu/help is used to guide users. The scanned and recognized pages can be linked on the basis of the page numbers detected, and can be stored in a file system for later use.*

Keywords: document image analysis, speech user interface, logical structure extraction, document reader for the blind

# 1 Introduction

Visually disabled people have long been unable to access printed documents, except for a few that have been translated into braille by sighted people. Some recent optical character recognition (OCR) systems allow visually disabled people to read printed documents by themselves by combining OCR with text-to-speech synthesizers. However, these systems only deliver a straightforward transcription of the printed text into speech, and the one-dimensionality of speech prohibits quick access to the parts of the document of interest to the user.

To cope with the problem, we propose a new approach for reading printed documents by extracting their logical structures as well as physical layouts [1]. We named our system Optical Media Reader (OMR) to emphasize that it is not just a character recognizer, but has functions for obtaining additional information. As examples of the range of functions it identifies image areas and text areas of the document and relates them if possible, and it detects the name of previously registered magazines by examining the logos on the cover pages.

The system extracts the logical tree structure from the scanned document page. The user can read the entire document by using basic functions for traversing the tree structures: descending an edge, listing the children, moving among siblings, ascending an edge, and so on. These commands can be easily input by using the numerical keypad on the keyboard. We also provided functions related to the physical layout of the page. The headers and footers are treated separately from the body of the document, and can be accessed independently while reading the body of the page. We made a prototype of the OMR and evaluated the system with visually disabled users.

In Section 2, we explain our system briefly. Then, in Section 3, we describe its major functions and methods for extracting the logical structure of a scanned document. Section 4 describes the management of the structured documents. Section 5 describes the evaluation results, while Section 6 discusses our future directions.

## 2 System Overview

The input to the OMR is obtained by scanning the documents. If the task is to determine the name of the magazine, the image is scanned in color and processed by the title-matching module. Inner pages are scanned in monochrome, and after preprocessing for estimating and correcting skew of the document, the text lines and its directions are detected. Then, the layout is analysed to extract the logical structure from the scanned page. An image of the text lines detected is sent to the OCR engine to obtain the character codes for the text lines. The text is augmented by tags expressing its logical structure, and is presented to the user in the form of synthesized speech, which reflects the tags and the user's settings as well as the content of the text itself. The user can control the position in the document from which to start reading, and the OMR settings, such as the gender of the synthesized voice, speed, and pitch, by pressing keys on the numerical keypad. Optionally, a Braille output device can be connected to the system. The system

configuration is shown in Figure 1.

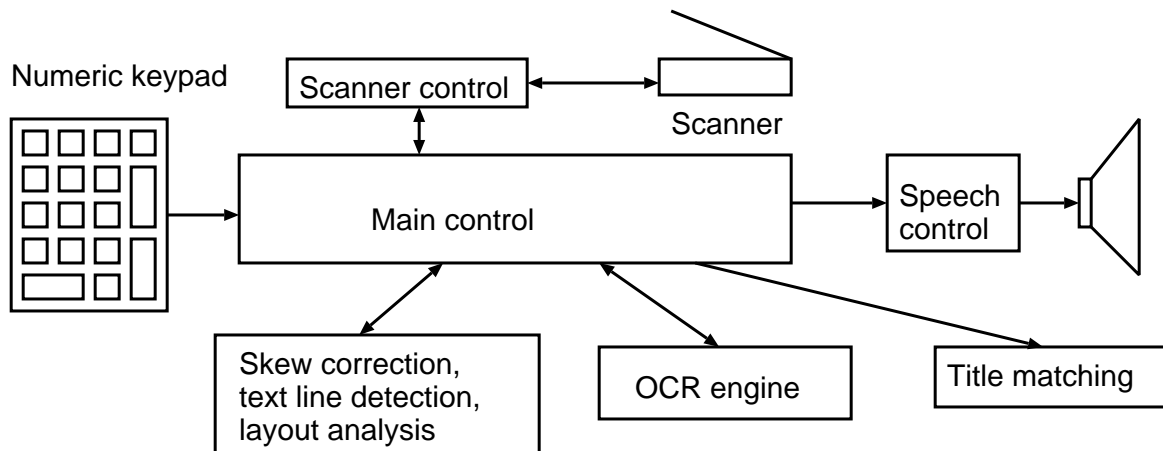


Figure 1: OMR System Configuration

### 3 Main Features and Functions

#### 3.1 Title Matching

To determine the name of the magazine, we provided a title-matching function. In many cases, the part of the cover page of the magazine that contains the logo is stable except as regards the color. This area is used to identify the magazine. The edges of the regions with the same color are detected and matched with previously collected cover page templates. Our preliminary experiment showed about 80% accuracy for 10 types of candidates. The name of the magazine as well as the confidence factor is reported to the user in synthesized speech.

#### 3.2 Skew Detection and Correction

To detect text lines, the OMR uses horizontal and vertical projections; it is therefore necessary to detect and correct skews in the scanned images. We used the Hough transform of the centers of connected components of the black pixels in the image [2, 3, 4]. The detection range is  $\pm 10$  degrees and the resolution is 0.1 degrees. The skew is determined for vertical and horizontal line candidates at the same time, and the original image is then rotated to the upright position.

#### 3.3 Text Area Analysis

Many magazines in the Japanese language have both vertical and horizontal text lines on the same page, making it necessary for us to detect lines in both directions. Our text line detection method works in two stages: first, it detects text areas, then, it determines the

orientation of text lines in the block –vertical or horizontal– by examining the arrangement of connected components within the block.

### 3.3.1 Text Area Detection

We first detect image areas in the page by using statistical features of connected components, namely, their widths, heights, and average run-length of black pixels.

In many magazines, images are not always placed in such a way that they occupy a single column width; often, they are spread across two or more columns. If the width of the image is not a multiple of the column width, the text lines are made to flow into the space formed by the difference between the column width and the picture width. This generates a concave text area, and causes the conventional text line detection method to fail. Our text line detection method is an extension of the recursive x-y cut [5], and can therefore handle concave text blocks.

We developed an L-shaped split method, which examines all the possibilities of separating text/image blocks by an L-shaped white space within a text/image block [Figure 2].

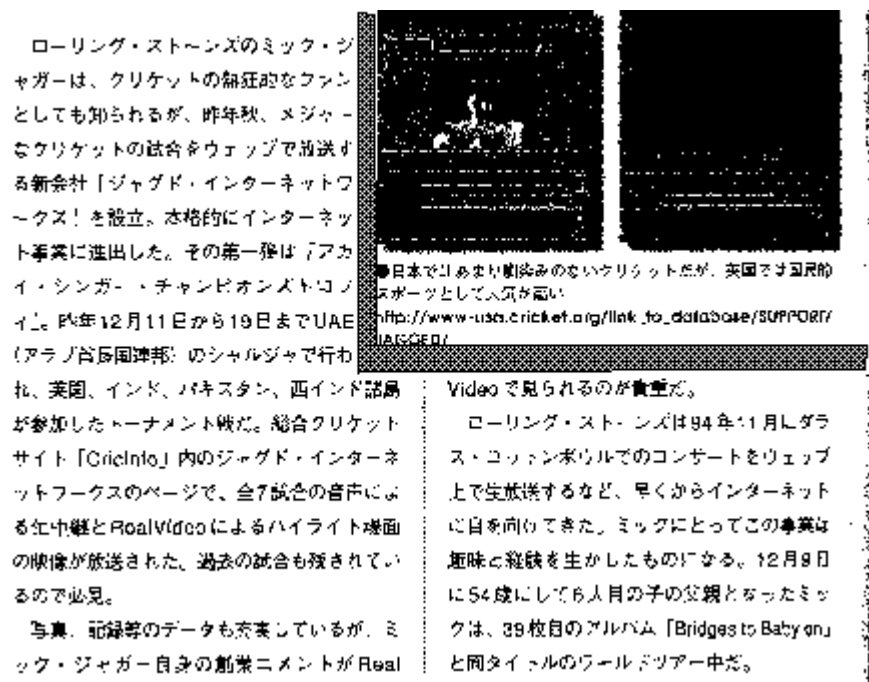


Figure 2: L-shaped Split Area (Hatched Area)

To find the best separation, we try four combinations of projections of the circumscribing rectangles of the connected components in the vertical and horizontal directions. If the upward and leftward projections, for example, of connected components leave some inner areas (rectangles) untouched, then each of them can be the corner of a separating L shape with downward and rightward edges [Figures 3, 4]. We choose the rectangle with the maximum area, and the L shape formed by moving it in the downward and rightward

directions is then the separation candidate for the combination of projection directions [Figure 5].

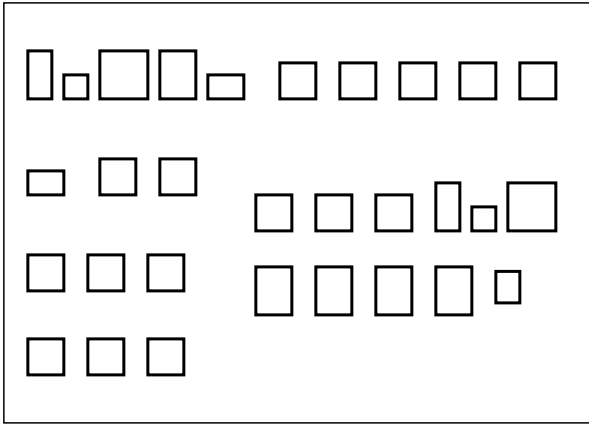


Figure 3: Circumscribing Rectangles of Connected Components

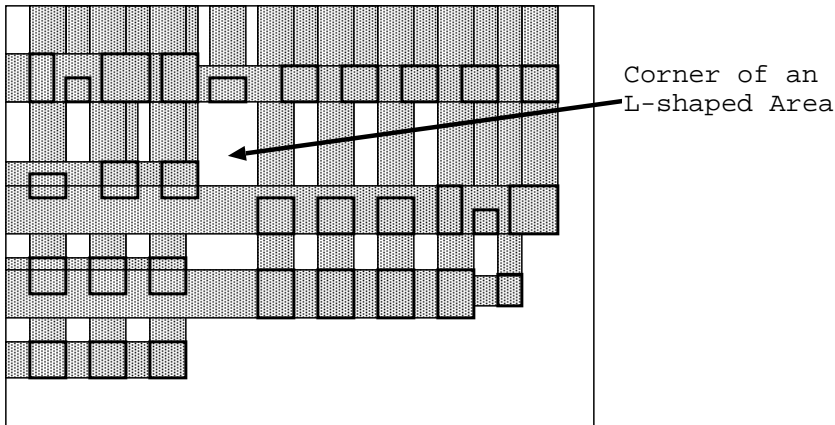


Figure 4: Upward and Leftward Projections and Inner Untouched Areas

The same procedure is repeated for the remaining three combinations of projection directions. The best separating L shape is selected and subsequent separations are then made. The L-shape split is iterated recursively in combination with ordinary x-y cuts.

### 3.3.2 Text Line Detection

To determine the text line orientation in a text area, we compute the orientation likelihood in two directions – vertical and horizontal – and then compare them to determine the orientation. For a given text area, we generate both vertical and horizontal text line candidate sets separately. Text lines are generated by projecting connected components in the vertical or horizontal direction, and examining their projection profiles.

Let  $T$  be a horizontal text line candidate. Let  $W$  be the width of  $T$ , and let  $W'$  be the sum of the widths of the connected components within  $T$ . For the height, we

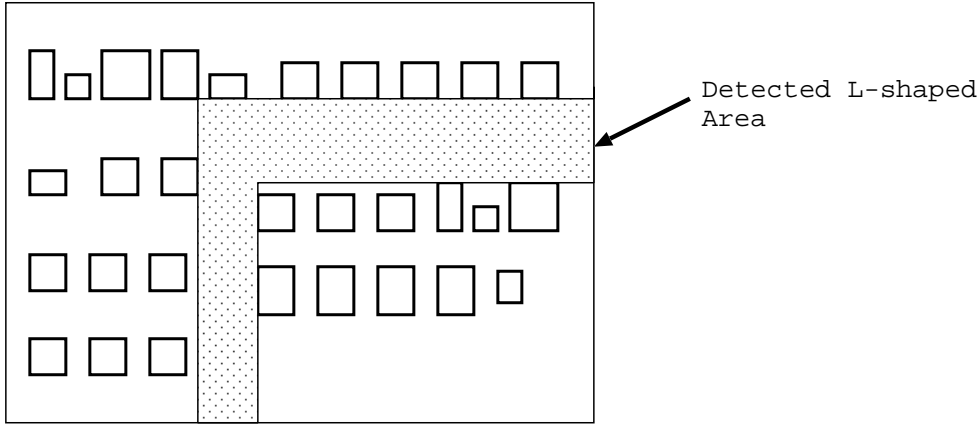


Figure 5: Detected L-shaped Area for Upward and Leftward Projections

define  $H$  and  $H'$  in the same way. Let  $n$  be the number of connected components within  $T$ . Since the text line is horizontal, we expect that  $R_{horz_H}(T) \equiv H'/(n \times H) \approx 1$  and  $R_{horz_W}(T) \equiv W'/W \approx 1$ . In the same way, if  $T$  is a vertical text line candidate, we expect that  $R_{vert_H}(T) \equiv H'/H \approx 1$  and  $R_{vert_W}(T) \equiv W'/(n \times W) \approx 1$ .

We calculate the sum of the squares of the logarithms of these ratios for text line candidates in each direction. Let  $T_{horz}$  and  $T_{vert}$  be the sets of horizontal and vertical text lines in a text area:

$$S_{horz} = \sum_{T \in T_{horz}} ((\log R_{horz_H}(T))^2 + (\log R_{horz_W}(T))^2)$$

$$S_{vert} = \sum_{T \in T_{vert}} ((\log R_{vert_H}(T))^2 + (\log R_{vert_W}(T))^2)$$

The smaller the sum, the more likely it is that the text area is oriented in that direction. This process identifies the text area orientation as well as the text line candidates.

### 3.4 Page Orientation Detection

The OCR engine we used cannot determine the orientation of the characters. It assumes that the image is in the right orientation and only returns the best matched characters or the list of the best candidates with matching scores. If the images are upside-down when input to the OCR engine, the output will be meaningless. OMR solves this problem by checking the OCR scores for a small number of text line candidates in both orientations. The orientation with better score is used for the subsequent full recognition.

The text lines are selected according to the following criteria:

1. Each line is longer than the average text length.
2. Each line has about the average character height.
3. The lines are selected from various parts of the page.

According to our preliminary experiment, about five text lines are sufficient to estimate the page orientation correctly in most cases.

### 3.5 Text Block Generation and Paragraph Connection

To provide quick access to the contents of the printed information on the scanned page, we extract the logical structure of the page and arrange the information according to that structure.

First, we extract the physical layout of the document by using recursive x-y cuts. Text blocks are obtained from this process. We then judge what is the most likely logical structure of the document on the basis of layout of the text lines.

The images of text lines are sent to the OCR engine and the recognition results are sent back to the system with recognition scores. The recognition results are processed together with the coordinates of the text lines to produce text blocks. Text blocks are determined by several features, such as the amount of space between two consecutive text lines, the existence of indentation, and the position of the end of the line relative to the text block.

If a text block consists of a small number of lines and its font size is larger than the average character size of the page, it is likely to be a heading. Other text blocks are paragraph segments. Paragraphs are often split by some image area or placed in different columns. To join these paragraph segments, we examine them with respect to the following features:

1. Differences of character size
2. Number of text lines in each paragraph segment
3. Position of the end of the line relative to the end of the paragraph segment area in the preceding paragraph segment
4. Depth of indentation in the succeeding paragraph segment

Hatched areas in Figure 6 show two text blocs, which are combined into one paragraph in the output shown in Figure 7.

In the output file, the text blocks are indicated by specifying the coordinates of the circumscribing rectangles (physical structure description). If several blocks are combined to form one paragraph, all of the blocks are enclosed by a pair of paragraph tags (logical structure description).

Table of contents pages are treated differently. We use the layout model to handle the special layout of such pages. Currently, our targets are limited to table of contents pages with horizontal text lines.





Figure 6: Detected Text Lines and Blocks (Hatched Areas Are Segments of Split Paragraph)

```

<p>
<textarea ul="(125,1915)" br="(711,2207)">
高野は昭和二四
年生まれで、この
とき四一ハ歳童在
は四七歳一/*一*/〇/*〇*/ホソ/*一*/ダ
の最年少役員であ
り、いわゆる団塊
世代一昭和二二年か
ら二四年に生まれ
た七〇〇/*〇*/万人の塊
だ〇/*〇*/昭和四七年に
</textarea>
<textarea ul="(2032,2286)" br="(2130,2893)">
東北大学工学部を卒業して、ホンダに
入社した〇/*〇*/
</textarea>
</p>

```

Figure 7: Output Text with Logical Tags and Coordinate Descriptions

### 3.6 Presentation of Recognition Results to Users

The recognition results for the text blocks, along with the extracted logical structures, are saved as a tagged document. The tags are defined so that they can convey the logical structures as well as physical information accompanying text blocks. Currently, we use tags for “headings,” “paragraphs,” “page numbers” and “coordinate information.”

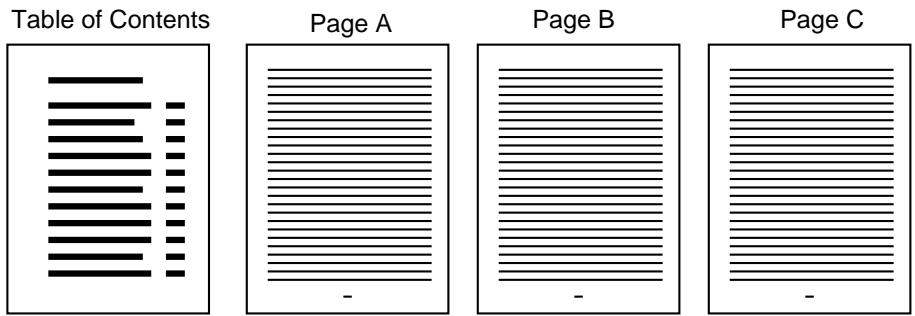
The recognized texts are read in synthesized speech. As a user interface, we provide logical tree-traversing commands for navigating through the recognized results, and a layered speech menu to help and guide users. All the commands are assigned to numerical keys in the numerical keypad area. The system has several operation modes such as “document reading,” “input device control,” and “speech setting control.” Each mode has its own key mapping to the numerical keys and the mapping itself is switched by pressing a specially assigned key (we use a key at the top of the numerical keypad area).

In the document reading mode, commands for descending and ascending trees are assigned to keys “6” and “4,” respectively, and commands for moving among siblings are assigned to keys “2” and “8.” Special functions such as “read from the current position to the end” or “read the page information (header or footer)” are assigned to the unused keys (“1” and “7”). The first position of the reading mode is at the beginning of the entire document, and the document’s title (file name) is spoken. The next layer consists of the document’s headings, if they exist. Users can navigate through the headings for items of interest. If a user finds an interesting item in a heading, he can read further by pressing the tree-descending key. The structure and the navigation method using it enable users to reach items of interest quickly.

## 4 Structured Document Management

Pages of a magazine are scanned, recognized, read, and saved if necessary. We use one directory for one volume of a magazine, so that its contents are handled as one entity. To facilitate the management of structured documents, we developed a function for automatically linking scanned pages on the basis of the page numbers extracted by the OMR. Table of contents (TOC) pages play an important role in the linking process described below. We assume the existence of only one TOC page in the following process:

1. Extract the correspondence between page numbers and title names from the TOC page file.
2. Extract the page numbers from the ordinary pages.
3. Create links between the page numbers and title names of TOC pages and the corresponding page files.
4. Make a sorted list of page numbers and create links based on the page numbers in ascending and descending orders.
5. In each ordinary page file, create a link to the TOC page file at the end of the file.



Link Generation  
by Page Numbers

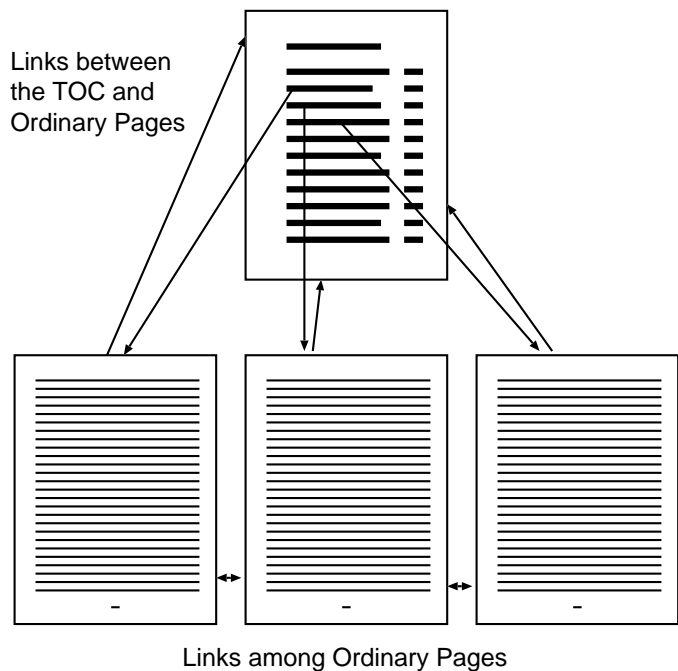
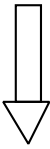


Figure 8: Linking Pages

Figure 8 shows an example of a linked result.

## 5 Evaluation

### 5.1 Evaluation of Text Area Discrimination Function

We evaluated the accuracy of the text area detection function. The criteria of the evaluation are the ratio of undetected, added, or deformed text areas against the correct data marked manually, and the correctness of the orientation of the text areas. We counted only those areas output by the layout analysis module as probable 'text areas' and processed properly by the OCR engine as actual text areas. Areas with bad OCR scores were excluded, because they would not be presented to the user.

The discrimination results are shown in the Table 1. The ratio of misidentified blocks were 18% of the total number of blocks but only 2% by area size.

Table 1: Results of Layout Analysis

Item	No. of Blocks	Del.	Add.	Deform.	Orientation Error	Total
number	1,084	9	158	21	3	191
ratio (%)	100	0.8	14.5	0.19	0.3	17.6

As to the paragraph connection accuracy, the spurious splits were caused mostly where the image areas were inset into text columns. The total error rate on the paragraph connections was nearly 8%, which can be reduced by using the recognition results of the OCR engine.

Table 2: Results of Paragraph Connection

	No. of Blocks	No. of Connections	Spurious Connections	Spurious Splits
number	1441	241	36	78
ratio (%)	100	16.7	2.6	5.4

### 5.2 User Evaluation

Evaluation of the OMR was done by two visually disabled users. After some introductory explanation, both were able to operate the system by themselves, and access the contents of the magazines. As to the mode of access to the information, access by using the logical structure was quick but caused the users to feel insecure because of omissions or skips due to mis-recognized logical structures.

## 6 Conclusion

We have built a document reading system for the visually disabled. To cope with the slow access speed resulting from the use of synthesized speech, we extract the logical structure from a scanned document and use it to provide quick access to the contents. The accuracy with which the logical structure is extracted plays a crucial role in determining the efficiency of our system. We need to enhance the extraction accuracy further. The scanned pages are saved and linked with the TOC page to form an organized tagged document, which enables the user to access them efficiently in spoken form. We plan to improve our system to a practical level.

### Acknowledgments

This work was supported by the New Energy and Industrial Technology Development Organization.

## References

- [1] Kazuhide Sugawara, "Document Reading System for the Visually Disabled," in Proc. IEEE Multimedia Systems '99, Vol.2, pp. 985-986, 1999.
- [2] Y. Nakano, Y. Shima, H. Fujisawa, J. Higashino, and M. Fujinawa, "An Algorithm for the Skew Normalization of Document Images," in Proc. ICPR '90, pp. 8-11, 1990.
- [3] S. C. Hinds, J. L. Fisher, and D. P. D'Amato, "A Document Skew Detection Method Using Run-Length Encoding and the Hough Transform," in Proc. ICPR '90, pp. 464-468, 1990.
- [4] Kazuhide Sugawara, "Weighted Hough Transform on a Gridded Image Plane," in Proc. ICDAR '97, pp. 701-704, 1997.
- [5] Jaekyu Ha, Robert M. Haralick, Ihsin T. Phillips, "Document Page Decomposition by the Bounding Box Projection Technique," in Proc. ICDAR '95, pp. 1119-1122, 1995.