Page Frame Detection for Marginal Noise Removal from Scanned Documents

Faisal Shafait[1], Joost van Beusekom[2], Daniel Keysers[1], and Thomas M. Breuel[2]

[1] Image Understanding and Pattern Recognition (IUPR) research group
German Research Center for Artificial Intelligence (DFKI) GmbH
D-67663 Kaiserslautern, Germany
faisal.shafait@dfki.de, daniel.keysers@dfki.de
[2] Department of Computer Science, Technical University of Kaiserslautern
D-67663 Kaiserslautern, Germany
joost@iupr.net, tmb@informatik.uni-kl.de

Abstract. We describe and evaluate a method to robustly detect the page frame in document images, locating the actual page contents area and removing textual and non-textual noise along the page borders. We use a geometric matching algorithm to find the optimal page frame, which has the advantages of not assuming the existence of whitespace between noisy borders and actual page contents, and of giving a practical solution to the page frame detection problem without the need for parameter tuning. We define suitable performance measures and evaluate the algorithm on the UW-III database. The results show that the error rates are below 4% for each of the performance measures used. In addition, we demonstrate that the use of page frame detection reduces the optical character recognition (OCR) error rate by removing textual noise. Experiments using a commercial OCR system show that the error rate due to elements outside the page frame is reduced from 4.3% to 1.7% on the UW-III dataset.

1 Introduction

For a clean document, the page frame is defined as the rectangular region enclosing all the foreground pixels in the document image. When a page of a book is scanned or photocopied, textual noise (extraneous symbols from the neighboring page) and/or non-textual noise (black borders, speckles, . . .) appears along the border of the document. The goal of page frame detection is to find the actual page, ignoring the noise along the page border. The importance of page frame detection in document image analysis is often underestimated, although a good page frame detection algorithm can help to improve the performance considerably. Since the state-of-the-art page segmentation algorithms report textual noise regions as text-zones [1], the OCR accuracy decreases in the presence of textual noise, because the OCR system usually outputs several extra characters in these regions. Including page frame detection as a document preprocessing step can thus help to increase OCR accuracy.

The most common approach to eliminate marginal noise is to perform document cleaning by filtering out connected components based on their size and aspect ratio [2–4]. However, when characters from the adjacent page are also present, they usually cannot be filtered out using these features alone. Some approaches try to perform document cleaning as a part of layout analysis [5]. These approaches remove black borders and speckles resulting from photocopy effects with high accuracy but report a number of false alarms [1].

Instead of removing individual components, researchers have also tried to explicitly detect and remove the marginal noise. Le et al. [6] have proposed a rule-based algorithm using several heuristics for detecting the page borders. The algorithm relies upon the classification of blank/textual/non-textual rows and columns, object segmentation, and an analysis of projection profiles and crossing counts to detect the page frame. Their approach is based on the assumption that the page borders are very close to edges of images and borders are separated from image contents by a white space, i.e. the borders do not overlap the edges of an image content area. However, this assumption is often violated when pages from a thick book are scanned or photocopied. Avila et al. [7] and Fan et al. [8] propose methods for removing non-textual noise overlapping the page content area, but do not consider textual noise removal. Cinque et al. [9] propose a method for removing both textual and non-textual noise from greyscale images based on image statistics like horizontal/vertical difference vectors and row luminosities. However, their method is not suitable for binary images.

Here, we propose an algorithm to detect the page frame that can be used to remove both textual and non-textual noise from binary document images. The method does not assume the existence of whitespace between noisy borders and actual page contents, and can locate the page contents region even if the noise overlaps some regions of the page content area. Instead of trying to detect and remove the noisy borders, we focus on using geometric matching methods to detect the page frame in a document image. The use of geometric matching for solving such a problem has several advantages. Instead of devising carefully crafted rules, the problem is solved in a more general framework, thus allowing higher performance on a more diverse collection of documents.

2 Geometric Matching for Page Frame Detection

Connected components, textlines, and zones form different levels of page segmentation. We use a fast labeling algorithm to extract connected components from the document image. In recent comparative studies for the performance evaluation of page segmentation algorithms [1, 10], it is shown that the constrained textline finding algorithm [3] has the lowest error rates among the compared algorithms for textline extraction, and the Voronoi-diagram based algorithm [5] has the lowest error rates for extracting zone-level information. Therefore, we use these two algorithms for extracting textlines and zones from the document image, respectively. After extracting connected components, textlines, and zones, the next step is to extract the page frame from the document image.

Page Frame Detection Given the sets of connected components C, textlines L, and zones Z, we are interested in finding the best matching geometric primitives for the page frame with respect to the sets C, L, and Z. Since the bounding box of the page frame is an axis-aligned rectangle, it can be described by four parameters $\vartheta = \{l, t, r, b\}$ representing the left, top, right, and bottom coordinates respectively. We compute the best matching parameters for the page frame by finding the maximizing set of parameters

$$\hat{\vartheta}(C, L, Z) := \arg\max_{\vartheta \in T} Q(\vartheta, C, L, Z) \tag{1}$$

where $Q(\vartheta, C, L, Z)$ is the total quality for a given parameter set, and T is the total parameter space.

Design of the Quality Function The design of an appropriate quality function is not trivial. We may define the page frame as a rectangle that touches many character bounding boxes on its four sides. The character bounding boxes are obtained from C by filtering out noise and non-text components based on their area and aspect ratio. However, this approach has some limitations:

1. The top and bottom lines do not necessarily touch more characters than other lines in the page (especially when there is only a page number in the header or footer). Also in some cases, there can be non-text zones (images, graphics, ...) at the top or bottom of the page. Hence the parameters t and b can not be reliably estimated using character level information.
2. Changes in text alignment (justified, left-aligned, etc) of a page may result in arbitrary changes in the estimated l and r parameters.

Instead of using connected component level information, we can use textlines. We may define the page frame as a rectangle that touches many line bounding boxes on its two sides, besides containing most of the textlines in the page. In order to search for optimal parameters, we decompose the parameters into two parts: $\vartheta_h = \{l, r\}$ and $\vartheta_v = \{t, b\}$. Although ϑ_h and ϑ_v are not independent, such a decomposition can still be used because of the nature of the problem. We first set parameters ϑ_v to their extreme values ($t = 0, b = H$ where H is the page height) and then search for optimal ϑ_h. This ensures that we do not lose any candidate textlines based on their vertical position in the image. The decomposition not only helps in reducing the dimensionality of the searched parameter space from four to two, but also prior estimates for ϑ_h make the estimation of ϑ_v a trivial task, as we will discuss below. Hence the optimization problem of Equation (1) is reduced to

$$\hat{\vartheta_h}(L) := \arg\max_{\vartheta_h \in T} Q(\vartheta_h, L) \tag{2}$$

We employ the RAST technique [11] to perform the maximization in Equation (2). RAST is a branch-and-bound algorithm that guarantees to find the globally optimal parameter set by recursively subdividing the parameter space and processing the resulting parameter hyper-rectangles in the order given by an upper bound on the total quality. The total upper bound of the quality Q can be written as the sum of local quality functions

$$Q(\vartheta_h, L) := \sum\nolimits_j q(\vartheta_h, L_j) \tag{3}$$

We then compute an upper and lower bound for the local quality function q. Given a line bounding box (x_0, y_0, x_1, y_1), we determine intervals $d(l, x_i)$ and $d(r, x_i)$ of possible distances of the x_i from the parameter intervals l and r, respectively. The local quality function for a given line and a parameter range ϑ_h can then be defined as

$$q_1(\vartheta_h, (x_0, x_1)) = \max\big(0, 1 - \frac{d^2(l, x_0)}{\epsilon^2}\big) + \max\big(0, 1 - \frac{d^2(r, x_1)}{\epsilon^2}\big) \tag{4}$$

Where ϵ defines the distance up to which a line can contribute to the page frame. Textlines may have variations in their starting and ending positions within a text column depending on text alignment, paragraph indentation, etc. We use $\epsilon = 150$ pixels in order to cope with such variations. This quality function alone already works well for single column documents, but for multi-column documents it may report a single column (with the highest number of textlines) as the optimal solution. In order to discourage such solutions, we introduce a negative weighting for textlines on the 'wrong' side of the page frame in the form of the quality function

$$q_2(\vartheta_h, (x_0, x_1)) = -\max\big(0, 1 - \frac{d^2(l, x_1)}{(2\epsilon)^2}\big) - \max\big(0, 1 - \frac{d^2(r, x_0)}{(2\epsilon)^2}\big) \tag{5}$$

the overall local quality function is then defined as $q = q_1 + q_2$ It can be seen that this quality function will yield the optimal parameters for ϑ_h even if there are intermediate text-columns with larger number of textlines. However, if the first or last column contain very few textlines, the column is possibly ignored.

Parameter refinement After obtaining the optimal parameters for ϑ_h in terms of mean square error, we refine the parameters to adjust the page frame according to different text alignments and formatting styles as shown in Figure 1. Hence we obtain an initial estimate for the parameters ϑ_v by inspecting all lines that contribute positively to the quality function $Q(\vartheta_h, L)$ and setting t to the minimum of all occurring y_0-values and b to the maximum of all occurring y_1-values. A page frame detected in this way is shown in Figure 1. This approach gives the correct result for most of the documents, but fails in these cases:

1. If there is a non-text zone (images, graphics, ...) at the top or bottom of the page, it is missed by the page frame.
2. If there is an isolated page number at the top or bottom of the page, and it is missed by the textline detection, it will not be included in the page frame.

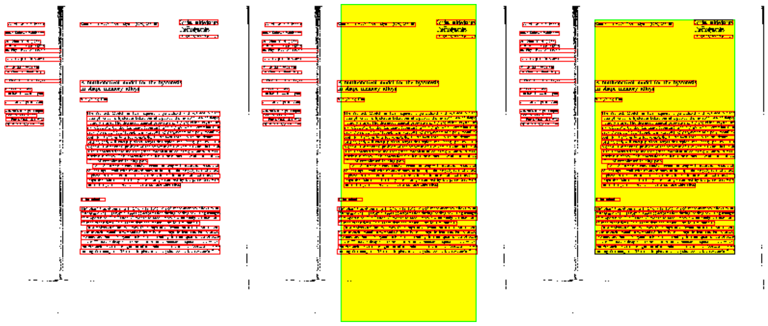


Fig. 1. Example demonstrating refinement of the parameters ϑ_h to adapt to text alignment. The detected textlines are shown in the left image, one paragraph being indented more than the remaining lines. A page frame corresponding to the optimal parameters with respect to Equation 3 is shown in the center. The image on the right shows the initial page frame after adjusting for text alignment.

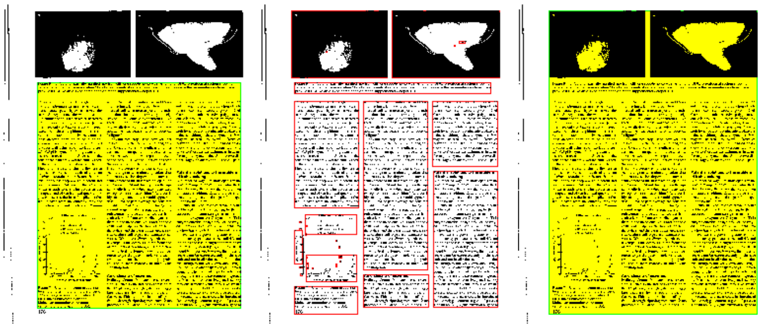


Fig. 2. Example image to demonstrate inclusion of non-text boxes into the page frame. The initially detected page frame based only on the textlines is shown on the left. Note that the images on the top and the page number at the bottom are not part of the page frame. The middle image shows the zones detected by the Voronoi algorithm. The right image shows the final page frame obtained using zone-level information.

An example illustrating these problems is shown in Figure 2. In order to estimate the final values for $\vartheta_v = \{t, b\}$, we use zone level information as given by the Voronoi algorithm [5]. We perform filtering on the zones obtained by the Voronoi algorithm, such that all the zones that lie completely inside, or do not overlap horizontally with the detected page frame are removed. Then, we consider including all possible combinations of the remaining zones into the page frame and calculate the aspect ratio of the resulting page frames. We finally select the page frame for which aspect ratio is closest to a target value. This target value can be chosen depending on the class of documents under consideration. For a typical journal article in A4 or letter size, the aspect ratio of the page frame usually lies in the interval $[1.4, 1.5]$. An example result is shown in Figure 2.

3 Error Measures

In order to determine the accuracy of the presented page frame detection algorithm, we need an error measure that reflects the performance of the evaluated algorithm. Previous approaches for marginal noise removal [6–9] use manual inspection to decide whether noise regions have been completely removed or not. While these approaches might be useful for small scale experiments, we need an automated way of evaluating border noise removal for evaluation on a large sized dataset. In the following, we introduce several performance measures to evaluate different aspects of our page frame detection algorithm.

Area overlap Let F_d be the detected page frame and F_g be the ground-truth page frame. Then the area overlap between the two regions can be defined as $A = (2|F_d \cap F_g|)/(|F_d| + |F_g|)$. However, the area overlap A does not give any hints about the errors made. Secondly, small errors like including a noise zone near the top or bottom of the page into the page frame may result in large errors in terms of area overlap.

Connected components classification The page frame partitions the connected components into two sets: the set of document components and the set of noise components. Based on this property, and defining components detected to be within the page frame as 'positive', the performance of page frame detection can be measured in terms of the four quantities 'true positive', 'false positive', 'true negative', and 'false negative'. The error rate can then be defined as the ratio of 'false' detections to the total number of connected components. This classification of connected components gives equal importance to all components, which may not be desired. For instance, if the page number is missed by the algorithm, the error rate is still very low but we loose important information about the document. Considering the page number as an independent zone, a performance measure based on detection of ground-truth zones is introduced.

Ground-truth zone detection For the zone-based performance measure, three different values are determined:

- Totally In: Ground-truth zones completely within the computed page frame
- Partially In: Ground-truth zones partially inside the computed page frame
- Totally Out: Ground-truth zones totally outside the computed page frame

Using this performance measure, we analyze the 'false negative' detections in more detail. As the page numbers are considered independent zones, losing page numbers will have a higher impact on the error rates in this measure.

OCR accuracy In order to demonstrate the usefulness of page frame detection in reducing OCR error rates by eliminating false alarms, we chose to use Omnipage 14 - a commercial OCR system. We use the edit distance [12] between the OCR output and the ground-truth text as the error measure. Edit distance is the minimum number of point mutations (insertion, deletions, and substitutions) required to convert a given string into a target string. The ground-truth text provided with the UW-III dataset has several limitations when used to evaluate an OCR system. First, there is no text given for tables. Secondly, the formating of the documents is coded as latex commands. When an OCR system is tested on this ground-truth using error measures like the Edit distance, the error rate is unjustly too high. Also, our emphasis in this work is on the improvement of OCR errors by using page frame detection, and not on the actual errors made by the OCR system. Hence, we first cleaned the UW-III documents using the ground-truth page frame, an then used the output of Omnipage on the cleaned images as the ground-truth text. This type of ground-truth gives us an upper limit of the performance of a page frame detection algorithm, and if the algorithm works perfectly, it should give 0% error rate, independent of the actual error rate of the OCR engine itself.

4 Experiments and Results

The evaluation of the page frame detection algorithm was done on the University of Washington III (UW-III) database [13]. The database consists of 1600 skew corrected English document images with manually edited ground-truth of entity bounding boxes. These bounding boxes enclose page frame, text and non-text zones, textlines, and words. The database contains a large number of scanned photocopies presenting copy borders and parts of text from the adjacent page, making it suitable for the evaluation of our page frame detection algorithm. The documents in the UW-III database can be classified into different categories based on their degradation type (see [13] for more details):

- Direct Scans (Scans): the original document has been scanned directly.
- First Generation Photocopies (1Gen): the original has been copied and this copy has been scanned.
- Nth Generation Photocopies (NGen): the Nth generation photocopy of the original has been scanned ($N > 1$).

The dataset was divided into 160 training images and 1440 test images. In order to make the results replicable, we included every 10th image (in alphabetical order) from the dataset into the training set. Hence our training set consists of images A00A, A00K, . . . , W1UA. The evaluation was done on the remaining 1440 images. Some examples of page frame detection for documents from the collection are shown in Figure 3. The rightmost image in Figure 3 shows an example where the marginal noise overlaps with some textlines at the bottom of the page. The use of page frame detection successfully detects the page contents region and removes the border noise while keeping the page contents intact.

When the page frame detection was evaluated on the basis of overlapping area, the overall mean overlap was 91%. However, the UW3 ground-truth page frame has a white border of unspecified size around the rectangle covering all

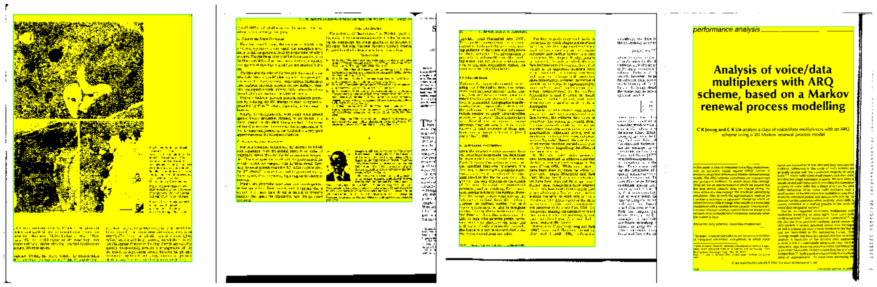

performance analysis
Analysis of voice/data multiplexers with ARQ scheme, based on a Markov renewal process modelling

Fig. 3. Some example images showing the detected page frame in yellow color.

Table 1. Results for the connected component based evaluation. The number in brackets gives the number of documents of that class. Error rates in [%]

Document Type		True Positive	False Negative	True Negative	False Positive
Scans	(392)	99.84	0.16	76.6	23.4
1Gen	(1029)	99.78	0.22	74.0	26.0
NGen	(19)	99.93	0.07	42.8	57.2
all	(1440)	99.80	0.20	73.5	26.5
total	(absolute)	4,399,718	8,753	187,446	67,605

page zones. In order to eliminate this border, we modified the ground-truth page frame by determining the smallest rectangle containing all of the ground-truth zones as page frame. Testing with these page frames as ground-truth gave an overall mean area overlap of 96%. In the following, when mentioning the ground-truth page frame, we refer to this corrected ground-truth page frame.

The results for the connected component based metric are given in Table 1. The high percentage of true positives shows that mostly, the page frame includes all the ground-truth components. The percentage of true negatives is about 73.5%, which means that a large part of noise components are removed. The total error rate defined as the ratio of 'false' detections to the total number of connected components is 1.6%. The results for the zone based metric are given in Table 2. Compared to the number of missed connected components, one can see that the percentage of missed zones is slightly higher than the corresponding percentage of false negatives on the connected component level. One conclusion that can be drawn from this observation is that the zones missed do not contain a lot of components, which is typically true for page numbers, headers, and footers of documents. In some cases, the textline finding merges the textlines consisting of textual noise to those in the page frame. In such cases, a large portion of the textual noise is also included in the page frame.

The use of page frame detection in an OCR system showed significant improvement in the OCR results. First, we ran the OCR on the original images and computed the Edit distance to the estimated ground-truth text (cf. Sec. 3). Then, we used the computed page frame to remove marginal noise from the

Table 2. Results for the zone based evaluation. Error rates in [%].

Document Type		Totally In	Partially In	Totally Out
Scans	(392)	97.6	0.7	1.7
1Gen	(1029)	97.1	1.0	1.9
NGen	(19)	97.5	0.0	2.5
all	(1440)	97.2	0.9	1.9

Table 3. Results for the OCR based evaluation with page frame detection (PFD) and without page frame detection.

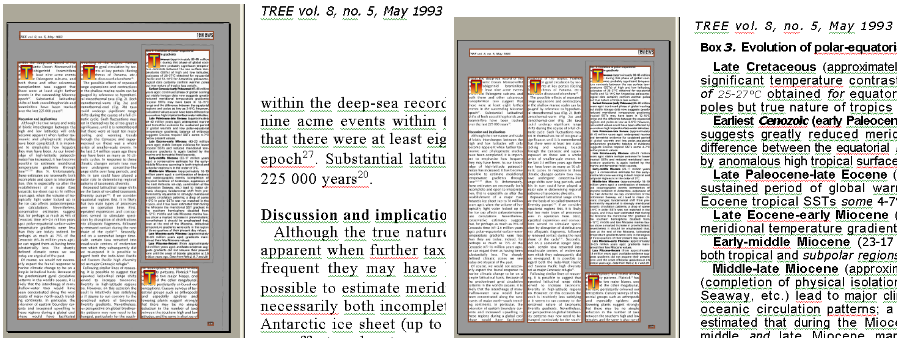
	Total Characters	Deletions	Substitutions	Insertions	Total Errors	Error Rate
Without PFD	4831618	34966	29756	140700	205422	4.3%
With PFD	4831618	19544	9828	53610	82982	1.7%

documents, and re-ran the experiments. The results (Table 3) show that the use of page frame detection for marginal noise removal reduced the error rate from 4.3% to 1.7%. The insertion errors are reduced by a factor of 2.6, which is a clear indication that the page frame detection helped in removing a lot of extraneous symbols that were treated previously as part of the document text. There are also some deletion errors, which are the result of changes in the OCR software's reading order determination. One example is shown in Figure 4, for which the reading order changed after document cleaning.

5 Conclusion

We presented an approach for page frame detection using geometric matching methods. Our approach does not assume the existence of whitespace between marginal noise and the page frame and can detect the page frame even if the noise overlaps some regions of the page content area. We defined several error measures based on area overlap, connected component classification, and ground-truth zone detection accuracy. It was shown that the algorithm performs well on all three performance measures with error rates below 4% in each case. The major source of errors was missing isolated page numbers. Locating the page numbers as a separate process and including them in the detected page frame may further decrease the error rates. The benefits of the page frame detection in practical applications were highlighted by using it with an OCR system, where we showed that the OCR error rates were significantly reduced.

Acknowledgments This work was partially funded by the BMBF (German Federal Ministry of Education and Research), project IPeT (01 IW D03).



TREE vol. 8, no. 5, May 1993

within the deep-sea recorc
nine acme events within t
that there were at least eig
epoch[27]. Substantial latitu
225 000 years[20].

Discussion and implicatio
Although the true nature
apparent when further tax
frequent they may have
possible to estimate merid
necessarily both incomplet
Antarctic ice sheet (up to

TREE vol. 8, no. 5, May 1993

Box 3. Evolution of polar-equatori
Late Cretaceous (approximatel
significant temperature contrast
of 25-27°C obtained for equator
poles but true nature of tropics
Earliest *Cenozoic* (early Paleocen
suggests greatly reduced meric
difference between the equatorial
by anomalous high tropical surface
Late Paleocene-late Eocene (
sustained period of global war
Eocene tropical SSTs *some* 4-7
Late Eocene-early Miocene (
meridional temperature gradient
Early-middle Miocene (23-17
both tropical and *subpolar regions*
Middle-late Miocene (approxin
(completion of physical isolatior
Seaway, etc.) lead to major cli
oceanic circulation patterns; a
estimated that during the Mioc

Fig. 4. Screenshot of Omnipage 14 showing the recognized text of the original document (left) and the document cleaned using page frame detection (right). Note that the reading order of the text has changed, probably due to the slightly changed geometry.

References

1. Shafait, F., Keysers, D., Breuel, T.M.: Pixel-accurate representation and evaluation of page segmentation in document images. In: 18th Int. Conf. on Pattern Recognition, Hong Kong, China (Aug. 2006) 872–875
2. Baird, H.S.: Background structure in document images. In Bunke, H.e.a., ed.: Document Image Analysis, World Scientific, Singapore (1994) 17–34
3. Breuel, T.M.: Two geometric algorithms for layout analysis. In: Document Analysis Systems, Princeton, NY (Aug. 2002) 188–199
4. O Gorman, L.: The document spectrum for page layout analysis. IEEE Trans. on Pattern Analysis and Machine Intelligence **15**(11) (Nov. 1993) 1162–1173
5. Kise, K., Sato, A., Iwata, M.: Segmentation of page images using the area Voronoi diagram. Computer Vision and Image Understanding **70**(3) (June 1998) 370–382
6. Le, D.X., Thoma, G.R., Wechsler, H.: Automated borders detection and adaptive segmentation for binary document images. In: 13th Int. Conf. Patt. Recog., Vienna, Austria (Aug. 1996) 737–741
7. Avila, B.T., Lins, R.D.: Efficient removal of noisy borders from monochromatic documents. In: Int. Conf. on Image Analysis and Recognition, Porto, Portugal (Sep. 2004) 249–256
8. Fan, K.C., Wang, Y.K., Lay, T.R.: Marginal noise removal of document images. Patt. Recog. **35** (2002) 2593–2611
9. Cinque, L., Levialdi, S., Lombardi, L., Tanimoto, S.: Segmentation of page images having artifacts of photocopying and scanning. Patt. Recog. **35** (2002) 1167–1177
10. Shafait, F., Keysers, D., Breuel, T.M.: Performance comparison of six algorithms for page segmentation. In: 7th IAPR Workshop on Document Analysis Systems, Nelson, New Zealand (Feb. 2006) 368–379
11. Breuel, T.M.: A practical, globally optimal algorithm for geometric matching under uncertainty. Electr. Notes Theor. Comput. Sci. **46** (2001) 1–15
12. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions and reversals. Soviet Physics Doklady **10**(8) (1966) 707–710
13. Phillips, I.T.: User's reference manual for the UW english/technical document image database III. Technical report, Seattle University, Washington (1996)