Document Skew Angle Detection Algorithm

Daniel X. Le, George R. Thoma

Proc. 1993 SPIE Symposium on Aerospace and Remote Sensing -
Visual Information Processing II, Orlando, FL
April 14-16, 1993
Vol. 1961, pp. 251-262.

Abstract

As part of research into document image processing, an algorithm has been developed to detect the degree of skew in a scanned binary image. The principal components of the algorithm are component labelling, a procedure to reduce the amount of data to be processed, a technique to minimize the effect of non-textual data (graphics, forms, line art, large fonts, and dithered images) on the measurement of skew angle, and the Hough transform. The performance of the algorithm has been evaluated using a sample size of several hundred images of medical journal pages. Evaluation shows that a skew angle may be detected with an accuracy of about 0.50 degree.

1. INTRODUCTION

The conversion of paper-based documents to electronic image format is important in systems for automated document delivery, document preservation and other applications. Document conversion includes scanning, displaying, quality assurance, image processing, text recognition, and creating image and text databases. During the document scanning process, the whole document or a portion of it can be fed through the looseleaf page scanner. Some pages may not be fed straight into the scanner, however, causing skewing of the bitmapped-images of these pages; these pages may eventually be identified and rescanned by a quality control operator. In an attempt to partially automate the quality assurance process as well as to improve the text recognition process, a document skew angle detection algorithm has been developed.

Previous work done to detect document skew angle have been reported in the literature.[1,3,5,10] Techniques implemented by Fletcher and Kasturi[3] and by Rastogi and Srihari[10] have been reviewed by Hinds, Fisher, & D'Amato.[5] The Fletcher and Kasturi[3] algorithm is robust and reliable, but is slow due to the use of a connected components analysis on the original image. The Rastogi and Srihari[10] algorithm gives good results but is also slow because of the excessive data required to be processed by the Hough transform. Baird[1] proposed an algorithm to determine skew angle using an energy function on sets of projection-counts of character locations. In this algorithm, a character is represented by the midpoint of the bottom bounding box that will be projected into an accumulator line $C(\theta)$ for each angle θ and $C(\theta)$ then will be partitioned into m bins $c^i(\theta)$, where i = 1,..,m. An energy function $A(\theta)$ is computed as the sum of m bins $c^i(\theta)$ squared and the global maximum of this energy function determines the skew angle. While this technique works well on a wide variety of portrait mode layouts, there is no discussion of its suitability to pages in landscape mode. Hinds, Fisher, & D'Amato[5] proposed an algorithm that uses the Hough transform to determine a document skew angle. To reduce the amount of data that is input to the Hough transform, a "burst image" is created by placing the length of the vertical run in its bottom-most pixel. The skew angle will then be determined by applying the Hough transform on the "burst image". While the technique correctly determines document skew angle on a variety of images, the data reduction factor is still small: 2 for a typical text document and 11 for a mixture of pictures and text document. As a result, the resolution of the document image has to be reduced before creating its "burst image" in order to increase the speed.

The document skew angle detection algorithm proposed in this paper is based on component labelling, a procedure to reduce the amount of data to be processed, a technique to minimize the effect of non-textual data (graphics, forms, line art, large fonts, and dithered images) on the measurement of skew angle, and the Hough transform. The algorithm is characterized by the following features: (1) it uses the bottom part of objects (characters); (2) the data necessary for skew angle computation is reduced by a factor of around 15 for a typical text page and more than 80 for a mixture of pictures and text page; (3) the detection process can be running while an image is scanned; (4) it is independent of text dominance.

The term "binary image" refers to a two-dimensional binary function $f(x,y)$, whose value is either 0 or 1, where x and y denote vertical and horizontal coordinates of the pixel respectively, and where the coordinates origin is located at the top left corner of the image. The maximum dimensions of the image are xmax and ymax, where xmax and ymax are the number of rows and the number of columns of a binary image, respectively. Figure 1 shows an example of a binary function $f(x,y)$, its dimensions and axes. Also, throughout this paper, a string of dots (...) represents a white run and a string of symbols x (xxxx) represents a black run.

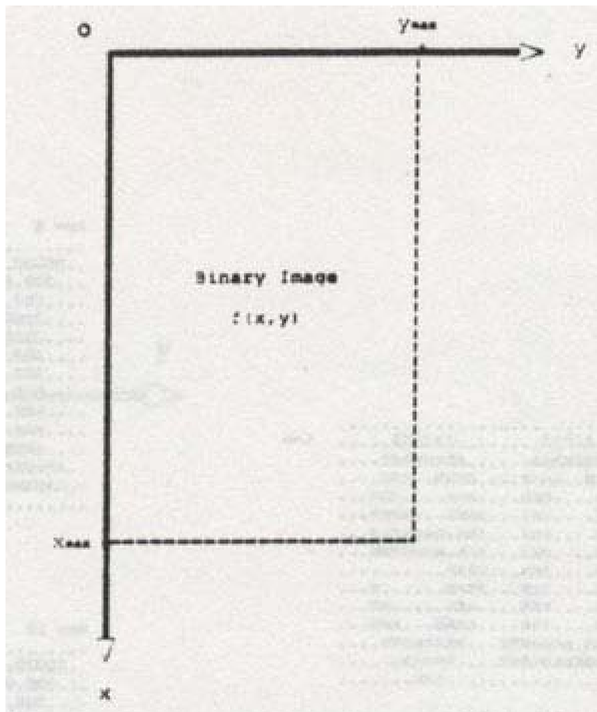


Figure 1 A binary image f(x,y)

Figure 2 A square portion of a skewed textual data binary image

2. DOCUMENT SKEW ANGLE DETECTION APPROACH

The process may be described by starting with a square textual portion of a skewed binary image as shown in Figure 2. This image is simplified as shown in Figure 3 from by removing all black pixels except those belonging to the last black run of each object. Figure 3 shows that the remaining pixels are almost all oriented in the same direction. Consequently, the skew angle of a binary image may be detected by applying the Hough transform on this simplified binary image. However, take a look at a non-textual (dithered) square portion of another skewed binary image shown in Figure 4. Its simplified version shown in Figure 5 contains scattered pixels, making skew angle detection difficult. Non-textual data is a major source of skew angle error and it is important to minimize its influence in the skew angle detection process, as done in the present work.

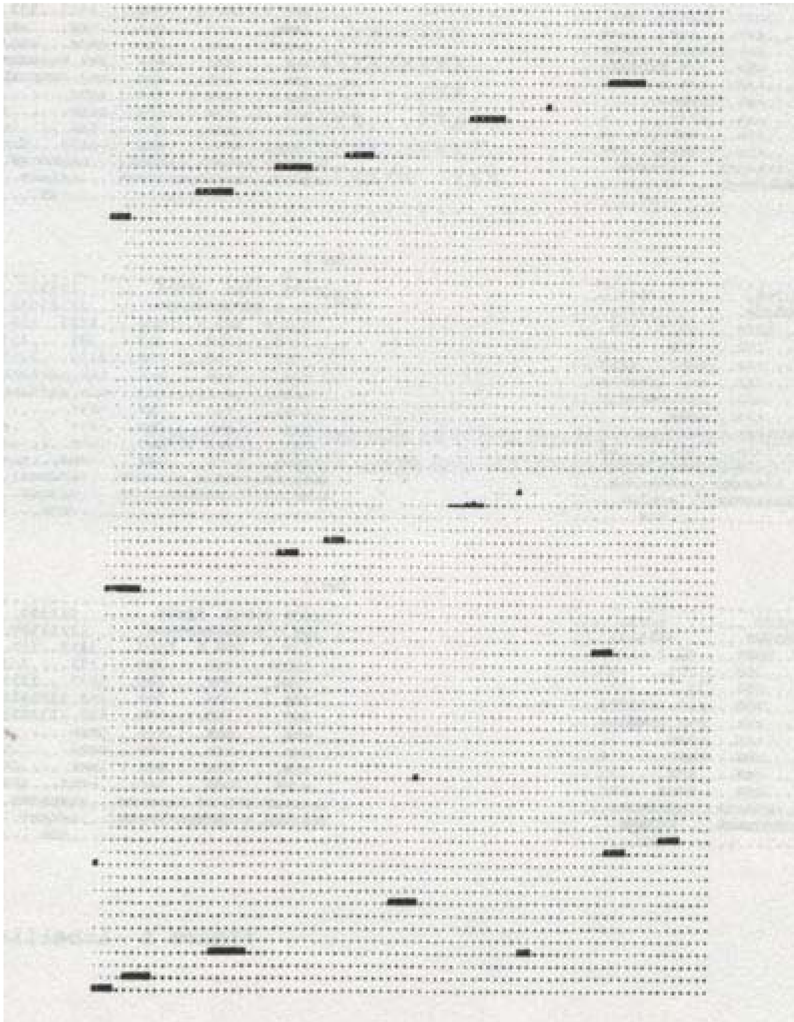


Figure 3 A simplified image of the square binary image in Figure 2

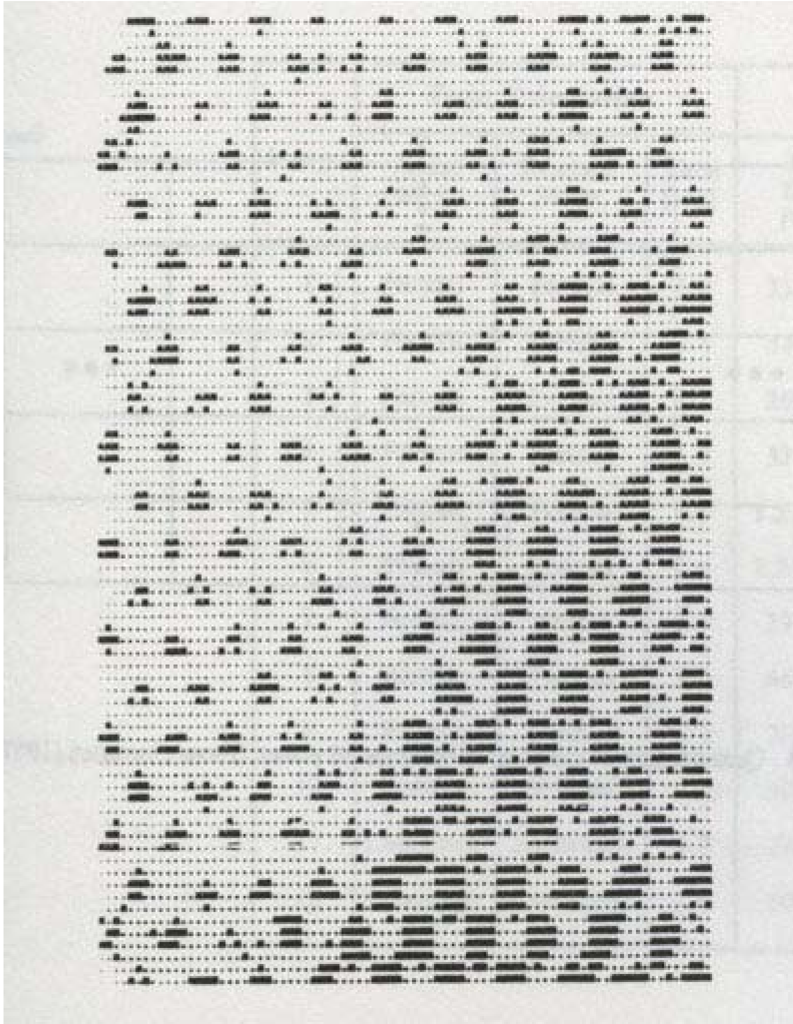


Figure 4 A square portion of a skewed dithering data binary image

In this paper, the algorithm proposed is based on the processing of pixels of the last black runs of objects. The algorithm segments a binary image into objects, creates a simplified binary image from the original binary image, minimizes the skew angle error due to non-textual data, and detects the skew angle. The algorithm consists of component labelling, the Hough transform, a data reduction technique, and a procedure to minimize the effect of non-textual data on the measurement of skew angle. In addition, this approach works only on a portrait mode binary image, it is required to know beforehand the page orientation to adjust the binary image into the appropriate mode before applying the skew angle detection algorithm. An algorithm for detecting a page orientation has also been proposed by us[7]. In the following subsections, each component of the document skew angle detection algorithm will be discussed in detail.

2.1 Component labelling

The purpose of component labelling is to segment a binary image by labelling its objects.[6] As used in this paper, an object in a binary image is defined as a collection of black runs that are 8-connected.[4] During the component labelling process, each black run is assigned an integer number called "label" and the labels of connected black runs must be the same. An object is created from connected black runs and its label is the same as that of its black runs. One object is separated from another object based on its label. The component labelling algorithm identifies black runs for each row, analyzes the connectivity of black runs between the current row and the previous row, assigns labels to black runs of the current row and/or updates labels of black runs of the previous rows, and then identifies objects of a binary image.

The component labelling algorithm is given below:

```
(I) For each row of a binary image,
      (A) Identify its black runs
      (B) If this is the first row of an
        image or if the previous row is a white run
        then assign a new and different label for
        each black run. Otherwise, for each black run
        of the current row do the followings:
        (1) If it is not connected to any black runs of
        the previous row then assign it a new label.
        (2) If it is connected to a black run of the
        previous row for the first time then assign to
        it the label of this previous row black run.
        (3) If it is connected, but not for the first
        time, to a black run of the previous row then uses
        its label to update the label of (a) this previous
        row black run and (b) all connected black
        runs of this previous row black run.
(II) Finally, identify each object of a binary image based on
 its black run label.
```

Figure 6 shows an example of labelling objects in a binary image using the component labelling algorithm. In this example, a string of integer symbols represents the label of the corresponding black run. The example in Figure 6 shows changes in labels of objects after processing each row in a binary image. It is noted that the row being processed is pointed to by an arrow. In row 1, four black runs are labeled with 0, 1, 2 and 3 because the previous row 0 is a white run. Now look at four black runs of the row 2 of which the first black run is labeled with 0 because it connects to the first black run labeled with 0 of the previous row 1. The second black run of the row 2 is labeled with 0 because it connects to the first black run labeled with 0 of the previous row 1. Because it also connects to the second black run originally labeled 1 of the previous row 1, the second black run of the previous row 1 is changed its label to 0. The third black run of the row 2 is labeled with 0 and it also makes the third black run of the previous row 1 change its label from 2 to 0. The fourth black run of the row 2 is labeled with 3. Consider the fourth black run of the row 12 of which the first three black runs are labeled with 0. In the first step, the fourth black run of the row 12 is labeled with 3 because it connects to the fourth black run labeled with 3 of the previous row 11. In the second step, because it also connects to the fifth black run labeled with 4 of the previous row 11, the fifth black run of the previous row 11 and all black runs connected to it - the black runs labeled 4 in the rows 9 and 10 - have their labels changed from 4 to 3.

2.2 The Hough transform

In this paper, the Hough transform is used to detect a straight line in a binary image. A representation of a straight line in the polar coordinate plane ρθ can be described via the equation[4]:

$$\rho = x \cdot \cos\theta + y \cdot \sin\theta \qquad (1)$$

As shown in Figure 7, is the normal distance OH from the origin O to the line and θ is the angle between the x axis and OH. The above equation describes a mapping of a point in the Cartesian coordinate xy plane to a sinusoidal curve in the polar coordinate ρθ plane.

Consider N points that are lying on a line $\rho o = x \cdot \cos\theta o + y \cdot \sin\theta o$. The Hough transform maps N points into N sinusoidal curves that cross at a point (ρo, θo) in the plane. It can also be said that the intersection point (ρo, θo) of N sinusoidal curves denotes a line in the xy plane corresponding to (ρo, θo) passing through these N points.

Given that the $\rho\theta$ plane is quantized into cells as shown in Figure 8 an accumulator array A(ρi, θi) is created to keep track of the number of intersections of sinusoidal curves at a cell (ρi, θi). When the mapping is done for all data points of an image, an accumulated value in A(ρi, θi) corresponds to a number of points in a corresponding line in the xy plane.[9]

The line detection in a binary image using the Hough transform algorithm[2] can be summarized as follows:

```
(1) Define the Hough transform parameters ρmin, ρmax, θmin and θmax.
(2) Quantize the ρθ plane into cells by forming an accumulator
cell array A (ρθ), where ρ is between ρmin and ρmax, and θis
between θmin and θmax.
(3) Initialize each element of an accumulator cell array A to
zero.
(4) For each black pixel in a binary image, perform the
following:
       (a) For each value of θi from min to max, calculate
       the corresponding ρi using the following equation:
         x . cosθi + y . sinθi  =  ρi
       (b) Round off the ρi value to the nearest allowed ρ
       value.
       (c) Increment the accumulator array element A (ρi,θi).
(5) Finally, local maxima in the accumulator cell array
correspond to a number of points lying in a corresponding
line in the binary image.
```

As stated in the Hough transform algorithm, the Hough transform parameters ρmin, ρmax, θmin and θmax have to be defined before any processing can occur. Consider a binary image f(x, y) as shown in Figure 9, where x = [0, xmax] and y [0, ymax]. Using xmax and ymax, the values of ρmin and ρmax can be calculated from θmin and θmax and vice versa.

The computational performance of the Hough transform algorithm can be determined by analyzing the step (4) above. If N is the number of black pixels in a binary image and K is the number of increments of θi from θmin to θmax, number of computations in the Hough transform is 3NK.

To increase the speed of the Hough transform, the amount of data points to be processed in a binary image should be reduced. In this paper, a data reduction procedure proposed to achieve this goal is discussed in the next section.

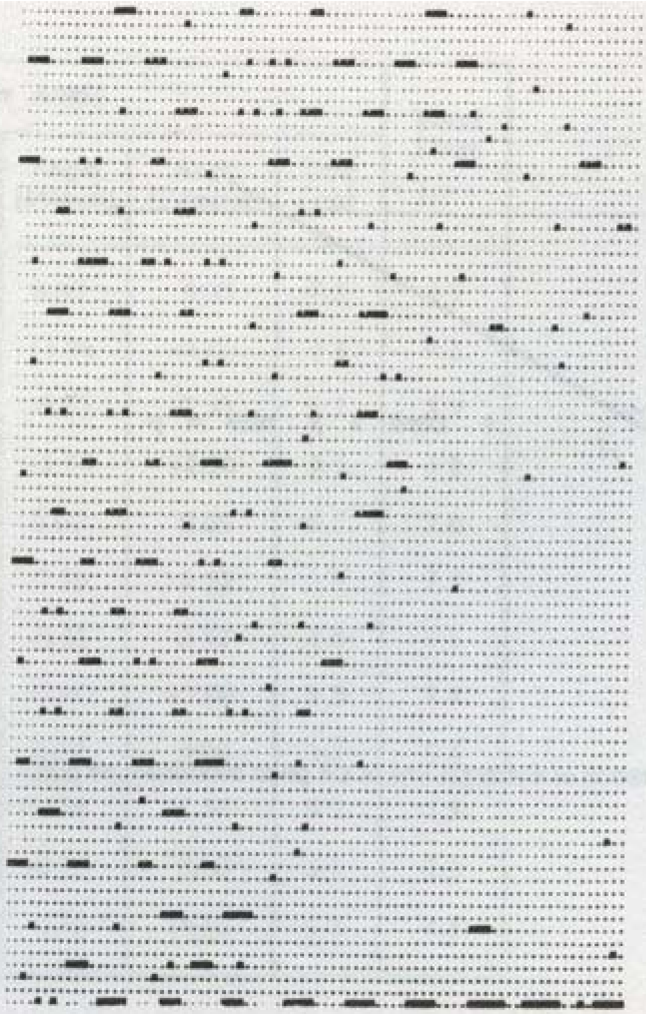


Figure 5 A simplified image of the square binary image in Figure 4

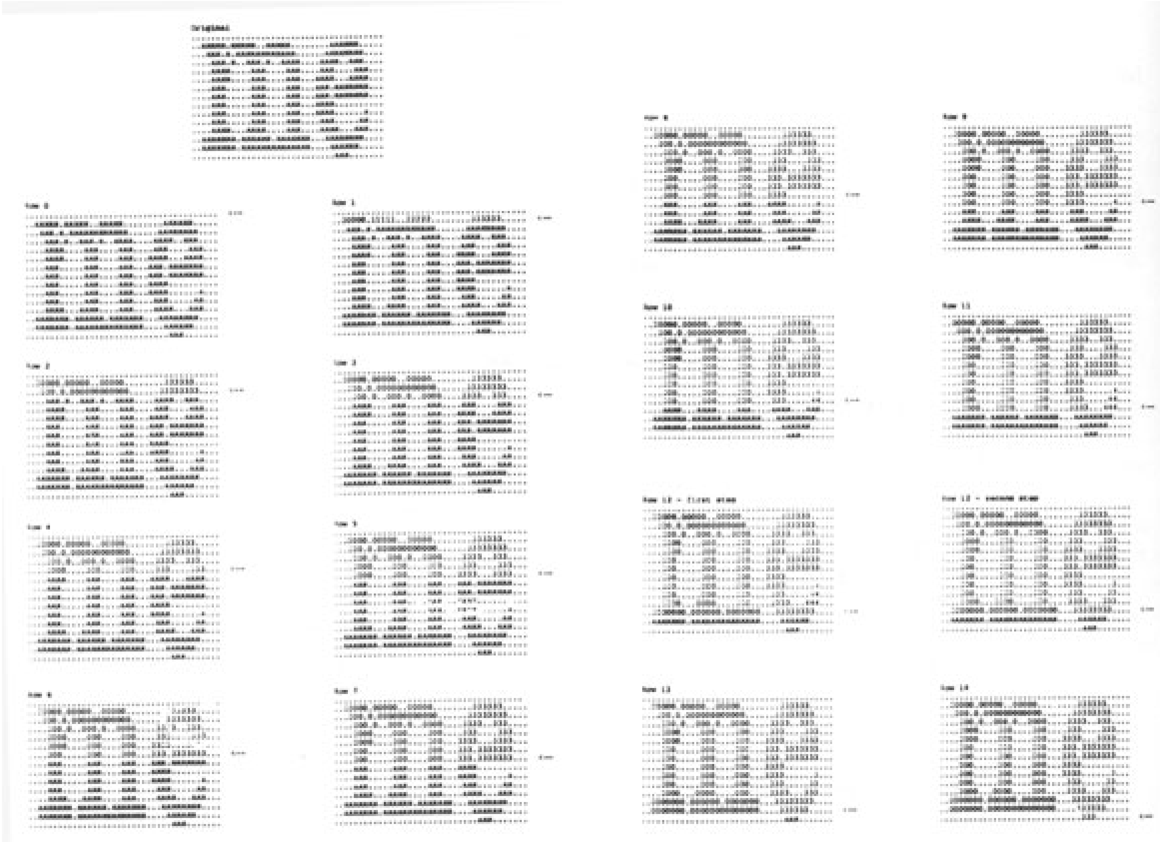


Figure 6 Labelling objects in a binary image.

2.3 Data reduction by preserving object bottom pixels

The purpose of this procedure is: (1) to extract features from a binary image that are used to detect the document skew angle and (2) to speed up the Hough transform. The extracted features are called "objects bottom pixels" which are black pixels belonged to the last black runs in the bottom row of the object. Figure 10a shows an object of a binary image and Figure 10b shows its object bottom pixels. These black pixels belong to the last two black runs in row 9.

To achieve these two goals, the data reduction procedure creates a simplified binary image from an original binary image. The simplified binary image consists only of bottom black pixels of objects in an original binary image. After a binary image is segmented into objects by the component labelling algorithm mentioned in the section 2.1, a simplified binary image can be created from an original binary image by removing all black pixels except objects bottom pixels.

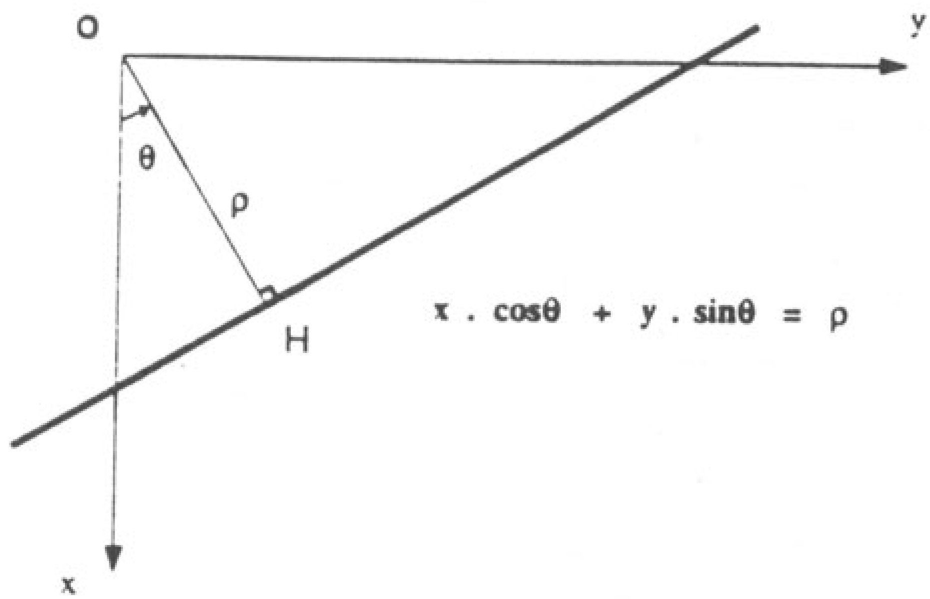


Figure 7 Representation of a straight line in the polar coordinate ρθ plane

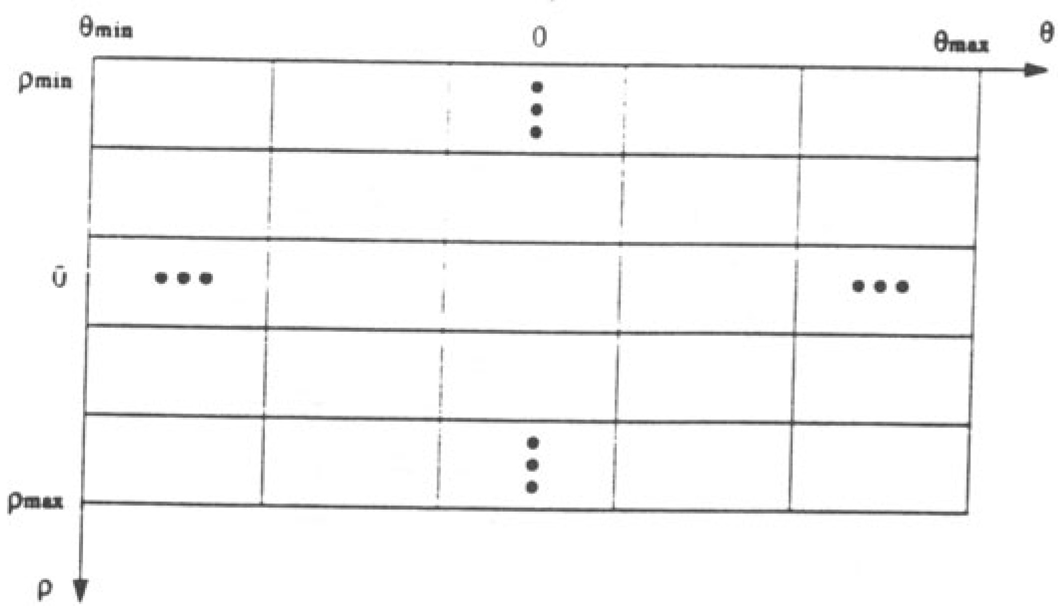
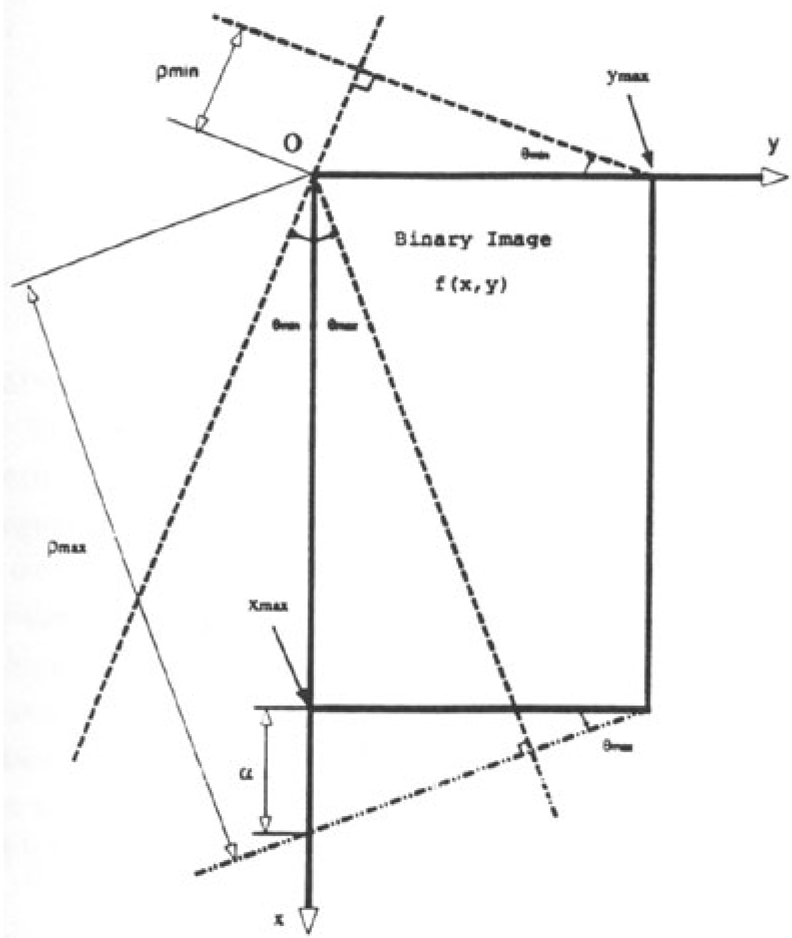


Figure 8 Quantization of the polar coordinate ρθ plane. (From Gonzalez [1987])

2.4 Skew angle error minimization procedure

After the simplified binary image is generated using the data reduction technique of section 2.3, the Hough transform can be applied on this simplified binary image to get the document skew angle. The accuracy of the skew angle is dependent on the contents of the simplified binary image which can be influenced by non-textual data such as graphics, forms, large fonts, line art, or dithered images.

In this section, a simple procedure is proposed to minimize the skew angle detection error due to non-textual data. The procedure is based on the analysis of an object feature called "object extreme coordinates" which provides information about the approximate size of an object in a binary image.



$$\alpha = y_{max} \,.\, \tan\theta_{max}$$
$$\rho_{min} = y_{max} \,.\, \sin\theta_{min}$$
$$\rho_{max} = (\alpha + x_{max}) \,.\, \cos\theta_{max}$$

Figure 9 Hough transform parameters in the polar coordinate ρθ plane.

```
Row 0    .................      .................
    1    ......XXXXX......      .................
    2    .....XX..XXX.....      .................
    3    ..........XXX....      .................
    4    ......XXXXXXX....      .................
    5    ....XXXXXXXXX....      .................
    6    ...XXXX...XXX....      .................
    7    ...XXXX...XXX....      .................
    8    ...XXXXXXXXXXX...      .................
    9    .....XXXXX..XXX..      .....XXXXX..XXX..
    10   .................      .................

               (a)                     (b)
```

Figure 10 An example of object bottom pixels: (a) original binary image and (b) image object bottom pixels.

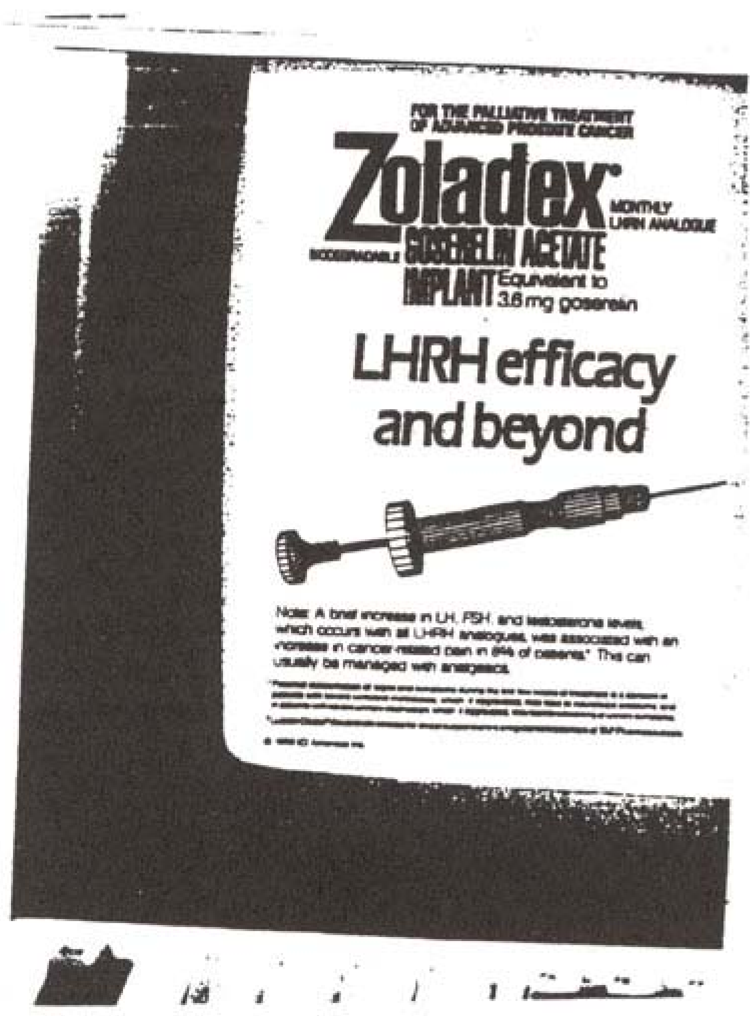


Figure 11

Figure 12 A simplified image of Figure 11

Eicosanoids

Preparation of leukotriene B_4 antibodies in sheep

Materials and methods

Introduction

Figure 13

Figure 14 A simplified image of Figure 13

Each object in a binary image has its own extreme coordinates with respect to the origin of the binary image *f*(x,y):
min_left_column, max_right_column, min_top_row and max_bottom_row. The definitions of these extreme coordinates are as follows:

```
min_left_column:     the column value of the left-most pixel of an object
max_right_column:    the column value of the right-most pixel of an object
min_top_row:         the row value of the top-most pixel of an object
max_bottom_row:      the row value of the bottom-most pixel of an object
```

As shown above, objects bottom pixels of a textual data binary image are almost all oriented in one direction and they can be used effectively to calculate the skew angle of the binary image. In contrast to the textual data binary image, no information can be drawn from objects bottom pixels of a non-textual data binary image as shown in Figures 4 and 5 because pixels scatter randomly around the image. Therefore, using non-textual data to detect skew angle may lead to unexpected errors.

In order to minimize the skew angle detection error, the proposed procedure redefines the simplified binary image as follows: A simplified binary image is created from an original binary image by preserving only bottom pixels of "allowable" objects.

Let "diff_col" and "diff_row" be "max_right_column - min_left_column + 1" and "max_bottom_row - min_top_row + 1" respectively. An object is allowable if it satisfies all of the following conditions:

```
(1) MIN_OBJECT_WIDTH      <  diff_col  < MAX_OBJECT_WIDTH
(2) MIN_OBJECT_HEIGHT     <  diff_row  < MAX_OBJECT_HEIGHT
(3) MIN_OBJECT_AREA       <  diff_row . diff_col  < MAX_OBJECT_AREA
```

The values of the following predefined parameters MIN_OBJECT_WIDTH, MAX_OBJECT_WIDTH, MIN_OBJECT_HEIGHT, MAX_OBJECT_HEIGHT, MIN_OBJECT_AREA, and MAX_OBJECT_AREA will be defined in section 4.

3. DOCUMENT SKEW ANGLE DETECTION ALGORITHM

As mentioned in the previous sections, since the document skew angle detection algorithm works only on a portrait mode binary image, knowledge of the page orientation[7] is required to adjust the binary image into the appropriate mode before starting the skew angle detection process. After adjusting the binary image accordingly, the algorithm creates a simplified binary image from the original binary image and applies the Hough transform on the simplified binary image to get the document skew angle. The document skew angle detection algorithm can be summarized as follows:

(1) Adjust a binary image $f(x,y)$ into portrait mode, if necessary.
(2) For each row of the binary image $f(x,y)$, generate and label its black runs.
(3) Build objects based on black runs of the same labels and update extreme coordinates of objects.
(4) Create a simplified binary image by preserving the last black runs of each "allowable" object.
(5) Apply the Hough transform on the simplified binary image.
(6) Analyze the local maxima of the Hough accumulator cell array to detect the skew angle of the binary image as follows:
(a) Collect the first and second maxima of Hough accumulator cell array elements.
(b) Collect all Hough accumulator cell array elements of which the values are greater than one-half of the second maxima of Hough accumulator cell element.
(c) Add these values together based on their angle θ.
(d) The skew angle is the angle corresponding to the maximum of these values.

4. PARAMETERS OF THE DOCUMENT SKEW ANGLE DETECTION ALGORITHM

In this section, all predefined parameters of the document skew angle detection algorithm will be defined and whenever possible formulas will be derived. Let dpi (dots per inch) represent the scanning resolution. Details of the parameter values and formulas are given in reference [8].

Parameters	**Value/Formula**	**For 200 dpi**
MAX_OBJECT_HEIGHT	15 · dpi/72 pixels	42 pixels
MAX_OBJECT_WIDTH	15 · dpi/72 pixels	42 pixels
MAX+OBJECT_AREA	$[15 \cdot dpi/72]^2$ $pixels^2$	1764 $pixels^2$
MIN_OBJECT_HEIGHT	1 pixel	1 pixel
MIN_OBJECT_WIDTH	1 pixel	1 pixel
MIN_OBJECT_AREA	4 $pixels^2$	4 $pixels^2$

5. EXPERIMENTAL RESULTS

There are two sets of test images used in the experiment: (1) the first set consists of 12 selected binary images which represent both non-textual and textual data pages, and (2) the second set consists of 238 pages from 3 different medical journals. Details of the results are given in reference [8]. The document skew angle algorithm is running on a Sun SPARC 10, Model 30 computer.

All binary images are scanned at 200 dpi resolution and they are 8.5 x 11 inches in size. It is assumed that the document skew angle with respect to the vertical axis x will be limited within [-15, +15] degrees. Therefore, the values of θmin and θmax with respect to the vertical axis x is chosen to be -15 degrees and +15 degrees, respectively. Also, the angle θi is incremented by 0.5 degree.

The summarized results of the test sample set 1 are tabulated in Table 1. The skew angle of each binary image is also measured manually and tabulated to test for the accuracy of the algorithm. Two examples from the 12 selected binary images and their simplified versions are presented in Figures 11 to 14 to show that most of the non-textual data from the original binary images is removed.

The reduction factor, which is the ratio between the total black pixels of the original binary image and that of its simplified binary image, in Table 1 shows that the speed of the Hough transform is increased by more than 18 times. In one case, the speed performance is improved by a factor 167 times. The average time to process a scanned binary image is about 6 seconds. Finally, the manually measured skew angles and the detected skew angles in Table 1 show that the document skew angle of a binary image can be detected with an accuracy of about 0.50 degrees.

Image	Page Orientation			Document Skew Angle						Total Time [sec]
	Actual Mode	Detected Mode	Time [sec]	Total Black Pixels	Objects Bottom Pixels	Reduction Factor	Angle [Manual]	Angle [Detected]	Time [sec]	
1	Portrait	Portrait	1.4	353,608	15,191	23	0.00	0.00	7.5	8.9
2	Portrait	Portrait	2.4	543,946	13,812	39	+3.86	+4.00	7.4	9.8
3	Portrait	Portrait	1.7	269,555	9,210	29	-0.20	-0.50	4.2	5.9
4	Portrait	Portrait	1.9	321,927	7,104	45	+2.80	+2.50	3.3	5.2
5	Portrait	Portrait	1.5	1,215,375	7,664	158	0.00	0.00	6.1	7.6
6	Portrait	Portrait	1.6	1,248,289	7,472	167	-2.56	-2.50	5.7	7.3
7	Portrait	Portrait	2.7	395,251	15,233	26	-0.35	-0.50	4.4	7.1
8	Portrait	Portrait	2.6	666,976	12,146	55	+7.40	+7.00	4.5	7.1
9	Portrait	Portrait	2.2	301,558	9,063	33	0.00	0.00	3.3	5.5
10	Portrait	Portrait	2.3	302,477	8,263	36	-1.38	-1.50	3.2	5.5
11	Landscape	Landscape	1.7	196,902	10,685	18	-0.16	0.00	2.6	4.3
12	Landscape	Landscape	1.6	193,398	8,430	23	-4.59	-4.50	2.7	4.3

Table 1 The first test set

6. CONCLUSION

An algorithm for document skew angle detection employing the Hough transform has been presented. However, the amount of computations in the Hough transform is linearly dependent on the number of data points to be processed. Also, the non-textual data in a binary image affects the accuracy of the skew angle.

A data reduction procedure, which creates a simplified binary image from the original binary image by preserving bottom pixels of "allowable" objects, has been implemented to improve the speed performance of the Hough transform as well as to reduce the impact of the non-textual data on the skew angle results. The algorithm has been tested on several hundred images of medical journal pages and it has successfully detected the document skew angles with an accuracy of about 0.50 degrees.

Although in this work the skew angle was correctly determined in all cases and the speed of the Hough transform has been improved, it is noted that the algorithm processes the entire binary image to determine its skew angle. Actually, the running time of the algorithm can be reduced more if it only processes text regions of animage. Fortunately, the page orientation algorithm proposed by us[7] can roughly classify textual and non-textual regions. A future task is to improve the skew angle detection running time by incorporating the page orientation algorithm with the document skew angle detection algorithm.

7. ACKNOWLEDGEMENTS

The authors thank Lewis Berman for his valuable suggestions, for reading the manuscript and providing important feedback. We also thank our other colleagues at NLM, Dr. Susan Hauser, Frank Walker, Rodney Long, and Mike Gill for their help.

8. REFERENCES

1. S. Baird, "The skew angle of printed documents," Proc. of Society of Photographic Scientists and Engineers, vol. 40, pp. 21-24, 1987.
2. D. H. Ballard and C.M. Brown, "Computer Vision," Prentice Hall, Englewood Cliffs, New Jersey, pp. 123-124, 1982.
3. L. A. Fletcher and R. Kasturi, "A robust algorithm for text string separation from mixed text/graphics images," IEEE Trans. on PAMI, vol 10, pp. 910-918, Nov. 1988.
4. R. C. Gonzalez and P. Wintz, "Digital Image Processing," 2nd ed., Addison-Wesley, Reading, Massachusetts, pp. 30-31 and 130-134, 1987.
5. S. C. Hinds, J. L. Fisher and D. P. D'Amato, "A Document Skew Detection Method Using Run-Length Encoding and the Hough Transform," 10th International Conference on Pattern Recognition, vol. 1, pp. 464-468, 1990.
6. A. K. Jain, "Fundamentals of Digital Image Processing," Prentice Hall, Englewood Cliffs, New Jersey, pp. 409-410, 1989.
7. D. X. Le and G. R. Thoma, "Automated portrait/landscape mode detection on a binary image," To appear in SPIE's 1993 Symposium on Aerospace and Remote Sensing.
8. D. X. Le, "Automated Document Skew Angle Detection Using Projection Profiles, Variances, Component Labelling and the Hough Transform," M.S. thesis, Computer Science Department, George Mason University, November 17th, 1992.
9. W. K. Pratt, "Digital Image Processing," A Wiley-Interscience publication, New York, pp. 523-525, 1978.
10. A. Rastogi and S. N. Srihari, "Recognizing textual blocks in document images using the Hough transform," TR 86-01, Dept. of Computer Science, SUNY Buffalo, NY, Jan 1986.