Noname manuscript No.
(will be inserted by the editor)

Combined Orientation and Skew Detection Using Geometric Text-Line Modeling

Joost van Beusekom · Faisal Shafait · Thomas M. Breuel

Received: date / Accepted: date

Abstract In large scale document digitization, orientation detection plays an important role, especially in the scenario of digitizing incoming mail. The heavy use of automatic document feeding (ADF) scanners and more over automatic processing of facsimiles results in many documents being scanned in the wrong orientation. These misoriented scans have to be corrected, as most subsequent processing steps assume the document to be scanned in the right orientation. Several existing methods for orientation detection use the fact that in Latin script text, ascenders are more likely to occur than descenders. In this paper, we propose a one-step skew and orientation detection method using a well established geometric text-line model. The advantage of our method is that it combines accurate skew estimation with robust, resolution independent orientation detection. An interesting aspect of our method is that it incorporates orientation detection into a previously published skew detection method allowing to perform orientation detection, skew estimation, and, if necessary, text-line extraction in one step. The effectiveness of our orientation detection approach is demonstrated on the UW-I dataset, and on publicly available test images from OCRopus. Our method achieves an accuracy of 99% on the UW-I dataset and 100% on test images from OCRopus.

Keywords Orientation detection, document preprocessing, line finding

Joost van Beusekom
Department of Computer Science
Technical University of Kaiserslautern
D-67663 Kaiserslautern, Germany
E-mail: beusekom@rhrk.uni-kl.de

Faisal Shafait
Image Understanding and Pattern Recognition (IUPR) Research Group
German Research Center for Artificial Intelligence (DFKI)
D-67663 Kaiserslautern, Germany
Corresponding author:
Tel: +49-631-20575-415
Fax: +49-631-20575-402
E-mail: Faisal.Shafait@dfki.de

Thomas M. Breuel
Department of Computer Science
Technical University of Kaiserslautern
D-67663 Kaiserslautern, Germany
E-mail: tmb@informatik.uni-kl.de

1 Introduction

Several initiatives have recently been launched for large scale scanning of documents, books and other paper-based materials [1]. But also in office environments as e.g. invoice processing, more and more high-volume digitization is being done. Especially in the domain of mail-room digitization, orientation detection is an important preprocessing step. The use of ADF scanners is likely to introduce misoriented scans. According to estimates by a company providing document management system solutions there are around $1\% - 5\%$ of the documents that are scanned in the wrong orientation. A more important number of misoriented pages originates from electronic facsimile processing. As most fax machines are using automatic document feeding that is opposite to the intuitive direction of paper insertion, it often occurs that misoriented facsimiles are received. The estimated percentage of misoriented facsimiles is about 50%.

Among noise removal [2] and binarization [3], skew and orientation detection plays an important role in document image preprocessing, as the subsequent processing methods will fail to work correctly since usually both layout analysis [4] and OCR [5,6] assume pages to be in the correct orientation.

In literature no clear definition or distinction between both steps can be found. Skew detection methods are often presented with minimum and maximum rotation angles that can be detected. These reach from $\pm 5°$ to $\pm 90°$. For orientation detection the same problem can be noticed. Some authors consider only upside up and upside down as possible orientations, others also consider upside left and upside right as possible orientations.

Due to the fact that most skew estimation methods allow skew angles of $\pm 45°$ or less, we define the problem of orientation detection as a four class problem, namely detecting one of the following orientations: 0°, 90°, 180° or 270°. We thus define skew estimation as finding a rotation angle lying between $-45°$ and $+45°$.

The remaining sections of the paper are organized as follows: Section 2 gives an overview over related methods. Section 3 explains the proposed method. In Section 4 evaluation procedures and experimental results are discussed. The paper is concluded by Section 5.

2 Previous Work

Previous work on orientation and skew detection can be categorized into three parts:

1. skew-only detection methods
2. orientation-only detection methods
3. combined skew and orientation detection methods.

2.1 Skew Detection Methods

For skew detection, a good overview of the state of the art methods is given by Cattoni et al. [7]. The methods are divided into different categories:

– analysis of projection profiles: the basic idea is to compute projection profiles for all skew angles and evaluate an objective function. The drawback of this method lies in its computationally expensive search: accurate skew estimates can only be obtained using a high the angular resolution, which leads to extensive computations.
– Hough transform: using the attributes that text-lines are parallel and that the characters in each text-line are aligned, the Hough transform can be applied to estimate the skew angle of the document. Having the advantage of being able to find any angle, the Hough transform itself presents considerable computational effort.
– clustering of connected components: assuming that characters of the same line are aligned and closer to each other as compared to characters of other lines, their spatial relationship can be used to estimate the skew angle.
– other techniques: other approaches include analysis of vertical deviation, analysis of the gradients on the downsampeled images, Fourier transform and morphological operations.

A more recent work on skew detection is presented by Lu [8]. Connected components are merged with their nearest neighbor components to chains. The slope is estimated from each of these chains and finally the document skew is estimated using the mean or mode of the slopes. Comparison and evaluation on a non public dataset showed reasonable results with a mean error of 0.32°.

2.2 Orientation Detection Methods

For orientation detection, several approaches have been presented in the past. A method using the ascender to descender ratio was presented by Caprari [9] in 1999. Top up and top down orientations are detected using an asymmetry measure computed from projection profiles, which makes it sensible to skewed pages. An accuracy of 100% is reported on a non-public dataset of 226 images containing mainly text and only little skew.

Aradhye [10] presents in 2005 an up/down orientation detection method for Roman and Non-Roman scripts. Text blobs are extracted and the so called opening to the left and to the right is computed for each blob. The ratio between left and right opening is then used to decide whether the page is rotated 0° or 180°. The method was tested on different datasets and reached accuracies from 87% to 100%. On the UW-I dataset an accuracy of 99% was obtained.

In 2006, Lu et al. [11] present a method for language and orientation detection using the distributions of the number and position of white to black transitions of the components in the line. The performance of their method is reported on a partially non-public dataset and achieves a success rate of 98.2% for documents with at least 12 text-lines.

2.3 Combined Skew and Orientation Detection Methods

Methods solving skew and orientation detection together can also be found in literature. However, most often two different approaches are used to detect skew and orientation. In 1994 Le et al. [12] presented a system capable of detecting portrait or landscape mode images and subsequently the skew angle. Top down document images are not considered. It uses rules based on projection profiles and textual features for orientation detection. Hough transform is then used in the second step to determine the skew of the page. Evaluation has been done on a non-public dataset of pages of medical journal articles. An error rate of 0.1% is reported.

In 1995 Bloomberg et al. [13] presented a method for orientation detection that uses the ratio of the number of ascender characters to the number of descender characters used in English. As ascenders are more frequent than descenders, this can be used to detect the correct orientation. Ascenders and descenders are extracted using morphological operations. Skew detection is also provided. This is done by finding the maximum of the so called “skew function” which is maximal for the correct skew angle. Using two different search strategies, the best angle is computed. The reported error rate for orientation detection on UW-I [14] dataset is one wrongly detected orientation versus 938 correctly detected. The orientation of 41 documents could not be extracted but are not considered as errors. Skew detection error histograms are also given for the UW-I dataset. An implementation of this method is available in Leptonica[1], an open source image processing library.

A more recent method using the ascender to descender ratio for orientation detection is presented by Avila et al. [15]. Using the x-height line and the base line of the text-lines, the number of ascenders and descenders is computed. If the number of descenders is higher than the number of ascenders, the page is considered being upside down. Skew detection is done in this paper by grouping connected components to lines and building an angle histogram of the lines. The reported error rate for orientation detection is below 0.1% on a non-public dataset.

Lu [16] et al. present another method to solve the orientation and skew detection problem. Similar to their previous work, white run histograms are used to detect a characteristic peak. This is then used to estimate the skew angle. Orientation detection is classically solved using ascender to descender ratio. A comparison between their method and other methods found in literature is done. Unfortunately, the test set consists of only 52 documents of a non public dataset

In this paper we propose a new method for combined skew and orientation detection using geometric modeling of Latin script text-lines. The method searches for text-line candidates within a skew range of all four orientations. The best fit of the model gives the estimate of both page skew and orientation.

[1] http://www.leptonica.com

3 Skew and Orientation Detection

The presented method for page skew and orientation detection is based on Latin script text-line model by Breuel [17]. We will first illustrate the geometric text-line model since it is crucial for the understanding of this work.

3.1 Geometric Text-Line Model

Breuel proposed a parameterized model for a text-line with parameters (r, θ, d), where r is the distance of the baseline from the origin, θ is the angle of the baseline from the horizontal axis, and d is the distance of the line of descenders from the baseline. This model is illustrated in Figure 1. The advantage of explicitly modeling the line of descenders is that it removes the ambiguities in baseline detection caused by the presence of descenders. A visualization of the different lines composing a text-line can be found in Figure 2.

Based on this geometric model of Latin script text-lines, we use geometric matching to extract text-lines from scanned documents as in [17]. A quality function is defined which gives the quality of matching the text-line model to a given set of points. The goal is to find a collection of parameters (r, θ, d) for each text-line in the document image that maximizes the number of bounding boxes matching the model and that minimizes the distance of each reference point from the baseline in a robust least square sense. The RAST algorithm [18, 19] is used to find the parameters of all text-lines in a document image. The algorithm is run in a greedy fashion such that it returns text-lines in decreasing order of quality.

Consider a set of reference points $\{x_1, x_2, \cdots, x_n\}$ obtained by taking the middle of the bottom line of the bounding boxes of the connected components in a document image. The goal of text-line detection is to find the maximizing set of parameters $\vartheta = (r, \theta, d)$ with respect to the reference points $\{x_1, x_2, \cdots, x_n\}$:

$$\hat{\vartheta} := \arg\max_{\vartheta} Q_{x_1^n}(\vartheta) \tag{1}$$

The quality function used in [17] is:

$$\begin{aligned} Q_{x_1^n}(\vartheta) &= Q_{x_1^n}(r, \theta, d) \\ &= \sum_{i=1}^{n} \max(q_{(r,\theta)}(x_i), \alpha q_{(r-d,\theta)}(x_i)) \end{aligned} \tag{2}$$

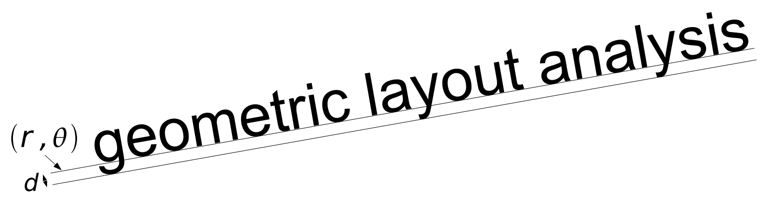


Fig. 1 An illustration of Latin script text-line model proposed by Breuel [17]. The baseline is modeled as a straight line with parameters (r, θ), and the descender line is modeled as a line parallel to the baseline at a distance d below the baseline.

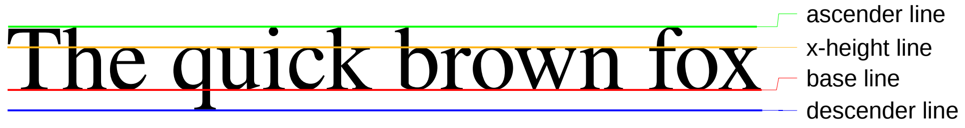


Fig. 2 The anatomy of a text-line in Roman script

where

$$q_{(r,\theta)}(x) = \max\left(0, 1 - \frac{D^2_{(r,\theta)}(x)}{\epsilon^2}\right) \tag{3}$$

The first term in the summation of Equation 2 calculates the contribution of a reference point x_i to baseline, whereas the second term calculates the contribution of a reference point x_i to the descender line. Since a point can either lie on the baseline or the descender line, maximum of the two contributions is taken in the summation. Typically the value of α is set to 0.75, and its role is to compensate for the inequality of priors for baseline and descender line such that a reference point is more likely to be matched with the baseline as compared to the descender line. The contribution of a reference point to a line is measured using Equation 3 and its value lies in the interval $[0, 1]$. The contribution $q_{(r,\theta)}(x)$ is zero for all reference points for which their Euclidean distance $D_{(r,\theta)}(x) \geq \epsilon$, ϵ given in pixel. These points are considered as outliers and hence do not belong to the line with parameters (r, θ). In practice, $\epsilon = 5$ proves to be a good choice for documents scanned at usual resolutions ranging from 150 to 400dpi: it gives enough flexibility to account for slight variations due to pixel noise, binarization degradations and optical corrections in typography that allow characters to start actually slightly below the baseline. The contribution $q_{(r,\theta)}(x) = 1$ if $D_{(r,\theta)}(x) = 0$ which means the contribution of a point to a line is one if and only if the point lies exactly on the line.

The RAST algorithm is used to extract the text-line with maximum quality as given by Equation 1. Then all reference points that contributed with a non-zero quality to the extracted text-line are removed from the list of reference points and the algorithm is run again. In this way, the algorithm returns text-lines in decreasing order of quality until all text-lines have been extracted from the document image.

3.2 Text-Line Extraction

As mentioned in Section 3.1, the text-lines are defined by three parameters (r, θ, d). The text-line extraction consists in finding parameter triples that maximize Equation 2. This is done using a branch-and-bound search. In the first step, the parameter space P is defined:

$$P = [r_{min}, r_{max}] \times [\theta_{min}, \theta_{max}] \times [d_{min}, d_{max}] \tag{4}$$

For practical applications, $r_{min} = 0$ and r_{max} can be fixed using the image size. As large skew angles $|\theta| > 20°$ are unlikely to occur (in the afore mentioned applications, such skew angles will likely lead to a paper feed error on the scanning device), the range of the rotation angle $[\theta_{min}, \theta_{max}]$ is set to $[-20°, 20°]$. Finally the possible distances of the descender line to the base line has to be set. For the current setup this is fixed to $[0, 30]$. The theoretical size of a 12pt character scanned with 300dpi is 50px. For common fonts the distance between the ascender and the baseline is about 1/4 to 1/3 of the total height, thus the above setting of the parameter d gives enough flexibility.

The branch and bound algorithm works as follows:

```
0: initialize the search space and insert it into priority
   queue $Q$
1: stop if Q is empty, else get the top search space $S$
   from $Q$
2: if $S$ is a solution: save $S$ and discard all image points
   contributing to the line from further consideration;
   Stop if enough results are found. Otherwise, continue with Step 1.
3: split $S$ into two subspaces $S_1$ and $S_2$
4: compute upper bounds for the quality of $S_1$ and $S_2$
5: put $S_1$ and $S_2$ on $Q$. Continue with Step 1.
```

The computation of the upper bound for the quality of the text-line defined by the parameter subspace is done using interval arithmetic. A priority queue is used to keep track of the subspaces that need to be analyzed. The number of lines to be extracted is used in Step 6 to define the stopping criterion. A space is considered as a solution if it is small enough to identify the unique line in the image. The parameters of the solution represent the detected line.

As mentioned in Section 3, the set of reference points $\{x_1, x_2, \cdots, x_n\}$ is obtained by taking the middle of the bottom line of the bounding boxes of the connected components in a document image. Instead of taking all connected components, a filtering step is used to discard too small or too big components, in order to increase the robustness of the method. The modes of the width and height of the connected components are computed. Based on these, $min_{area} = 2 \times y_{mode}$, $max_{height} = 10 \times y_{mode}$, $min_{height} = 0.5 \times y_{mode}$, $max_{width} = 10 \times x_{mode}$, $min_{width} = 0.5 \times x_{mode}$ and $min_{aspect} = 10.0$ are defined and used as thresholds for the width, height, aspect-ratio and area of the connected components. To simplify the reading, in the following we refer to the filtered set of connected components simply as the connected components.

3.3 One-Step Skew and Orientation Detection

The key idea in our approach is to use ascender modeling in the same way as modeling descenders. An illustration is shown in Figure 3. The x-line (the line passing through the top of non-ascending lower case characters like x, a, c, etc.) is modeled as a straight line with parameters (r, θ), and the ascender line is modeled as a line parallel to the x-line at a distance a above the x-line.

Consider a set of reference points $\{y_1, y_2, \cdots, y_n\}$ obtained by taking the middle of the top line of the bounding boxes of the connected components in a document image. The goal of text-line detection is to find the maximizing set of parameters $\vartheta = (r, \theta, a)$ with respect to the reference points $\{y_1, y_2, \cdots, y_n\}$:

$$\hat{\vartheta} := \arg\max_{\vartheta} Q_{y_1^n}(\vartheta) \tag{5}$$

The quality function can then be defined as:

$$\begin{aligned} Q_{y_1^n}(\vartheta) &= Q_{y_1^n}(r, \theta, a) \\ &= \sum_{i=1}^{n} \max(q_{(r,\theta)}(y_i), \alpha q_{(r+a,\theta)}(y_i)) \end{aligned} \tag{6}$$

where the local quality can be computed in the same way as Equation 3.

Since in Latin script ascenders are more likely to occur than descenders, more components will match the ascender line than the descender line. A component matching to descender/ascender gets a lower score (due to the factor α in Equations 2 and 6) as compared to a component matching to baseline/x-line. Therefore, in general the total quality of the descender line (Equation 2) will be higher than the total quality of the ascender line (Equation 6). This information is used in this work to find the upside down orientation of the page. We sum the quality of n best lines returned by RAST first using the descender model and then using the ascender model. If the quality of the ascender model is higher than the descender model, the page is reported as upside down (180 degrees rotated).

Note that computing the ascender model for a given page image in a correct orientation is equivalent to computing the descender model for a 180 degree rotated page. Therefore, for any given image we only compute the descender model for the original image and for 180 degree rotated image. The image that results in better descender quality is reported as the one with the correct orientation. This concept is then easily extended to detected pages with a 90 degree and 270 degree orientation. The horizontal text-line model does not fit well on vertical text-lines, so for a right side up portrait page the total quality of n best lines in the vertical direction is much lower than the total quality of n best horizontal lines. Hence by computing the descender quality by rotating the page with all four orientation, we can find out the correct orientation of the page. An illustration is shown in Figure 4.

An interesting aspect of our method is that besides an estimate of page orientation, we automatically get estimates for page skew since the skew angle θ of all detected text-lines is known at this stage. We choose the skew of the text-line with the highest quality as the global skew of the page since this has already shown to give very accurate results for skew detection [17].

Another, even more interesting aspect of the method in the context of a subsequent OCR system, is the line information that is already computed. This information can then be used to extract the lines and feed them to a line recognizer.

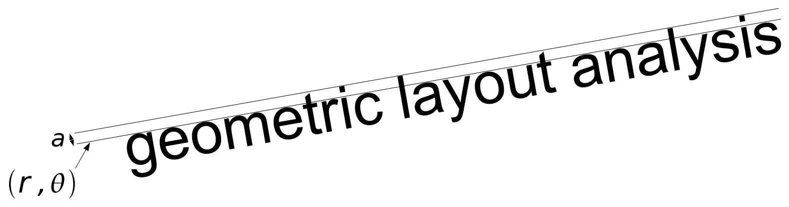


Fig. 3 An illustration of a text-line ascender model. The ascender line and the x-height line are computed from the middle of the top line of the bounding boxes of the connected components.

3.4 Performance Considerations

In a high volume digitization setup, computation time is an important aspect. Although the computation time needed by the text-line extraction process is justifiable in the context of page segmentation, running the extraction 4 times might be too slow if one is only interested in the orientation of the page. Furthermore, the line extraction converges only slowly in the case of pages that have only few aligned components, as it is the case with 90° and 270° misoriented pages. This computational overhead may render the method impractical.

Therefore, in order to speed up the proposed approach, a fast pre-classification is done on the basis of the aspect ratio of the bounding boxes of the connected components. Measurements on correctly aligned document images showed that the ratio of the number of bounding boxes with an aspect ratio $ar > 1$ to the number of bounding boxes with an aspect ratio $ar < 1$ is about 3.5 : 1. We use this to discard two possible orientations from the process. The distribution of this ratio can be seen in Figure 5. It can be seen that a strict decision at the boundary of $ar = 1$ leads to some misclassifications. To avoid this, a margin has been defined for which all four orientations are analyzed. According to the measurements in Figure 5 this was set to $0.66 < ar < 1.5$.

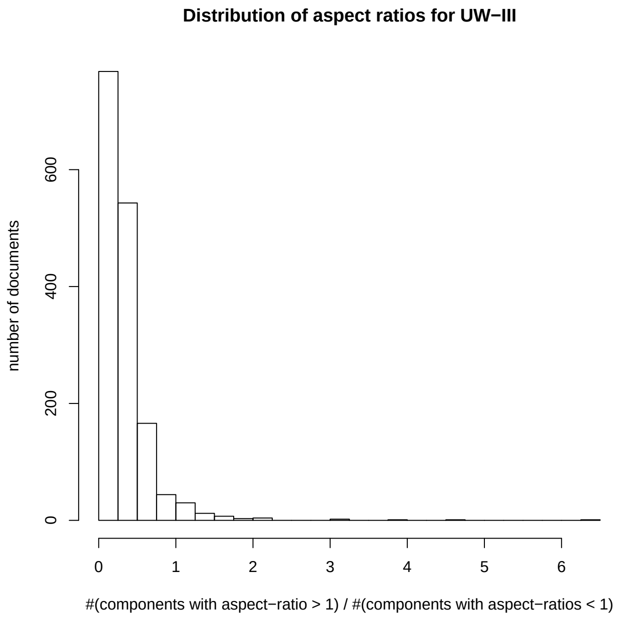


Fig. 5 Histogram of the aspect ratio of connected components of UW-III documents. For each document the number of connected components with aspect ratio < 1 was divided by the number of components with aspect ratio > 1.

4 Evaluation

The evaluation of the proposed method is divided into two main parts: first, the skew estimation accuracy is evaluated. The second part of the evaluation is dedicated to measure the performance on orientation detection.

The dataset used for adapting the method and the parameters is composed of 159 images of the UW-III [14] dataset. The total size of the UW-III dataset is 1600 images. Every 10th image (in alphabetical order) from the total dataset was included. Hence the training set consists of images A00A, A00K, . . . , W1UA. Image W0H4 was manually removed as it contains text in two different orientations. In order to have equal distribution over all orientations, the first image in alphabetical order was left unchanged, the second was rotated by 90°, the third one by 180°, etc.

The main evaluation of the system's performance was done on UW-I dataset. The UW-I dataset consists of 979 binarized images containing scans of scientific journals. Text is written in Latin script and the language used in the journal articles is English.

Bloomberg's [13] measurement on the frequency of ascenders to descenders in English text obtained a ratio of three to one. This is sustained by our measurements for several different languages that can be found in Table 4. It can thus be concluded that the proposed method will work on most languages using Latin Script.

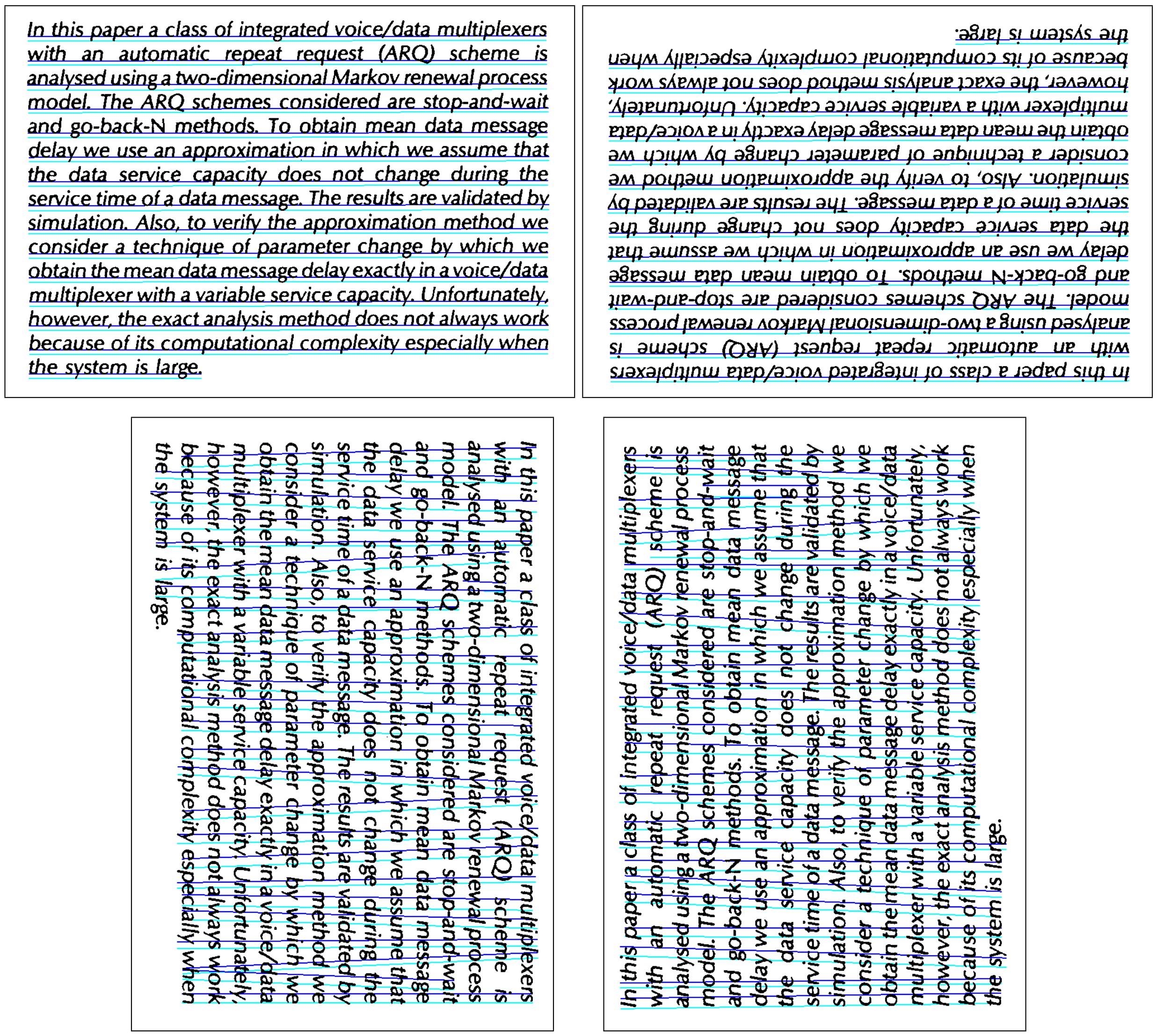

In this paper a class of integrated voice/data multiplexers with an automatic repeat request (ARQ) scheme is analysed using a two-dimensional Markov renewal process model. The ARQ schemes considered are stop-and-wait and go-back-N methods. To obtain mean data message delay we use an approximation in which we assume that the data service capacity does not change during the service time of a data message. The results are validated by simulation. Also, to verify the approximation method we consider a technique of parameter change by which we obtain the mean data message delay exactly in a voice/data multiplexer with a variable service capacity. Unfortunately, however, the exact analysis method does not always work because of its computational complexity especially when the system is large.

Fig. 4 An example figure illustrating the results of fitting descender line model on all four orientations of the image. The blue line represents the baseline, whereas the cyan line represents the descender line. Note that the model fits well on only two of the four orientations. Among these two the one in the correct orientations gets more quality due to a fewer number of descenders.

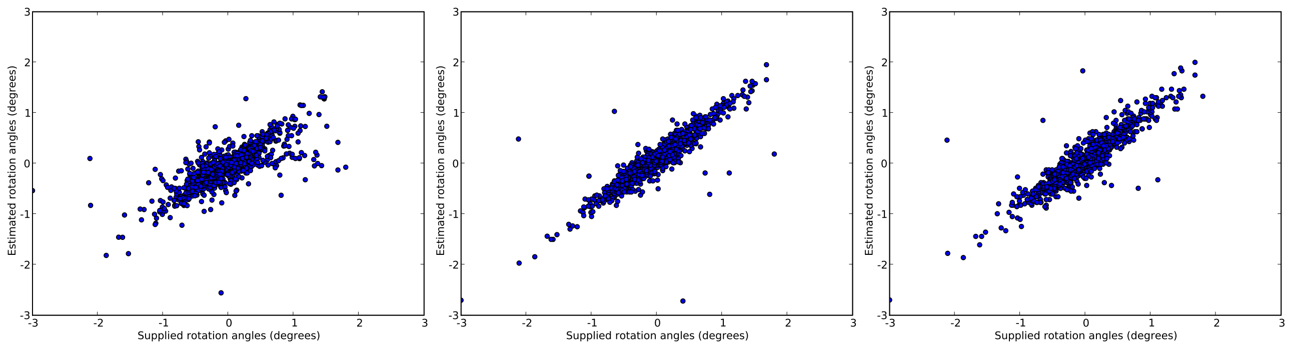
4.1 Skew Detection Accuracy

In this experiment, we evaluated the accuracy of the skew detection capability of our method on the UW-I dataset. For this purpose we chose to use the slope/skew information of the best line (line with the highest quality defined by Equation 2) only. Other possibilities, like the median, mean, weighted median, or weighted mean, of all the different rotations of lines have been reported to give similar results [17]. Using only one line also makes the skew detection algorithm fast. Figure 6 shows a scatter plot of the detected rotation angles and the ground-truth rotation angles as supplied with the UW-I dataset. For the sake of comparison, we also evaluated the skew detection accuracy of Tesseract [5] and Leptonica [13] document analysis systems. It is clear from the plot that we are able to accurately find the skew angles of most document images. Some outliers are due to pages having only formulas or images in them so no reliable text-line could be found.

Page rotations in the UW-I dataset were estimated using a triangulation scheme on multiple sets of three points on each document page [20]. Since multiple sets of points were used there is an associated standard deviation for each estimated page rotation angle. Figure 7 shows a plot of the histogram of standard deviations of skew angles within one page in UW-I dataset. Hence there is no single page rotation angle that can be called “correct” for these pages. Also for many pages, there is a significant difference in the skew of lines that are

Language	Total (in million)	Ascenders	Descenders	Ratio
English	2150	19.9%	7.2%	2.8
French	256	12.5%	7.1%	1.7
Finnish	64	14.9%	6.9%	2.1
Dutch	45	17.4%	7.9%	2.2
German	41	20.9%	5.2%	4.0
Spanish	30	14.5%	7.6%	1.9
Italian	11	13.5%	7.0%	1.9
Swedish	8	19.7%	8.4%	2.3
Portuguese	8	12.6%	7.0%	1.8

Table 1 Measurements of ascender to descender ratios for different languages on the Project Gutenberg dataset. The first column gives analyzed language, the second the total number of analyzed characters. The third and fourth column contain the percentage of ascender and descender characters and the last column contains the ascender to descender ratio.



(a) Tesseract (b) Leptonica (c) This Paper

Fig. 6 Scatter plot of supplied rotation angles of the UW-I dataset and skew angles estimated by (a) Tesseract open source OCR system, (b) Leptonica open source image processing library, and (c) method presented in this paper.

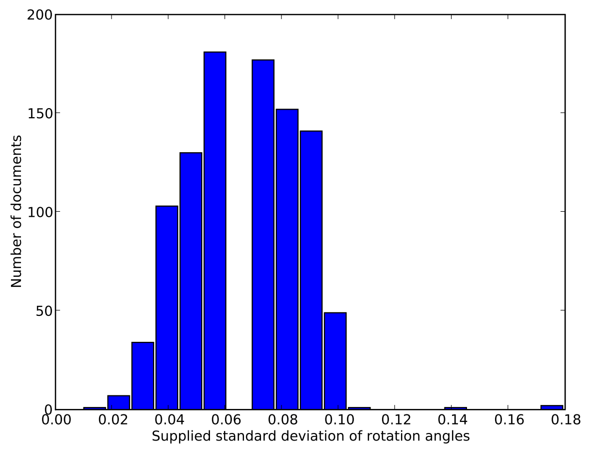


Fig. 7 Histogram of the distribution of standard deviation of skew angles within one page in UW-I dataset.

near the top of the page and the lines that are near the page bottom. This means that the variance between estimated and ground-truth page rotations may simply be due to the intrinsic variability of baseline orientations on these pages. If we consider a difference of less than 0.5 degrees between the estimated and the supplied rotation angle as correct, rotation angles of 960 out of 979 documents are correctly identified resulting in an accuracy of 98%. Using the same criteria, Leptonica correctly identified skew of 966 (98.7%) documents, whereas Tesseract identified skew of 895 (91.4%) documents correctly.

4.2 Orientation Detection Accuracy

Four different tests have been run to evaluate the orientation detecting performance:

1. The overall performance of the full system is analyzed, thus using four line extraction steps. The main parameter that influences the speed is the number of extracted lines. In case that only skew and orientation information is needed, not all lines need to be extracted. The effect of the number of extracted lines on the accuracy and the running time has been analyzed.
2. The performance of the system using the preliminary classification on the aspect ratios of the connected components is measured. Accuracy and processing time versus the number of extracted lines is analyzed.
3. The performance comparison of the proposed approach to other methods is done.

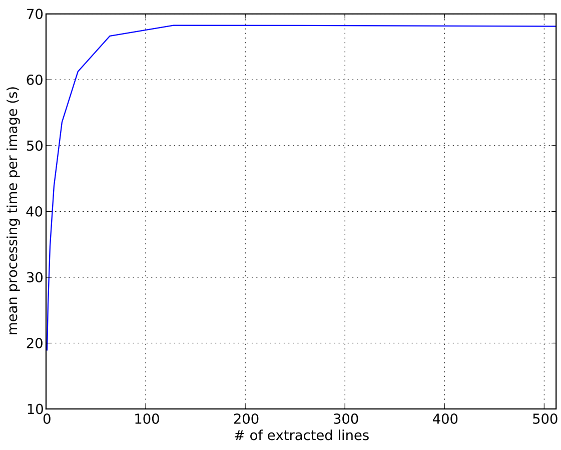


Fig. 8 Plot of the processing time vs. the number of extracted lines for the method using text-line extraction for all four orientations.

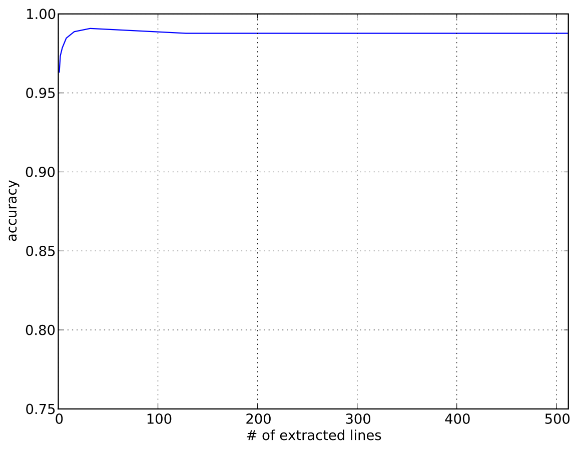


Fig. 9 Plot of the accuracy vs. the number of extracted lines for the method using text-line extraction for all four orientations.

4. As a part of ongoing work, we evaluated the proposed method on a different script. Preliminary results are presented here.

4.2.1 Orientation Detection Using Four Line Extraction Steps

In this evaluation, the performance of the orientation and skew detection method that analyzes the text-lines for all four orientations is presented. The test was run on the UW-1 dataset composed of 979 binary document images with corresponding ground truth. It contains a large variety of document images of scientific journal pages with different fonts and layouts, many of them including images, graphs and mathematical formulas.

For each image, the output of the orientation detection system was compared to the ground truth. The accuracy of the method is defined as the number of correctly detected orientations divided by the total number of test images. The following test parameter set for the maximum number of extracted lines is used:
$\{1, 2, 4, 8, 16, 32, 64, 128, 256, 512\}$

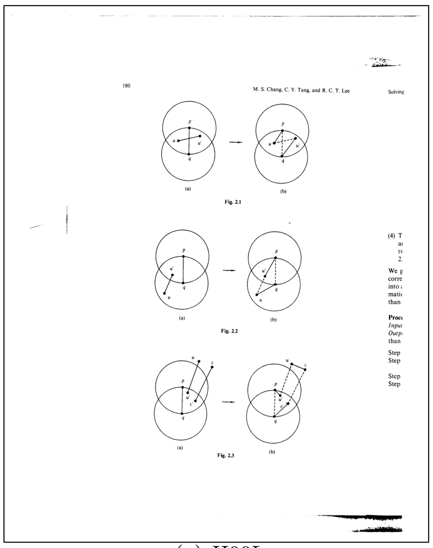
The plot in Figure 8 shows the computation time needed with respect to the number of extracted text-lines. Here it can be seen, that the processing time increases with the number of text-lines to be extracted.

In the plot in Figure 9 the accuracy of the method in relation with the number of extracted lines is shown. It can be seen, that the accuracy for very few lines is already quite high, that it then reaches a maximum around 32 extracted lines and then decreases slightly to finally stay at a constant level. The absolute difference in correctly detected orientations between extracting 32 and 512 lines is only three documents. A manual inspection of different failures showed, that extracting more lines on documents with only few text-lines can lead to misdetection of the orientation. Examples can be found in Figures 10(a) and 10(b). It can also be noted that for some documents extracting more lines is advantageous. The orientation of image *A00M* (see Figure 10(c)), e.g. is recognized correctly when 512 lines are extracted but not if only 32 lines are extracted. In that case a rotation angle of 90° is detected, due to the long nicely structured tables.

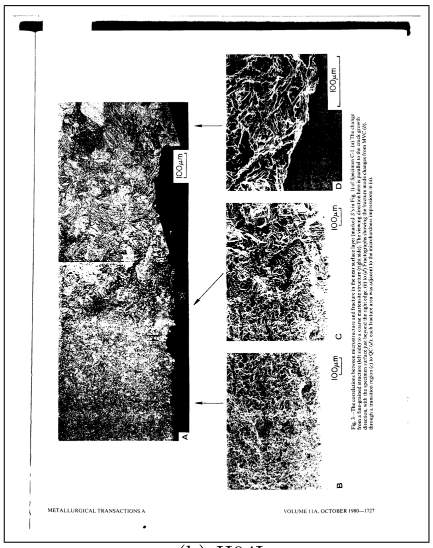
Finally, the choice of $\epsilon = 5$, which was motivated in Section 3.1 is validated empirically. In Figure 11, the accuracy versus different values of ϵ is shown. The tested values for:
$\epsilon \in \{0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30\}$
It can be seen that for values of ϵ between $[2 - 20]$, the performance stays relatively stable at a high value. For values of $\epsilon > 20$, the performance drops dramatically.

4.2.2 Orientation Detection Using Two Line Extraction Steps

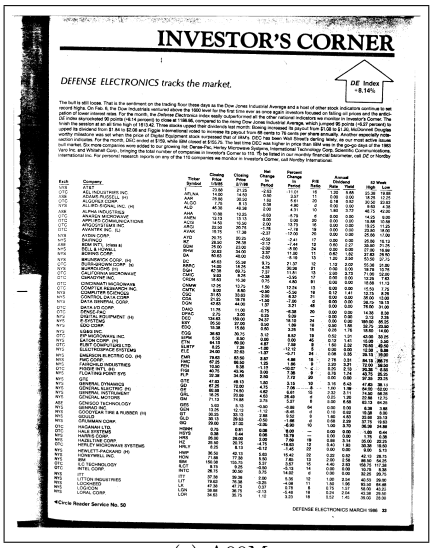
In this experiment, the performance of the orientation and skew detection method using the pre-classification on the basis of the aspect ratio of the connected components is analyzed. Just as before, the 979 images of the UW-I dataset are used for this test. The accuracy is defined as the number of correctly detected page orientations divided by the total number of analyzed document images. The decision whether an orientation is correct or not is made using the ground truth information. The following test parameter set for the maximum number of extracted lines is used:
$\{1, 2, 4, 8, 16, 32, 64, 128, 256, 512\}$



(a) H00L



(b) H04I



(c) A00M

Fig. 10 Examples of images that lead to errors depending on the number of text-lines extracted. Orientations of images *H00L* and *H04I* are correctly recognized when extracting only 32 lines, but not if at maximum 512 lines are extracted. Image *A00M* is recognized correctly for 512 extracted lines but not for 32 extracted lines.

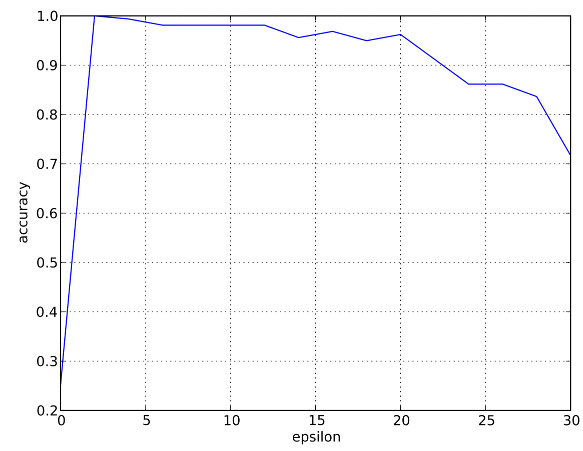


Fig. 11 Plot of the accuracy vs. the parameter ϵ for the method using text-line extraction for all four orientations.

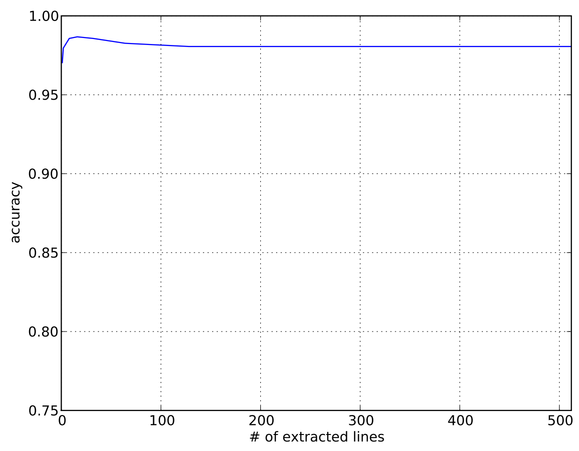
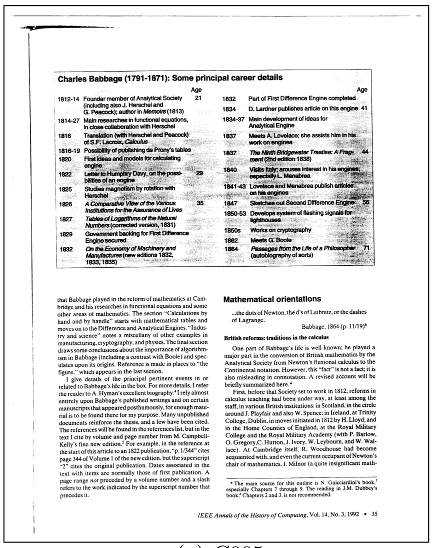
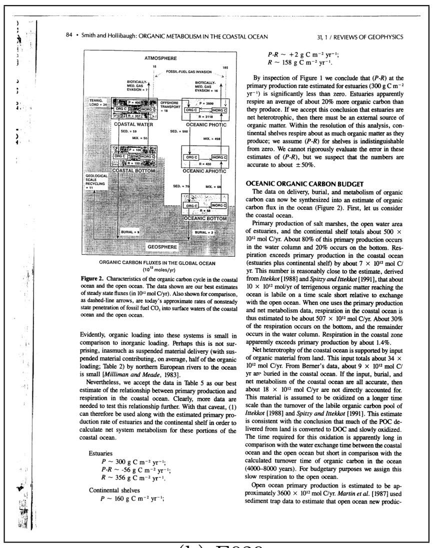


Fig. 13 Plot of the accuracy vs. the number of extracted lines for the method using pre-classification on the bases of aspect ratio of the connected component.

The plot in Figure 13 shows the accuracy of the method in relation to the number of extracted lines. We observe again, that there is a peak for a moderate number of extracted lines, in this case 16. Then the accuracy drops slightly but stays stable afterwards. A manual inspection was done to find out how many documents were wrongly classified on the aspect ratio of the connected components: it showed that only 7 images were misclassified on the basis of their aspect ratios of the connected components. Examples can be found in Figure 12. It can be noticed, that all images present areas where due to binarization many small connected components are present. These are likely to disturb the statistics of the total page.

The plot in Figure 14 shows the time consumption of the method in relation with the number of extracted text-lines. As expected, it can be noted that extracting less text-lines speeds up the method. This information, in combination with the above mentioned accuracy considerations allow the user to specify the parameters according to his needs: if only orientation and skew detection are needed, only very few lines have to be extracted. If the full text-lines are needed anyway


(a) C005


(b) E029

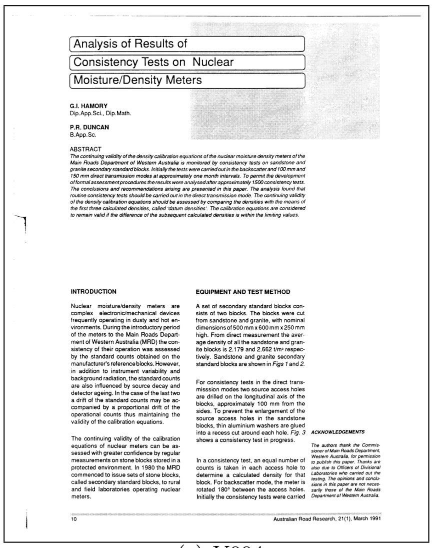

(c) V004

Fig. 12 Example images whose orientations are misclassified on the basis of the aspect ratios of the connected components. It can be seen that all images show areas where many small connected components are present.

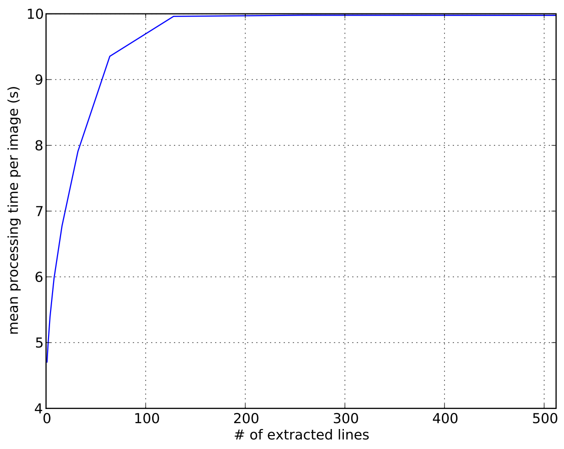


Fig. 14 Plot of the processing time vs. the number of extracted lines for the method using pre-classification on the bases of aspect ratio of the connected component.

for subsequent processing steps, then all lines can be extracted with nearly the same overall performance.

Comparing the performance of the full method to the method using fast pre-classification shows that the gain in running time is significant, around the factor of seven. The accuracy however drops only slightly, approximately about 0.7%.

4.2.3 Performance Comparison on UW-I

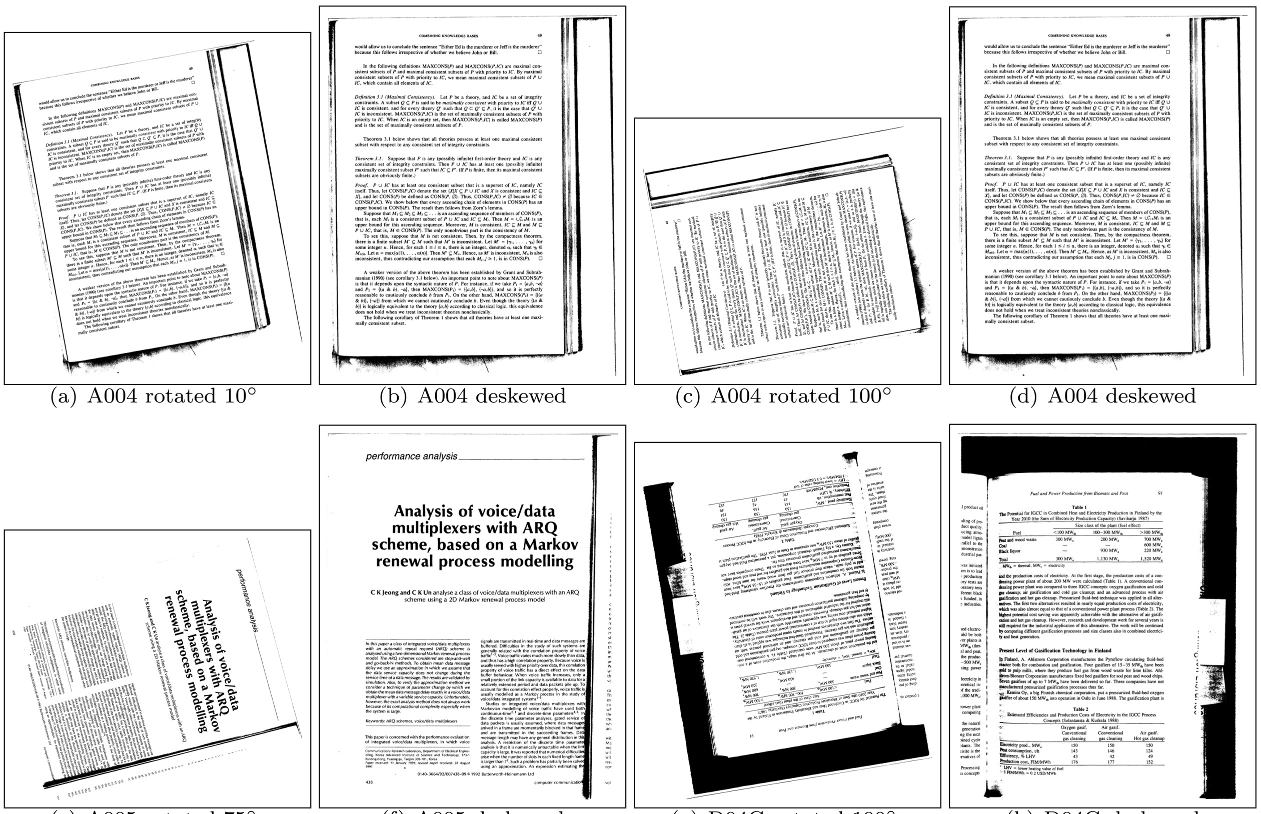
In this part, we compare our method to a previously published method by Bloomberg [13]. For this evaluation, we followed the test setup used by Bloomberg, which is the same as has been used in the two previous tests: the document images from the UW-I dataset are fed to the orientation detection system and the detected orientation is compared to the ground truth orientation. Examples of resulting images of the one-step skew and orientation detection can be found in Figure 15. Our method was set to extract at maximum 512 lines, which for normal documents should cover all lines.

Table 4.2.3 shows the results per orientation and a comparison to Bloomberg's method. It shows that although Bloomberg's method works quite well, our approach achieves even better results. From 979 images in the data set only 9 could not be classified correctly.

A manual verification of the errors showed the following problems:

– Upper case letters only: in image *N02J* and *N03G* no lower case letters were present, leading to a false classification as 180° rotated images. So the ascender to descender ratio can not be used. For the same reason *D067* has been classified as 270° instead of 90°.
– Long columns: in image *A00M* the main part of the page consists of long lists of numbers in a table. This lead to the misclassification to 90° orientation although it is a 0° rotated page.
– Mathematical symbols: pages consisting mainly of mathematical symbols are also hard to be oriented correctly as line finding will not find many good lines. This holds for *A002* (classified as 270° instead of 90°) and *H00Y* (classified as 180° instead of 0°).

Examples of these failures can be found in Figure 16. A performance comparison on single pages showed, that


(a) A004 rotated 10° (b) A004 deskewed (c) A004 rotated 100° (d) A004 deskewed

(e) A005 rotated 75° (f) A005 deskewed (g) D04G rotated 190° (h) D04G deskewed

Fig. 15 Examples for the output of the proposed orientation and skew detection algorithm.

Bloomberg's method is about twice as fast as our method using preclassification and only one extracted line. The comparison however is not too significant: on the one hand, different code bases are used. Especially in the case of Leptonica, the code base has been very well optimized. On the other hand, the nature and also the resulting information (orientation versus orientation, skew angle and line extraction) obtained by both methods widely differ.

4.2.4 Resolution Independence

In this setup, the resolution independence was tested. Therefore we used a set of images that are available with OCRopus [6] open source OCR [2]. These images are synthesized from electronic documents using the following resolution settings: 150, 200, 300 and 400 dpi. Nine different images are present in four different resolutions. For each image four different orientations were tested, leading to a total test set size of 144 images. These images were chosen as they can be freely obtained on the web, so that they can be used for comparison by other researchers.

[2] http://ocropus.googlecode.com/files/ocropus-0.2-test-images.tar.gz

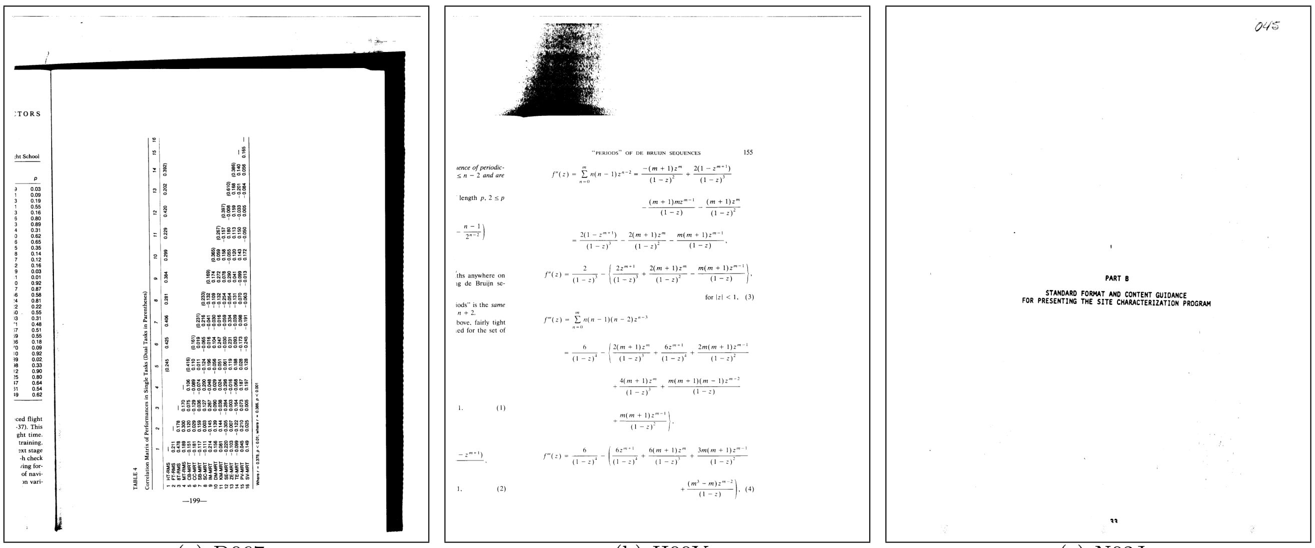
The results for the second test showed a 100% success rate on the 144 test images. This good result is favored by the test images containing no disturbing elements like images, numbers or mathematical symbols.

The same test was run for Bloomberg's method. The results can be found in Table 4.2.4. It shows that on high resolutions (400dpi) it is not able to return high confidences for the obtained orientation estimates. Inspection of the intermediate images of Bloomberg's method shows problems for the dilation using the horizontal structuring element for pages with 90° and 270° rotation. This is likely due to the too small horizontal structuring element.

4.2.5 Application to different Scripts

As a part of the ongoing work, the method was run on a subset of the UW-II dataset consisting of 477 binary images of Japanese journals. The images of UW-II present small skew angles which were not removed beforehand. Example images can be found in Figure 17. The images were rotated in the same way as done for the test on the OCRopus dataset.

The idea to extend the method to other scripts bases on the following idea: most scripts known to the authors



045

PART 8
STANDARD FORMAT AND CONTENT GUIDANCE
FOR PRESENTING THE SITE CHARACTERIZATION PROGRAM

(a) D067 (b) H00Y (c) N02J

Fig. 16 Examples of failures. Image *D067* shows a page being disoriented by our method due to the many lines consisting only of numbers. Image *H00Y* shows another page where our method failed due to mathematical formulas. Finally image *N02J* shows an image that fails to be oriented correctly due to the capitalized text.

	Results from [13]		Our Results (Full)		Our Results (Fast)		Ground-Truth
Orientation	Found	Correct	Found	Correct	Found	Correct	GT
top up	936	935	963	962	955	955	970
top left	2	2	5	5	6	5	9
top down	0	0	8	0	8	0	0
top right	0	0	3	0	10	0	0
No result	41	0	0	0			–
Total Correct		95.8%		98.8%		98.0%	–

Table 2 A comparison of orientation detection results on UW-I dataset. The table shows that our method detects the right orientation for a larger number of documents than the Bloomberg's method.

Resolution	Total	BB [13] Correct	BB [13] No result	Proposed Method
150dpi	36	36	0	36
200dpi	36	36	0	36
300dpi	36	36	0	36
400dpi	36	27	9	36
Total Correct		93.8%		100.0%

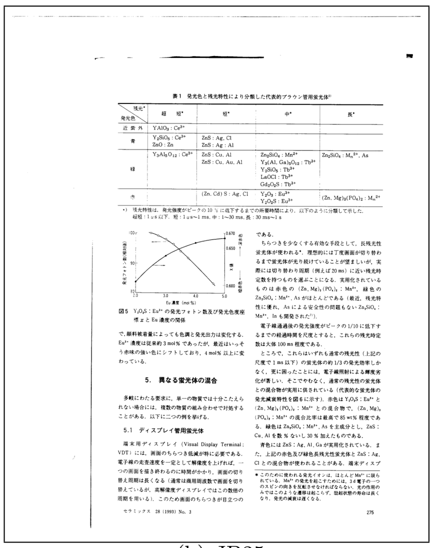
Table 3 Results of Bloomberg's [13](BB) and the proposed method on test images with different resolution. The results show that Bloomberg's method failed to find a correct orientation for some documents rendered at 400dpi. Our Method found the correct orientation in all cases yielding a 100% result in this test.

have the concept of a base line. Most characters should thus be aligned to the baseline. There might be one or even more ascender lines, but under the assumption, that less components touch the ascender line than the descender line, it seems reasonable to expect the proposed method to work reasonable on these scripts.

Only the results for the full approach are given. The plot in Figure 18 shows the dependence of accuracy on the number of extracted lines.

The results of the test on the Japanese images can be found in Table 4. It shows the confusion matrix between the ground truth orientation and the detected orientation. Although the line model was not designed for Japanese script, the results are still reasonable with an overall accuracy of 85.7%. The reason for this lies in the fact that in Japanese text, characters line up on the baseline, whereas due to varying heights of different characters, alignment of top-line is not so strong. Hence top line gets a lower score in the least square fit that the baseline giving a good hint about the correct orientation of the page.

(a) JA00


(b) JB25

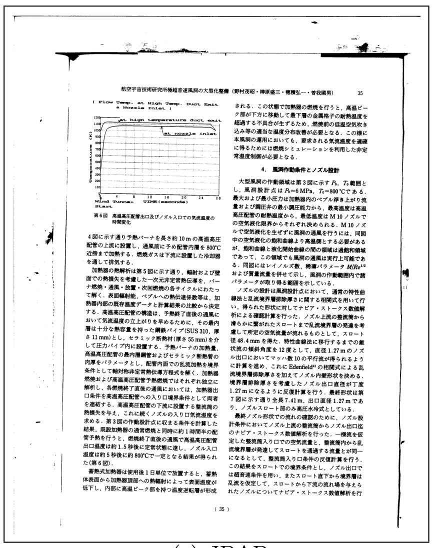

(c) JBAB

Fig. 17 Examples of Japanese script document images from the UW-II dataset.

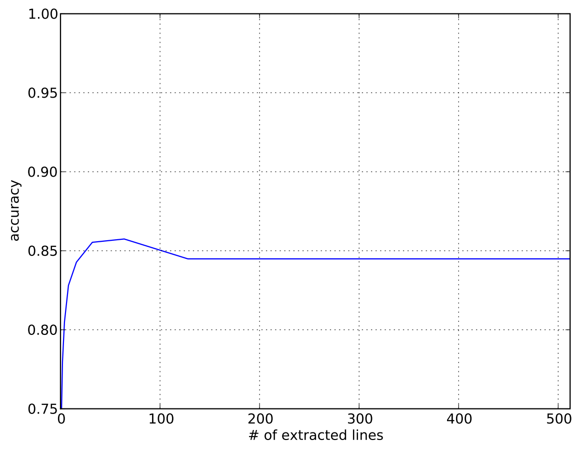


Fig. 18 Plot of the accuracy vs. the number of extracted lines for the full method on Japanese document images from the UW-II dataset.

	0 °	90 °	180 °	270 °
0 °	113	0	7	0
90 °	1	93	0	25
180 °	7	0	112	0
270 °	1	27	0	91
Total accuracy: 85.7%				

Table 4 Confusion matrix of accuracies for our approach on 477 Japanese documents from the UW-2 dataset (64 extracted lines).

5 Conclusion

In this paper we presented a one step method for skew and orientation detection using a line extraction algorithm. We showed the effectiveness of the method on a publicly available dataset. A comparison to a widely used open source orientation detection technique was done. Experiments showed that our method outperformed the analyzed method for both UW-I dataset and our own dataset having documents rendered at different resolutions. A particular advantage of our method is that we get both orientation and skew estimates in one step. We plan to make our implementation publicly available as a part of OCRopus open source OCR system.

6 Acknowledgments

This work was partially funded by the BMBF (German Federal Ministry of Education and Research), project PaREn (01 IW 07001).

References

1. L. Vincent, Google book search: Document understanding on a massive scale, in: Proc. of the 9th Int. Conf. on Document Analysis and Recognition [21], pp. 819–823.
2. F. Shafait, J. van Beusekom, D. Keysers, T. M. Breuel, Document cleanup using page frame detection, Int. Jour. on Document Analysis and Recognition 11 (2) (2008) 81–96.
3. F. Shafait, D. Keysers, T. M. Breuel, Efficient implementation of local adaptive thresholding techniques using integral images, in: Proc. of SPIE Document Recognition and Retrieval XV [22], pp. 681510–681510.
4. F. Shafait, D. Keysers, T. M. Breuel, Performance evaluation and benchmarking of six page segmentation algorithms, IEEE Trans. on Pattern Analysis and Machine Intelligence 30 (6) (2008) 941–954.
5. R. Smith, An overview of the Tesseract OCR engine, in: Proc. of the 9th Int. Conf. on Document Analysis and Recognition [21], pp. 629–633.

6. T. M. Breuel, The OCRopus open source OCR system, in: Proc. of SPIE Document Recognition and Retrieval XV [22], pp. 0F1–0F15.
7. R. Cattoni, T. Coianiz, S. Messelodi, C. M. Modena, Geometric layout analysis techniques for document image understanding: a review, Tech. Rep. 9703-09, IRST, Trento, Italy (1998).
8. Y. Lu, C. Tan, A nearest-neighbor chain based approach to skew estimation in document images, Pattern Recognition Letters 24 (14) (2003) 2315–2323.
9. R. S. Caprari, Algorithm for text page up/down orientation determination, Pattern Recognition Letters 21 (4) (2001) 311–317.
10. H. Aradhye, A generic method for determining up/down orientation of text in roman and non-roman scripts, Pattern Recognition 38 (11) (2005) 2114–2131.
11. S. Lu, C. L. Tan, Automatic document orientation detection and categorization through document vectorization, in: Proc. of the 14th ACM Int. Conf. on Multimedia, New York, NY, USA, 2006, pp. 113–116.
12. D. Le, G. Thoma, H. Wechsler, Automated page orientation and skew angle detection for binary document images, Pattern Recognition 27 (10) (1994) 1325–1344.
13. D. S. Bloomberg, G. E. Kopec, L. Dasari, Measuring document image skew and orientation, in: Proc. of SPIE Document Recognition and Retrieval II, San Jose, CA, USA, 1995, pp. 302–316.
14. I. T. Phillips, User's reference manual for the UW english/technical document image database III, Tech. rep., Seattle University, Washington (1996).
15. B. T. Ávila, R. D. Lins, A fast orientation and skew detection algorithm for monochromatic document images, in: Proc. of the 5th ACM symposium on Document engineering, New York, NY, USA, 2005, pp. 118–126.
16. S. Lu, J. Wang, C. Tan, Fast and accurate detection of document skew and orientation, in: Proc. of the 9th Int. Conf. on Document Analysis and Recognition [21], pp. 684–688.
17. T. M. Breuel, Robust least square baseline finding using a branch and bound algorithm, in: Proc. of SPIE Document Recognition and Retrieval IX, San Jose, CA, USA, 2002, pp. 20–27.
18. T. M. Breuel, A practical, globally optimal algorithm for geometric matching under uncertainty, Electronic Notes in Theoretical Computer Science 46 (2001) 1–15.
19. T. M. Breuel, Implementation techniques for geometric branch-and-bound matching methods, Computer Vision and Image Understanding 90 (3) (2003) 258–294.
20. S. Chen, M. Y. Jaisimha, J. Ha, R. M. Haralick, I. T. Phillips, Reference manual for the UW english document image database, Tech. rep., Seattle University, Washington (1993).
21.
22. Proc. of SPIE Document Recognition and Retrieval XV, Vol. 6815.