Resolution Independent Skew and Orientation Detection for Document Images

Joost van Beusekom[b], Faisal Shafait[a], Thomas M. Breuel[a, b]

[a]Image Understanding and Pattern Recognition (IUPR) Research Group
German Research Center for Artificial Intelligence (DFKI) GmbH
D-67663 Kaiserslautern, Germany
joost@iupr.dfki.de, faisal@iupr.dfki.de
[b]Department of Computer Science, Technical University of Kaiserslautern
D-67663 Kaiserslautern, Germany
tmb@informatik.uni-kl.de

ABSTRACT

In large scale scanning applications, orientation detection of the digitized page is necessary for the following procedures to work correctly. Several existing methods for orientation detection use the fact that in Roman script text, ascenders are more likely to occur than descenders. In this paper, we propose a different approach for page orientation detection that uses this information. The main advantage of our method is that it is more accurate than compared widely used methods, while being scan resolution independent. Another interesting aspect of our method is that it can be combined with our previously published method for skew detection to have a single-step skew and orientation estimate of the page image. We demonstrate the effectiveness of our approach on the UW-I dataset and show that our method achieves an accuracy of above 99% on this dataset. We also show that our method is robust to different scanning resolutions and can reliably detect page orientations for documents rendered at 150, 200, 300, and 400 dpi.

Keywords: Orientation detection, line finding

1. INTRODUCTION

A number of initiatives have recently been launched for large scale scanning of documents, books and other paper-based materials.[1] Among noise removal and binarization, skew and orientation detection plays an important role: if document images are wrongly oriented, the subsequent processing methods will fail to work correctly since usually both layout analysis[2] and OCR[3, 4] assume pages to be in the correct orientation.

The problem of orientation detection deals with finding the correct orientation of the page so that the characters are in an upright position. Four main orientations are considered: top up, top left, top down and top right. Skew detection, a similar yet different problem deals with finding the rotation angle of a document in a given interval of angles.

Many methods exist for skew estimation. Cattoni et al.[5] give a good overview of the state of the art methods back in 1998. All methods presented there allow at most only rotation angles between $-90\,^\circ$ and $+90\,^\circ$. These methods are not designed to be used to determine if a page is top down or not.

In 1994 Le et al.[6] present a system capable only of detection of portrait or landscape mode images. Top down document images are not considered. It uses rules based on projection profiles for orientation detection. Hugh 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 and obtains an error rate of 0.1%.

In 1995 Bloomberg et al.[7] presented a method for orientation detection that uses the ascender to descender ratio of the English language. Ascenders and descenders are extracted using morphological operations. The reported error rate on UW-I[8] dataset is 1 wrongly detected orientation versus 938 correctly detected. The

Figure 1. An illustration of Roman script text-line model proposed by Breuel.[12] 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.

orientation of 41 documents could not be extracted but are not considered as errors. An implementation of this method is available in the Leptonica*, an open source image processing library.

Another method using the ascender/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.

A more recent method using the ascender/descender ratio for orientation detection is presented by Avila et al.[10] 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. The reported error rate for orientation detection is below 0.1% on a non-public dataset.

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 an success rate of 98.2% for documents with at least 12 text lines.

In this paper we propose a new method for combined skew and orientation detection using geometric modeling of Roman 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.

The rest of the paper is organized as follows: Section 2 explains the line finding method and its use for page orientation detection. In Section 3 experimental results are discussed. The paper is concluded by Section 4.

2. SKEW AND ORIENTATION DETECTION

The presented method for page orientation detection is based on Roman script text-line model by Breuel.[12] We will first illustrate the geometric text-line model since it is crucial for the understanding of this work.

2.1 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.

Based on this geometric model of Roman script text-lines, we use geometric matching to extract text-lines from scanned documents as in.[12] 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[13,14] is

*http://www.leptonica.com

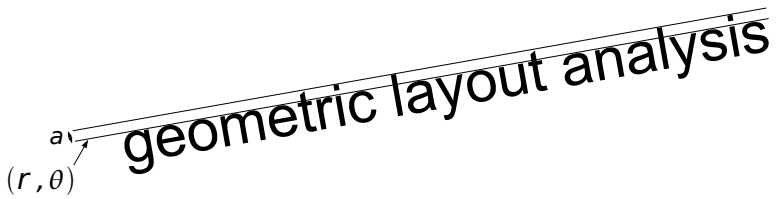


Figure 2. An illustration of a text-line ascender model. The ascender line and the x-height line are computed from the middle of the top line of the bounding boxes of the connected components.

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[12] is:

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

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 such that a reference point has more chances to match 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 $d_{(r,\theta)}(x) \geq \epsilon$. 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. 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 extract 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.

2.2 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 2. 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.

If we now 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 $\{y_1, y_2, \cdots, y_n\}$:

$$\hat{\vartheta} := \arg\max_{\vartheta} Q_{y_1^n}(\vartheta) \tag{4}$$

The quality function can then be defined as:

$$Q_{y_1^n}(\vartheta) = Q_{y_1^n}(r, \theta, a) = \sum_{i=1}^{n} \max(q_{(r,\theta)}(y_i), \alpha q_{(r+a,\theta)}(y_i)) \tag{5}$$

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

Since in Roman script ascenders are more likely to occur than descenders, more components will match to the ascender line than to the descender line. A component matching to descender/ascender gets a lower score (due to the factor α in Equations 2 and 5) 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 5). 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 3.

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.[12]

3. EXPERIMENTS AND RESULTS

Two different tests were done to measure the performance of the orientation detection performance of our approach. For the deskewing performance of the presented approach we refer the reader to Breuel's paper.[12] In the first test, we followed the test setup used by.[7] From UW-I dataset 979 binarized images from journal scans were analyzed for their orientation. The angle range for skew detection was set to $[-45\,^\circ, 45\,^\circ]$. The minimum line length was set 30 pixels and at the minimum of least 2 components. The maximum number of lines extracted was set to 50, as this is mostly enough to cover all important lines.

In a second test, the resolution independence was tested. Therefore we used a set of images that are available with OCRopus open source OCR.[4] These images are synthesized from a electronic document using the following resolution settings: 150, 200, 300 and 400 dpi. Nine different images are present in the 4 different resolutions. For each image the 4 different orientations were tested, leading to a total test set size of 144 images.

Figure 4 shows some examples of the one-step orientation and skew detection.

Table 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:

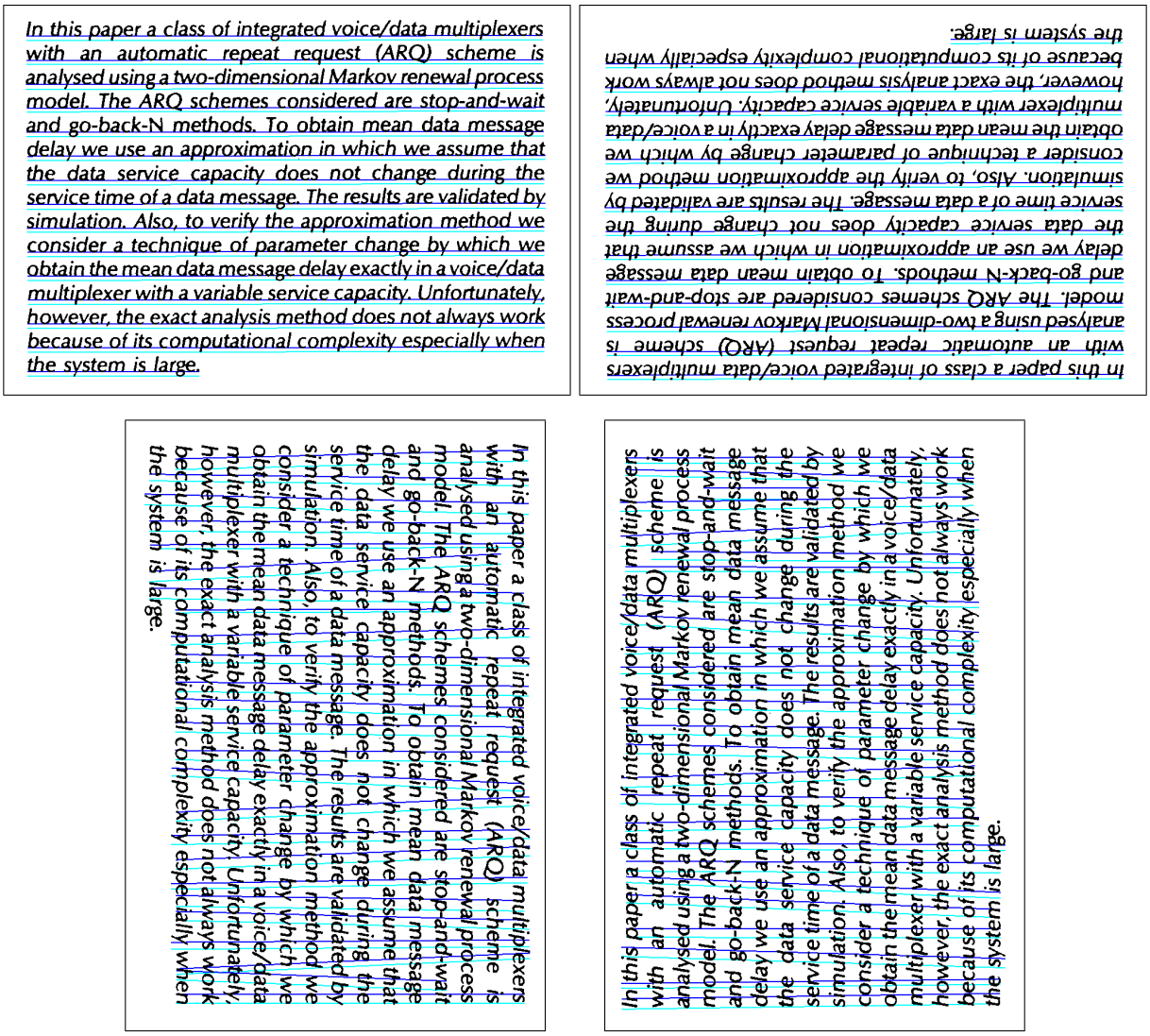
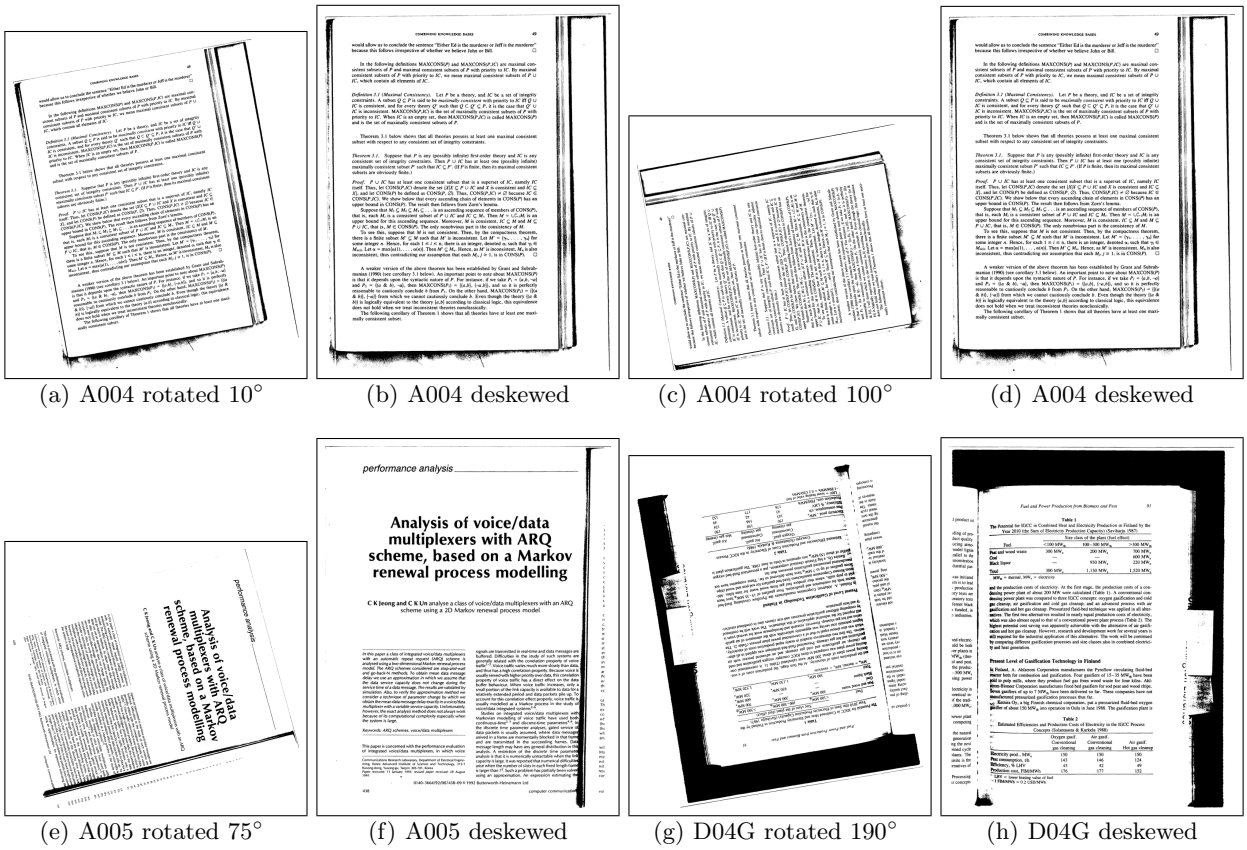


Figure 3. An example figure illustrating the results of fitting descender line model on all four orientations of the image. The blue line represents the baseline, whereas the cyan line represents the descender line. Note that the model fits well on only two of the four orientations. Among these two the one in the correct orientations gets more quality due to a fewer number of descenders.

- Upper case letters only: in image *N02J* and *N03G* no lower case letters were present, leading to a false classification as top-down images. So the ascender/descender ratio can not be used. For the same reason *D067* has been classified as top-right instead of top-left.
- 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 top-left orientation although it is an top-up oriented 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 top-right instead of top-left) and *H00Y* (classified as top-down instead of top-up).

Examples of these failures can be found in Figure 5.

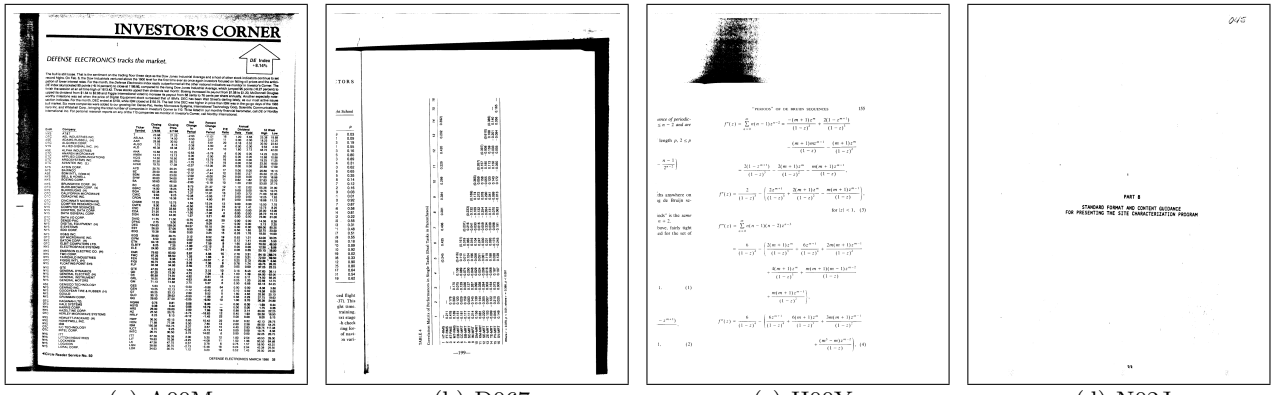
Another weak point in this evaluation is the data set on its own: although the distributions of the orientations



(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

Figure 4. Examples for the output of the proposed orientation and skew detection algorithm.



(a) A00M (b) D067 (c) H00Y (d) N02J

Figure 5. Examples of failures. Image *A00M* shows an example of page with columnar structure leading to a high number of lines with high qualities along the columns of the table. 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 [7]		Our Results		Ground-Truth
Orientation	Found	Correct	Found	Correct	GT
top up	936	935	963	963	970
top left	2	2	8	7	9
top down	0	0	6	0	0
top right	0	0	2	0	0
No result	41	0	0	0	–
Total Correct		95.8%		99.1%	–

Table 1. 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 Correct	BB No result	Proposed Corr.
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 2. Results of Bloomberg's[7](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.

reflect the real-world problem better than a uniform distribution, it is clear that a dummy method only returning top-up as orientation would achieve an accuracy of 98.4%, which is better as Bloomberg's method.

The results for the second 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 3. It shows that on high resolutions (400dpi) it is not able to return high confidences for the obtained orientation estimates.

4. CONCLUSION

In this paper we presented a one step method for skew and orientation detection using a line finding algorithm and a script dependent line model. 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 state-of-the-art 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.

REFERENCES

1. L. Vincent, "Google book search: Document understanding on a massive scale," in *9th Int. Conf. on Document Analysis and Recognition*, pp. 819–823, (Curitiba, Brazil), Sep. 2007.
2. F. Shafait, D. Keysers, and T. M. Breuel, "Performance evaluation and benchmarking of six page segmentation algorithms," *IEEE Trans. on Pattern Analysis and Machine Intelligence* **30**(6), pp. 941–954, 2008.
3. R. Smith, "An overview of the Tesseract OCR engine," in *Proc. 9th Int. Conf. on Document Analysis and Recognition*, pp. 629–633, (Curitiba, Brazil), Sep. 2007.
4. T. M. Breuel, "The OCRopus open source OCR system," in *Proc. SPIE Document Recognition and Retrieval XV*, pp. 0F1–0F15, (San Jose, CA, USA), Jan. 2008.
5. R. Cattoni, T. Coianiz, S. Messelodi, and C. M. Modena, "Geometric layout analysis techniques for document image understanding: a review," Tech. Rep. 9703-09, IRST, Trento, Italy, 1998.
6. D. Le, G. Thoma, and H. Wechsler, "Automated page orientation and skew angle detection for binary document images," *Pattern Recognition* **27**, pp. 1325–1344, October 1994.

7. D. S. Bloomberg, G. E. Kopec, and L. Dasari, "Measuring document image skew and orientation," in *Proc. SPIE Document Recognition II*, pp. 302–316, (San Jose, CA, USA), Feb. 1995.
8. I. T. Phillips, "User's reference manual for the UW english/technical document image database III," tech. rep., Seattle University, Washington, 1996.
9. R. S. Caprari, "Algorithm for text page up/down orientation determination," *Pattern Recognition Letters* **21**(4), pp. 311–317, 2001.
10. B. T. Ávila and R. D. Lins, "A fast orientation and skew detection algorithm for monochromatic document images," in *DocEng '05: Proceedings of the 2005 ACM symposium on Document engineering*, pp. 118–126, (New York, NY, USA), 2005.
11. S. Lu and C. L. Tan, "Automatic document orientation detection and categorization through document vectorization," in *MULTIMEDIA '06: Proceedings of the 14th annual ACM international conference on Multimedia*, pp. 113–116, ACM, (New York, NY, USA), 2006.
12. T. M. Breuel, "Robust least square baseline finding using a branch and bound algorithm," in *Proc. SPIE Document Recognition and Retrieval IX*, pp. 20–27, (San Jose, CA, USA), Jan. 2002.
13. T. M. Breuel, "A practical, globally optimal algorithm for geometric matching under uncertainty," *Electronic Notes in Theoretical Computer Science* **46**, pp. 1–15, 2001.
14. T. M. Breuel, "Implementation techniques for geometric branch-and-bound matching methods," *Computer Vision and Image Understanding* **90**(3), pp. 258–294, 2003.