IJDAR (1998) 1: 43–51

International Journal on Document Analysis and Recognition IJDAR
© Springer-Verlag 1998

Projection profile based skew estimation algorithm for JBIG compressed images

Junichi Kanai, Andrew D. Bagdanov

Information Science Research Institute, University of Nevada, Las Vegas, 4505 Maryland Parkway, Las Vegas, NV 89154-4021, USA; e-mail: {kanaij,beleg}@isri.unlv.edu

Received September 23, 1997 / Revised December 17, 1997

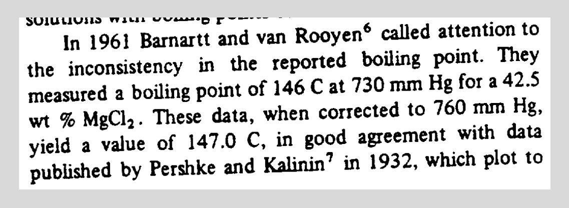
Abstract. A new projection profile based skew estimation algorithm is presented. It extracts fiducial points corresponding to objects on a page by decoding a JBIG compressed image. These points are projected along parallel lines into an accumulator array. The angle of projection within a search interval that maximizes alignment of the fiducial points is the skew angle. This algorithm and three other algorithms were tested. Results showed that the new algorithm performed comparably to the other algorithms. The JBIG progressive coding scheme reduces the effects of noise and graphics, and the accuracy of the new algorithm on 75 dpi unfiltered images and 300 dpi filtered images was similar.

Key words: Skew estimation – JBIG compression – Document image analysis – Performance evaluation

1 Introduction

Many document image layout analysis algorithms are designed to process page images with zero skew. A skew angle of a few degrees has significant visual effect as shown in Fig. 1. When readers view page images on a computer monitor, they expect the images to be displayed without any skew. Thus, it is important to estimate the skew angle of a page image accurately and to eliminate the effects of the skew.

Almost all techniques appearing in the technical literature extract features used to estimate skew angles from raw (uncompressed) bitmaps. Examples include a Fourier transform based method [8], the projection profile based methods [2,3,15,16], a Hough transform based method [9], a left margin search method [5], a local region complexity based method [11], interline cross-correlation based methods [1,20], a text row accumulation method [17], a fractal and least squares method [21], and a morphological method [4].



solutions with boiling p

In 1961 Barnartt and van Rooyen[6] called attention to the inconsistency in the reported boiling point. They measured a boiling point of 146 C at 730 mm Hg for a 42.5 wt % $MgCl_2$. These data, when corrected to 760 mm Hg, yield a value of 147.0 C, in good agreement with data published by Pershke and Kalinin[7] in 1932, which plot to

Fig. 1. Example of skewed text

However, document images are usually compressed before being exchanged and archived. Traditional document analysis algorithms decompress a page image completely before performing feature extraction. Algorithms that extract features during the compression or decompression process result in significant computational savings.

Recently several algorithms for processing CCITT Group III and IV compressed images have been developed. For example, Spitz presented an algorithm for skew estimation [18], and another for logotype detection [19]. Maa described an algorithm for barcode identification [14], and Hull and Cullen developed a method for determining document similarity and equivalence [10].

We present a projection profile based skew estimation algorithm that processes JBIG compressed images. Large scale experiments are conducted, and the performance of this algorithm is compared to that of other projection profile based algorithms.

The rest of the paper is organized as follows. Sect. 2 reviews projection profile based skew estimation algorithms. An overview of the JBIG compression scheme is presented in Sect. 3. Section 4 describes our skew estimation algorithm in detail. The accuracy performance of our algorithm and the algorithms reviewed in Sect. 2 is evaluated in Sect. 5. The speed of our algorithm is discussed in Sect. 6, and results are summarized in Sect. 7.

Correspondence to: J. Kanai

2 Projection profile based method

Projection profile based skew estimation is performed by the following three functions:

- A fiducial reduction function F reduces a source image, I, into a set of triples (x, y, w), which represent the position and weight of a fiducial point, corresponding to objects in I. Conceptually, fiducial points are a set of points considered representative of some image features. In the case of skew estimation, we desire fiducial points that represent characters in the source image.
- A projection function P projects the fiducial points along parallel lines into an accumulator array A. The accumulator array is typically partitioned into fixed height bins. The $BINSIZE$ is selected so as to maximize the chance of projecting fiducial points belonging to the same text-line into the same bin. The angle of projection, θ, is varied within an angular search interval, $[\theta_{\min}, \theta_{mathrmmax}]$, and a projection is done for each desired angle. This process creates a sequence of accumulator arrays, A_θ, corresponding to the search angles.
- Once the projection is performed, an optimization function ϕ calculates the alignment premium for each accumulator array A_θ. The angle θ corresponding to the accumulator array that maximizes the alignment premium, $\max_{\theta \in [\theta_{min}, \theta_{max}]} \phi(A_\theta)$, is then given as the skew estimate of the image.

Since the only substantial differences among projection profile based algorithms are the fiducial reduction method F and optimization function ϕ, three algorithms that will be compared to the new algorithm are characterized as follows.

2.1 Postl [16]

Fiducial points are extracted by sub-sampling the input image according to fixed horizontal and vertical sampling frequencies. Black pixels in the subsampled image become the fiducial points. Postl's fiducial reduction (F_P) can be described as:

$$F_P(I) = \{(x \times \varDelta\xi, y \times \varDelta\eta, 1) \mid 0 < x < w/\varDelta\xi, \\ 0 < y < h/\varDelta\eta, I[x \times \varDelta\xi, y \times \varDelta\eta] = 1\},$$

where x and y are pixel coordinates in the subsampled image, w and h are the width and height of the original image, and $\varDelta\xi$ and $\varDelta\eta$ are the horizontal and vertical sampling frequencies, respectively.

Postl's alignment function is then given by:

$$\phi_P(A_\theta) = \sum_{\rho=0}^{Height-1} (A_\theta[\rho+1] - A_\theta[\rho])^2$$

2.2 Baird [2]

Fiducial points are extracted by detecting connected components in the input image.

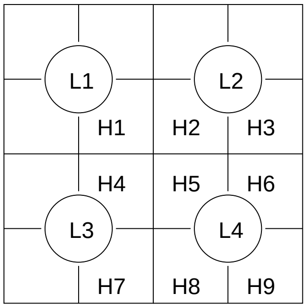


Fig. 2. Pixels that determine the value of a lower-resolution pixel

$$F_B(I) = \{(x, y, 1) \mid (x, y) \text{ is the center of lower bounding segment of a connected component in } I\},$$

$$\phi_B(A_\theta) = \sum_{\rho=0}^{height} A_\theta[\rho]^2.$$

2.3 Nakano [15]

Fiducial points are extracted by detecting connected components in the input image, and the width of a bounding rectangle is used as a weighting factor. The optimization process depends on the number of empty bins in the accumulator arrays.

$$F_N(I) = \{(x, y, w) | (x, y) \text{is lower left corner of a connected component of width } w\},$$

$$\phi_N(A_\theta) = \sum_{\rho=0}^{Height} [1 - u(A_\theta[\rho])],$$

where

$$u(x) = \begin{cases} 1 \text{ if } x = 0 \\ 0 \text{ otherwise} \end{cases}$$

3 JBIG compression scheme

The JBIG standard consists of two techniques, a progressive encoding method and a lossless compression method for the lowest resolution layer [12]. The main ideas of these techniques are summarized as follows.

The JBIG specification provides a table for generating lower-resolution pixels by combining two-by-two blocks from the higher-resolution image. In general, the color of the pixel labeled L4 in Fig. 2 is determined by the following rule.

```
if      4×H5+2×(H2+H4+H6+H8)
        + (H1+H3+H7+H9)-3×(L2+L3)
        - L1 > 4.5
then    L4 = 1
else    L4 = 0
```

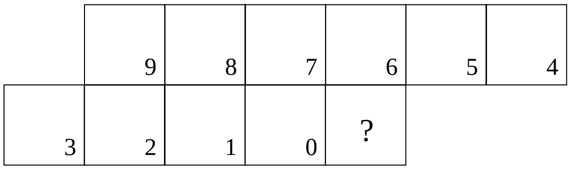


Fig. 3. Two-line template for determining coding context

However, the table contains certain exceptions to reduce the amount of edge smearing and to preserve periodic and dither pattern. The number of lower-resolution layers required to encode the input image is not specified. Four templates, which consist of 10 bits each, are used to determine coding contexts, and an arithmetic encoder is used to encode the lower resolution layer.

The first step in encoding the lowest resolution layer is to subdivide it into fixed height horizontal stripes which then are coded independently. The raster lines in a stripe can be coded in two distinct ways. A "pseudo-pixel" is coded at the beginning of the raster line to indicate which coding mode is used to encode it. If the current raster line is identical to the previous line, the pseudo-pixel will indicate the lowest-resolution-layer typical prediction (TP) mode. The previous line can simply be copied to reconstruct the current line, and no additional information about the current line is encoded.

If the current raster line is not identical to the previous line, each pixel in the raster line is encoded using one of 1024 parallel arithmetic encoders/decoders according to the context (pattern) made by its neighbor pixels. Figure 3 shows a two-line template to determine context, and the pixel labeled "?" indicate the one currently being coded. (The JBIG specification also supports a three-line template.) Let p_i, where $0 \leq i \leq 9$, denote the value of a pixel in the coding context; 0 and 1 correspond to white and black, respectively. An integer representing the coding context C_x of the current pixel is given by

$$C_x = \sum_{i=0}^{9} 2^i p_i .$$

This context C_x is used to index the array of arithmetic coders/decoders, and a sequence of pixels is coded using the corresponding contexts.

4 Skew estimation from JBIG compressed images

This algorithm uses the fiducial reduction strategy used for processing CCITT Group IV compressed images proposed by Spitz [18]. The CCITT Group IV compression algorithm is a two-dimensional coding scheme that encodes the current *coding* line with respect to the previous (*reference*) line [6]. Group IV compression has three coding modes, but we are only interested in the white pass-mode coding situations as shown in Fig. 4. These white pass-codes in the compressed image are shown by Spitz to be good fiducial points for skew estimation. Fig. 5 shows an example of fiducial points extracted by this approach.

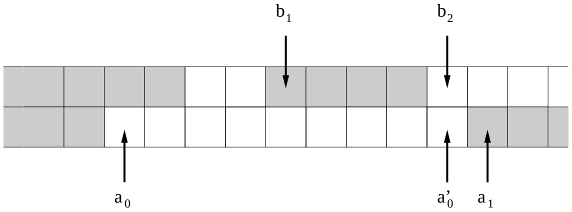


Fig. 4. A white pass-mode coding situation

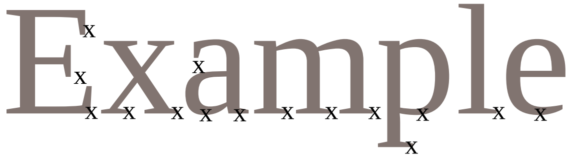


Fig. 5. An example of fiducial points extracted by detecting white pass codes

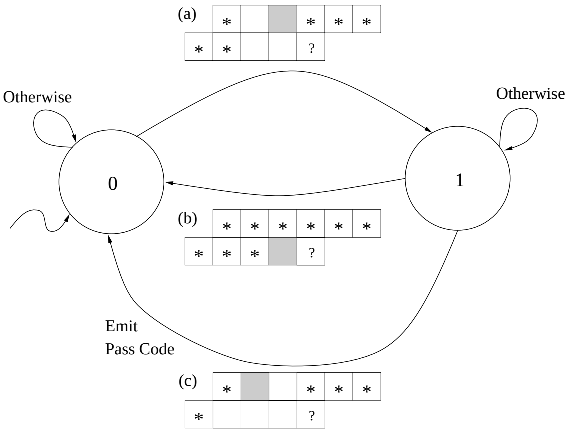


Fig. 6. Automaton to detect white passes from JBIG contexts

The JBIG compression algorithm does not generate a white pass code. Yet, this situation can be detected in the processes of encoding and decoding an image in the JBIG process by analyzing the sequence of coding contexts used to decode incoming pixels using the automaton shown in Fig. 6. In the figure, a shaded box in a context must be black for the transition to occur. An empty box indicates that the corresponding pixel must be white. A "$_*$" in a context position indicates that either pixel value is allowed. The "?" again indicates the pixel currently being coded. The operation of the above machine can be described as follows:

- In State 0, no potential pass code has been encountered. Figure 6a shows a sequence of two white pixels with a black pixel above, which indicates the beginning of a potential pass code. The automaton moves to State 1 if the pattern is seen. Otherwise, it remains in this state.
- In State 1, if a black pixel is encountered on the decoding line, the candidate is not a pass code as shown in Fig. 6b, and the automaton returns to State 0. If the machine encounters the end of a pass code (i.e., a black pixel followed by a white pixel on the refer-

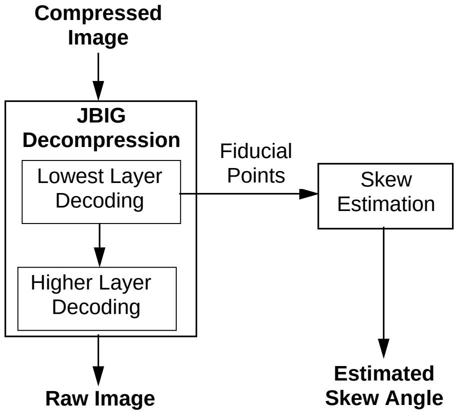


Fig. 7. Concurrent processing of JBIG decompression and skew estimation

ence line as shown in Fig. 6c), the machine emits the ending position of a run encoded by the pass code and returns to State 0. Otherwise, it remains in this state.

The use of 10 bit integers to represent the coding contexts makes the implementation of the above automaton particularly simple. Two 1024 entry transition tables are used to encode the automaton. Each entry in the tables consists of two bits: one for the new state, and one to indicate the output of a pass code. Because only two bits per entry are required, each table can be packed into a 256 byte table. When executing, the current coding context (a 10 bit integer) is used to index the appropriate transition table. Thus, the automaton can be implemented by storing two 256 byte tables, and a single bit of state information.

Since a fiducial point can be projected immediately without waiting for the remaining fiducial points to be identified, the projection process can be performed concurrently with the JBIG decompression process including the white pass code detection process as shown in Fig. 7.

In this project, the function ϕ_P proposed by Postl [16] was used to determine the optimal alignment. Therefore, this algorithm is characterized as follows:

$$F_J(I) = \{(x, y, 1) | \text{a white pass code is detected at } (x, y)\}$$

$$\phi_J(A_\theta) = \sum_{\rho=0}^{h-1} (A_\theta[\rho+1] - A_\theta[\rho])^2$$

More precisely, a fiducial point $(x, y, 1) \in F_J(I)$ is determined as $(a'_0, row, 1)$, where a'_0 is the variable computed as in Fig. 4 for a white pass code, and row is the current coding line at the time the white pass code is detected.

This algorithm was implemented in C. A free implementation of the JBIG standard [13] was used to facilitate our implementation of the white pass code detector that generates fiducial points.

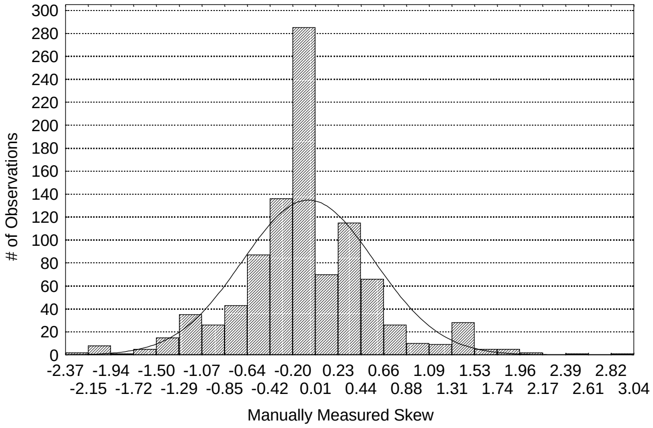


Fig. 8. The distribution of skew angles in test sample

5 Experiments

A number of experiments were conducted to evaluate the performance of our algorithm and the three projection profile based algorithms as described in Sect. 2. A $BINSIZE$ of 8 was used for all algorithms, and for Postl's algorithm the parameter settings $\Delta\eta = 8$ and $\Delta\xi = 16$ were used. All experiments were conducted using our implementations of the algorithms.

5.1 Test data

To identify typographical features and image defects that affect the performance of skew estimation algorithms, a random sample of real world images rather than synthesized images were used in this project. A test set consisting of 460 pages was selected at random from a collection of approximately 2 500 technical documents containing approximately 100 000 pages.

Each page was digitized at 300 dpi and manually segmented into a set of text zones. A text zone was classified into one of the following classes: *MainBody, Caption, Footnote, OtherText,* or *Table.* For each text zone, human operators manually measured its skew angle as accurately as they could. Some of the difficulties encountered include curved baselines and non-stationary skew angles. Figure 8 shows the distribution of manually measured skew angles in the test set containing 1 246 text zones.

Assigning a single skew angle to a given page was often difficult because some pages contained text zones skewed at different angles. Thus, two aggregate measures of skew were evaluated separately. The first aggregate measure, called the *dominant* skew angle, is defined to be the manually measured skew angle for the largest text zone on the page. The second measure is a *weighted average* of all zone skew angles for the page. If a zone on a given page is denoted as (a, θ), where a is the area of the zone, and θ is its estimated skew angle, the weighted average angle of a page I is given by

$$WeightedAverage = \frac{\sum_{(a,\theta)\in I} a \times \theta}{\sum_{(a,\theta)\in I} a}$$

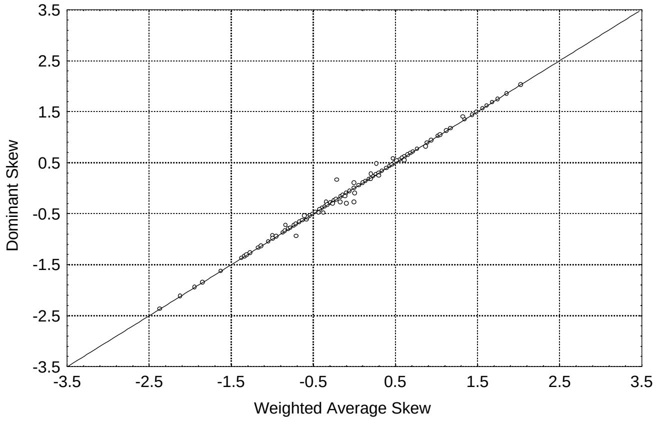


Fig. 9. Scatter plot of weighted average vs. dominant skew

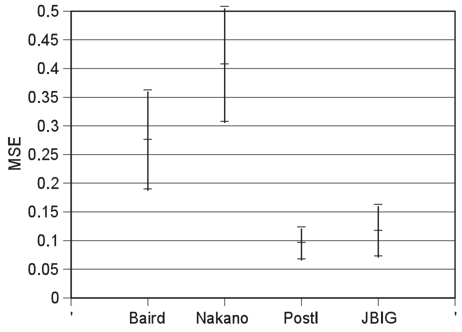


Fig. 10. Confidence intervals for MSE using all zones

Figure 9 shows a scatter plot of the two measures as computed for the 460 page sample. The correlation between the dominant angles and weighted average skew angles was 0.998, and there is a nearly perfect correlation between the two measures as the figure shows. Thus, the weighted average skew angle of a page is used as the ground truth data in this work.

5.2 Zone based evaluation

The skew angle of a given textzone was determined with each skew estimation algorithm. For each algorithm, a linear correlation between the estimated angle and the ground truth skew angle was determined. Approximate 95% confidence intervals for the mean of squared errors (MSEs) were also computed using the *jackknife estimator* [7]. A narrow confidence interval indicates that an algorithm performed consistently within a given set of test data. On the other hand, a wide interval indicates considerable variability. When two intervals do not overlap, there is a statistically significant difference between the performance of two algorithms.

Table 1 summarizes the linear correlation coefficients, y-intercepts, and slopes obtained from the linear regressions. A perfect estimator should achieve a correlation coefficient of 1.0, y-intercept of 0.0, and slope of 1.0. Figure 10 presents the confidence intervals for MSEs. The results show that, on average, both Postl's algorithm and the JBIG algorithm are more accurate than the methods that extract fiducial points from connected components.

Table 1. Summary of linear regression for all sample zones

Algorithm	Correlation	Y-intercept	Slope
	ρ	α	β
Baird	0.796	-0.159	0.937
Postl	0.897	-0.018	0.829
Nakano	0.736	-0.260	0.908
JBIG	0.889	-0.029	0.945

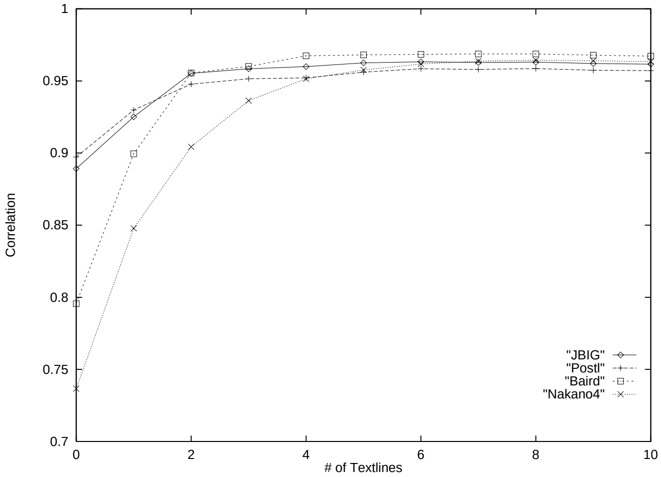


Fig. 11. Effect of textlines on accuracy. Each point represents the correlation of the algorithm on zones of greater than x textlines

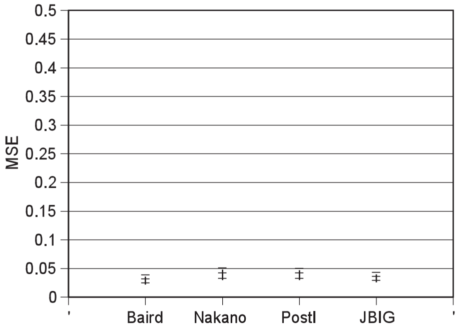


Fig. 12. Confidence intervals for MSE using zones containing at least five textlines

Table 2. Summary of linear regression for zones of type *MainBody*

Algorithm	Correlation	Y-intercept	Slope
	ρ	α	β
Baird	0.882	-0.101	0.926
Postl	0.926	-0.025	0.866
Nakano	0.844	-0.152	0.921
JBIG	0.896	-0.017	0.934

A visual inspection of a set of common zones that caused all algorithms to make large errors revealed that they contained few textlines, such as *Caption*. Table 2 summarizes the regression information for zones of type *MainBody* that contain many textlines. The results show improved correlation and suggest that these algorithms are sensitive to the number of textlines in a text zone.

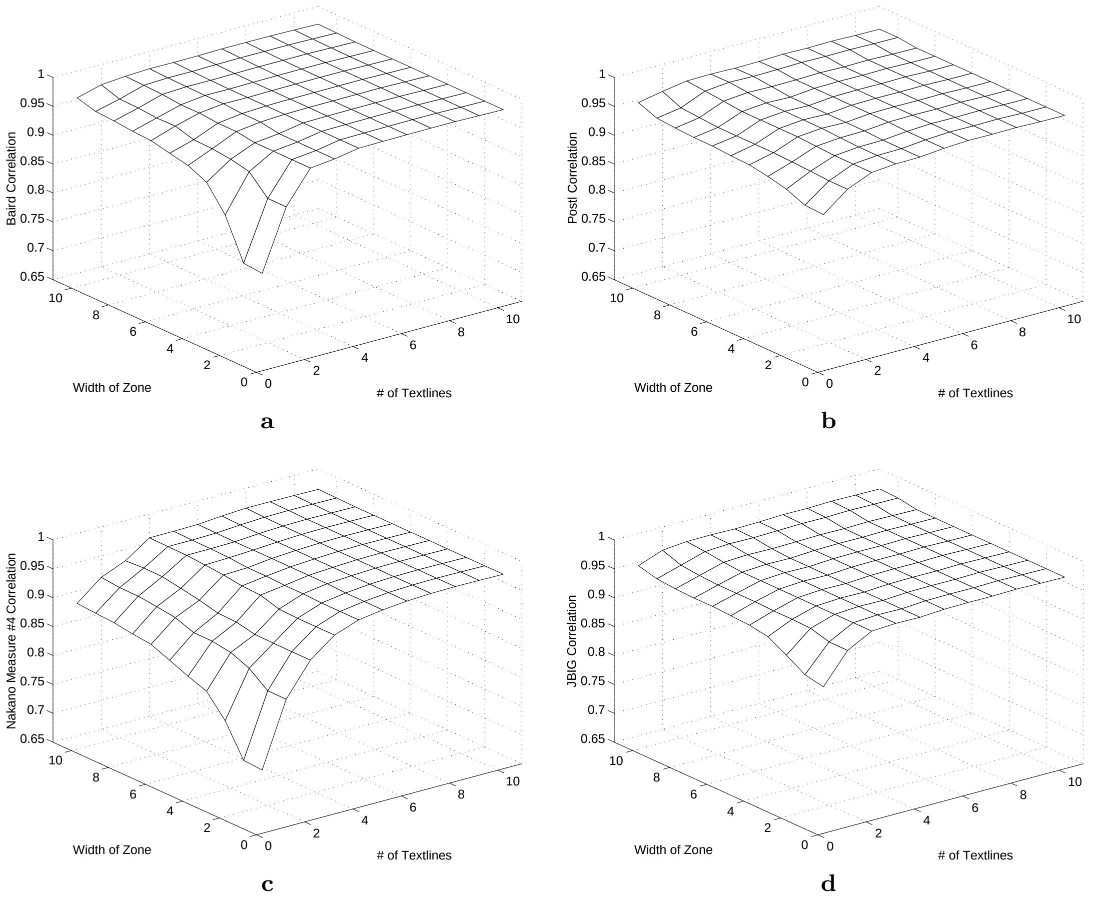


Fig. 13a–d. Effect of textlines and zone width on accuracy. Each point (x, y) represents the correlation on zones of greater than x textlines and width greater than y. **a** Baird, **b** Postl, **c** Nakano, and **d** JBIG

To test this hypothesis, the correlation of each algorithm as a function of the number of textlines in each zone was calculated. The observations corresponding to zones with fewer than a certain number of textlines were iteratively eliminated from the linear regression. Figure 11 shows that most algorithms achieve a stable correlation value below 10 textlines. On the other hand, the accuracy of these algorithms on small zones varies widely; therefore, they should be used cautiously to determine the local skew angles of small text zones.

Figure 12 shows the confidence intervals for MSEs using zones containing at least five textlines. These results also confirm that there is no significant difference among them on large text zones.

The linear correlation analysis was also used to determine whether or not the zone width, which is related to the textline length (and therefore the number of characters), has any effect on the accuracy above and beyond the number of textlines. Figure 13 shows that the width of a zone has some effect on the accuracy of the algorithms. Specifically, if a zone has few textlines, the width of a zone can affect the performance of skew estimation. However, the number textlines is certainly the dominant feature.

Table 3. Summary of linear regressions for pages and weighted average skew

Algorithm	Correlation	Y-intercept	Slope
	ρ	α	β
Baird	0.784	-0.019	0.778
Postl	0.859	-0.016	0.786
Nakano	0.866	-0.038	0.857
JBIG	0.783	0.001	0.749

5.3 Page-based evaluation

The skew angle of each page was estimated using each algorithm without filtering non-text objects, such as figures and noise. Table 3 summarizes the regression analysis for the weighted average skew. The surprising result is that Nakano's algorithm outperformed all of the others on the basis of linear regression. It appears that the weighting used in the fiducial reduction function, F, reduces the effects of non-text elements.

Figure 14 presents the confidence intervals for MSE on page-based skew estimation. The results show that there is no statistically significant difference among them in processing the page images.

The performance, $\rho = 0.78$, of the JBIG algorithm was significantly worse than its performance, $\rho = 0.89$, for processing the zone-based test data. It was hypoth-

Table 4. Summary of linear regressions for the ground truth data and weighted average skew after filtering non-text elements out

Algorithm	Correlation ρ	Y-intercept α	Slope β
Baird	0.922	-0.038	0.938
JBIG	0.888	0.005	0.923

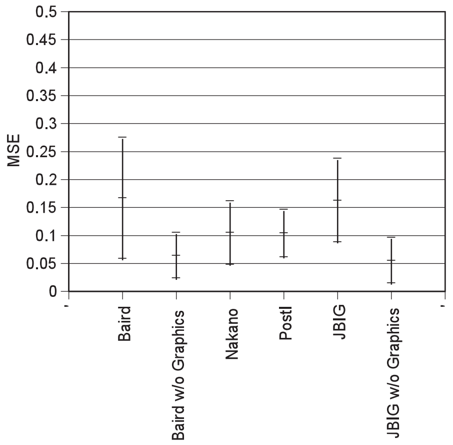


Fig. 14. Confidence intervals for MSE using page-based data

esized that the problem was caused by non-text elements because the JBIG algorithm could potentially detect many pass codes from a single non-text element. To test this hypothesis, the skew angle of each page was re-estimated with the JBIG algorithm using only pass codes falling inside of text zones. Similarly, Baird's algorithm also re-estimated the skew angle of each page.

Table 4 shows that the performance of both algorithms improved. Figure 14 also shows the improvements made by the filtering process as indicated by the narrower confidence intervals and smaller MSEs. These results indicate that the non-text features of the pages are contributing a significant amount of the error for projection profile-based skew estimation techniques.

5.4 Effects of progressive coding

Additional test data sets consisting of lower resolution images, 150 dpi, 75 dpi, and 38 dpi, generated by the JBIG progressive encoding method were processed with the JBIG skew estimation algorithm. Table 5 summarizes their effects of on the performance of the algorithm, and Fig. 15 shows the confidence intervals for MSE. Both analyses show that reducing the image resolution improves the performance of the new algorithm.

The best result was achieved on the set of 75 dpi images, and the correlation, $\rho = 0.91$, was better than the correlation, $\rho = 0.89$, on 300 dpi images with filtering non-text objects. The corresponding confidence interval is also similar to the interval for processing 300 dpi images with the filtering process. These results are also competitive with the performance of Baird's algorithm on 300 dpi filtered images. Visual inspection of the detected fiducial points from low resolution images confirmed that, in general, the number of fiducial points generated from noise and graphic objects decreased as the image resolution was reduced.

Table 5. Effects of the JBIG progressive encoding method

Resolution	Correlation ρ	Y-intercept α	Slope β
300 dpi	0.783	0.001	0.749
150 dpi	0.837	-0.004	0.842
75 dpi	0.906	-0.020	0.859
38 dpi	0.870	-0.072	0.828

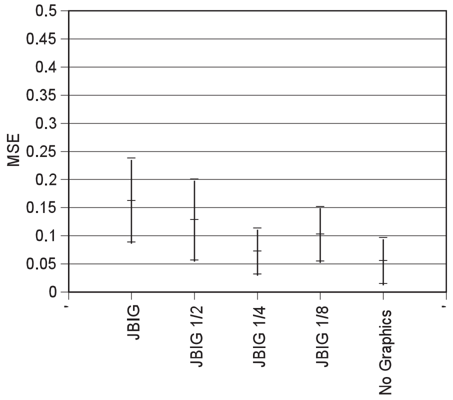


Fig. 15. Confidence intervals for MSE using the JBIG progressive coding

These results show that the JBIG algorithm can reasonably estimate the skew angle of a 38 dpi unfiltered page image. The process time required by the fiducial reduction process is approximately proportional to the number of pixels in the lowest resolution layer. Since an eightfold reduction in image resolution can result up to 64-fold reduction in the fiducial extraction process, significant speed improvement can be achieved by processing 38 dpi images.

6 Speed issues

It is hard to perform a fair and accurate comparison of skew estimation algorithms on the basis of processing speed because of difficulties in optimizing implementations and setting parameters. Thus, our observations are presented but no speed data are analyzed in this section.

Due to the simplicity of our pass code extraction procedure, our method adds only minimal overhead to the JBIG decompression process. The progressive coding scheme also assists in increasing the speed. Resolution reduction decreases the amount of time spent estimating skew by reducing the number of fiducial points processed. Moreover, as described in Sect. 4, the skew estimation can be performed in parallel with the decompression of subsequent layers. Hence, if the source images are compressed by the JBIG method, our approach is faster than those that must decompress the files first.

Experimental results showed the improved accuracy of our method on lower resolution images generated by

JBIG progressive encoding. Significant speed improvement can be achieved by processing lower resolution images without degrading estimation accuracy.

Converting a raw image to JBIG format is probably not justified because the same lower resolution images can be generated without encoding differential layers and the same set of fiducial points can easily be extracted from a raw image or a lowest resolution layer using a lookup table and an automaton similar to the one in Fig. 6.

7 Conclusions

A new projection profile based skew estimation algorithm has been presented. This algorithm extracts fiducial points by decoding the lowest resolution layer of a JBIG compressed image. The accuracy of this algorithm was determined using 460 page images and 1 246 single column text zones extracted from the page images. The results show that the new algorithm is competitive with other projection profile based algorithms. For single column text zones, the number of textlines in a zone significantly affects the accuracy of the skew estimation. For page images, graphic and noise elements contributed a significant amount of the error for this new algorithm; therefore, a method is needed to remove these elements from JBIG compressed images.

The JBIG compression algorithm supports a progressive coding scheme that allows the user to change the resolution of the lowest resolution layer. The results show that this resolution reduction scheme minimizes the effects of noise and graphic objects in a page in the skew estimation process. The best result was achieved by the new algorithm when the resolution was reduced from 300 dpi to 75 dpi. Moreover, the performance was competitive with Baird's algorithm processing 300 dpi images with filtering non-text objects.

Our fiducial reduction algorithm for JBIG images can be easily extended to detect both white and black pass-modes. Other document image analysis algorithms that detect pass-codes in CCITT compressed images could be implemented for JBIG compression as well. The effects of (and potential benefits of) JBIG resolution reduction with respect to these algorithms should be studied.

Acknowledgements. This project was supported in part by a grant from the U.S. Department of Energy and NASA Research Grant NAG5-3994. The authors would like to thank L. Spitz (Document Recognition Technologies, Inc.) and Prof. G. Nagy (RPI) for helpful discussions. We received valuable assistance from Prof. T. A. Nartker, D. Vinas, other ISRI members.

References

1. Avanindar, Chaudhuri S (1997) Robust Detection of Skew in Document Images. IEEE Transactions on Image Processing 6:2:344–349 1997
2. Baird HS (1987) Skew Angle of Printed Documents. In: Proc. of SPSE's 40th Annual Conference and Symposium on Hybrid Imaging Systems. Rochester, NY, pp 21–24
3. Bloomberg DS, Kopec GE, Dasari L (1995) Measuring Document Image Skew and Orientation. In: Vincent LE, Baird HS (eds) Document Recognition II. SPIE, pp 302–316
4. Chen S, Haralick RM, Phillips IT (1995) Automatic Text Skew Estimation in Document Images. In: Proc. of 3rd ICDAR. Montréal, Canada, pp 1153–1156
5. Dengel A (1991) ANASTASIL: A System for Low-Level and High-Level Geometric Analysis of Printed Documents. In: Baird HS, Bunke H, Yamamoto K (eds) Structured Document Image Analysis, Springer-Verlag, Berlin Heidelberg New York, pp 70–98
6. CCITT Recommendation T.6 (1989) Facsimile Coding Schemes and Control Functions for Group IV Facsimile Apparatus. In: Terminal Equipment and Protocols for the Telematic Services VII, Fascicle VII.3
7. Dudewicz EJ, Mishra SN (1988) Modern Mathematical Statistics, John Wiley & Sons
8. Hase M, Hoshino Y (1985) Segmentation Method of Document Images by Two-Dimensional Fourier Transformation. Systems and Computers in Japan 16:3:38–47
9. Hinds SC, Fisher JL, D'Amato DP (1990) A Document Skew Detection Method Using Run-Length Encoding and the Hough Transform. In: Proc. of 10th ICPR, Atlantic City, NJ, pp 464–468
10. Hull JJ, Cullen JF (1997) Document Image Similarity and Equivalence Detection. In: Proc. of 4th ICDAR, Ulm, Germany, pp 308–312
11. Ishitani Y (1993) Document Skew Detection Based on Local Region Complexity. In: Proc. of 2nd ICDAR, Tsukuba, Japan, pp 49–52
12. International Standard ISO/IEC 11544:1993 and ITU-T Recommendation T.82 (1993) Information Technology – Coded Representation of Picture and Audio Information – Progressive Bi-Level Image Compression
13. Kuhn M (1996) JBIG-KIT V0.9. Available from ftp://ftp.informatik.uni-erlangen.de/pub/doc/ISO/JBIG/jbigkit-0.9.tar.gz.
14. Maa C (1994) Identifying the Existence of Bar Codes in Compressed Images. In: CVGIP: Graphical Models and Image Processing 56:4:352–356
15. Nakano Y, Shima Y, Fujisawa H, Higashino J, Fujinawa M (1990) An Algorithm for the Skew Normalization of Document Images. In: Proc. of 10th ICPR, Atlantic City, NJ, pp 8–11
16. Postl W (1986) Detection of Linear Oblique Structures and Skew Scan in Digitized Documents. In: Proc. of 8th ICPR, Paris, France, pp 687–689
17. Smith R (1995) A Simple and Efficient Skew Detection Algorithm via Text Row Accumulation. In: Proc. of 3rd ICDAR, Montréal, Canada, pp 1145–1148
18. Spitz AL (1992) Skew Determination in CCITT Group IV Compressed Document Images. In: Proc. of Symposium on Document Analysis and Information Retrieval, Las Vegas, NV, pp 11–25
19. Spitz AL (1996) Logotype Detection in Compressed using Alignment Signatures. In: Proc. of Symposium on Document Analysis and Information Retrieval, Las Vegas, NV, pp 303–310
20. Yan H (1993) Skew Correction of Document Images Using Interline Cross-Correlation. CVGIP Graphical Models and Image Processing 55:538–543

21. Yu CL, Tang YY, Suen CY (1995) Document Skew Detection Based on the Fractal and Least Squares Method. In: Proc. of 3rd ICDAR, Montréal, Canada, pp 1149–1152

Junichi Kanai received the B.S. degree in Electrical Engineering, the M.Eng. and the Ph.D. degrees in Computer and Systems Engineering from Rensselaer Polytechnic Institute in 1983, 1985, and 1990, respectively. He is currently Assistant Research Professor of the Information Science Research Institute at the University of Nevada, Las Vegas. His research interest include document image analysis, optical character recognition, data compression, signal processing, and pattern recognition. Dr. Kanai is a member of ACM, IAPR, IEEE, and Sigma Xi.

Andrew Bagdanov is currently a Ph.D. student in Computer Science at the University of Nevada Las Vegas (UNLV). He received a dual Bachelors in Mathematics and Computer Science and his M.S. in Computer Science from UNLV in 1995 and 1996, respectively. Mr. Bagdanov has worked for the Information Science Research Institute since 1991.