IJDAR (2008) 11:81–96
DOI 10.1007/s10032-008-0071-7

ORIGINAL PAPER

Document cleanup using page frame detection

Faisal Shafait · Joost van Beusekom · Daniel Keysers · Thomas M. Breuel

Received: 12 March 2008 / Revised: 1 August 2008 / Accepted: 25 August 2008 / Published online: 30 September 2008
© Springer-Verlag 2008

Abstract 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, …) appear along the border of the document. Existing document analysis methods can handle non-textual noise reasonably well, whereas textual noise still presents a major issue for document analysis systems. Textual noise may result in undesired text in optical character recognition (OCR) output that needs to be removed afterwards. Existing document cleanup methods try to explicitly detect and remove marginal noise. This paper presents a new perspective for document image cleanup by detecting the page frame of the document. The goal of page frame detection is to find the actual page contents area, ignoring marginal noise along the page border. We use a geometric matching algorithm to find the optimal page frame of structured documents (journal articles, books, magazines) by exploiting their text alignment property. We 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. Further tests were run on a dataset of magazine pages and on a set of camera captured document images. To demonstrate the benefits of using page frame detection in practical applications, we choose OCR and layout-based document image retrieval as sample applications. Experiments using a commercial OCR system show that by removing characters outside the computed page frame, the OCR error rate is reduced from 4.3 to 1.7% on the UW-III dataset. The use of page frame detection in layout-based document image retrieval application decreases the retrieval error rates by 30%.

Keywords Document analysis · Marginal noise removal · Document pre-processing

F. Shafait · D. Keysers
Image Understanding and Pattern Recognition (IUPR) Research Group, German Research Center for Artificial Intelligence (DFKI), 67663 Kaiserslautern, Germany
e-mail: faisal@iupr.dfki.de

J. van Beusekom (✉) · T. M. Breuel
Department of Computer Science, Technical University of Kaiserslautern, 67663 Kaiserslautern, Germany
e-mail: joost@iupr.net

D. Keysers
e-mail: keysers@iupr.dfki.de

T. M. Breuel
e-mail: tmb@informatik.uni-kl.de

1 Introduction

Paper positioning variations is a class of document degradations that results in skew and translation of the page contents in the scanned image. Document skew detection and correction has received a lot of attention in last decades and several skew estimation techniques have been proposed in the literature (for a literature survey, please refer to [1]). However, estimating the global position of the page has been largely ignored by the document analysis community. This is perhaps due to the fact that most of the layout analysis methods are robust to global translation of the page and would produce the same segmentation of the page for different translations as long as all page contents are visible. Hence the OCR output is usually not affected by global translation of the page. This effect can be seen in Fig. 1, where a page segmentation algorithm is shown to correctly identify the page segments irrespective of the translation of the page in each image.

Different amount of noise can be present along the border of a document image depending on the position of the paper on the scanner. Figure 1 shows the effect of paper positioning

Fig. 1 Example images showing the results of a page segmentation algorithm on pages with different amounts of global translation. The results show that the algorithm identifies the page blocks quite well in each case irrespective of the translation in the page

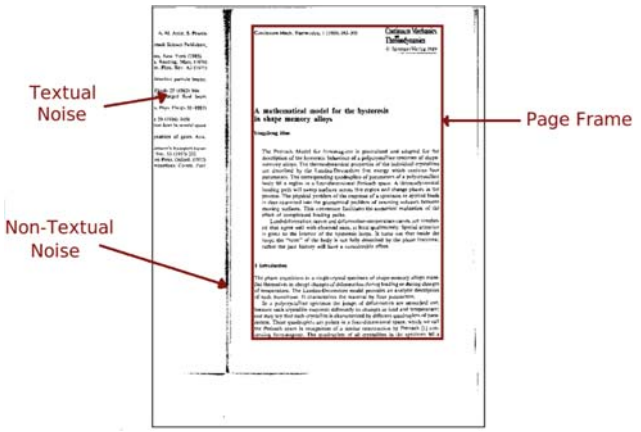


Fig. 2 Example image showing textual and non-textual noise along the page border

variations on the amount of marginal noise in the resulting scanned image. In general, marginal noise along the page border can be classified into two broad categories based on its source:

- non-textual noise (black bars, speckles, …) resulting from the binarization process
- textual noise coming from the neighboring page

An example image showing textual and non-textual noise along the page border is shown in Fig. 2.

The most common approach to deal with non-textual noise is to perform document cleaning by filtering out connected components based on their size and aspect ratio [2–4]. This usually works out quite well in removing black bars and isolated specks. However, when characters from the adjacent page are also present, they cannot be filtered out using this approach. Therefore, state-of-the-art page segmentation algorithms report a number of false alarms originating from textual noise regions [5]. When these textual noise regions are fed to a character recognition engine, extra characters appear in the output of the OCR system along with the actual contents of the document. Hence the edit distance between the OCR output and the ground-truth text increases resulting in decreased OCR accuracy.

Textual noise can be avoided altogether by scanning only the page contents area. Typical desktop scanners come with a graphical user interface to allow the users to conveniently mark the region to be scanned. This allows the user to manually select the page frame during document scanning. The resulting document image is then free of textual noise. However, if a large number of documents have to be scanned, manually defining the page frame for each one of them becomes quite cumbersome.

Researchers have also tried to explicitly detect and remove marginal noise in scanned documents. For example, Le et al. [6] have proposed a rule-based algorithm using several heuristics to detect the page borders. The algorithm relies upon the classification of document rows and columns into blank, textual or non-textual classes. Then, an analysis of projection profiles and crossing counts is done to detect the marginal noise. Their approach is based on the assumption that the marginal noise is very close to the edges of the image and borders are separated from image contents by a large whitespace, 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

(see Fig. 6). Avila et al. [7] and Fan et al. [8] propose techniques for removing non-textual noise overlapping the page content area, but do not consider textual noise removal. Cinque et al. [9] propose an algorithm for removing both textual and non-textual noise from grayscale images based on image statistics like horizontal/vertical difference vectors and row luminosities. However, their method is not suitable for cleaning binary images. Also, their approach is very sensitive to the amount of noise present in the document image and the error rates increases monotonically with the artifact area.

Other more recent attempts for border noise removal were by Peerawit et al. [10] and Stamatopoulos et al. [11]. Both these approaches try to remove textual and non-textual noise from document images. The approach in [10] tries to identify borders of noise regions based on an analysis of the projection profiles of the edges in the image. Their technique is based on the observation that non-textual marginal noise areas have much higher density of edges than normal text. Again, this observation may not hold for all documents (see Fig. 9). The approach in [11] tries to detect page borders based on an analysis of the projection profiles of the smeared image combined with a connected component labeling process. They have demonstrated their technique on flat camera captured documents. A common feature of these techniques is that they try to design some rules to detect noisy regions along the page border. However, in practice such rules tend to work only on a small collection of documents or on documents captured under similar scanning conditions. None of the above mentioned approaches has been tested on large publicly available datasets. So it is hard to judge their performance under real-world circumstances.

This paper presents a new approach for dealing with paper positioning variations in scanned documents. Instead of identifying and removing noisy components themselves, the proposed method focuses on identifying the actual content area. This is accomplished by using a geometric matching algorithm. Including page frame detection as a document pre-processing step can help to increase OCR accuracy by removing textual noise from the document. Also in applications like document image retrieval based on layout information [12], noise regions result in incorrect matches. Using the page frame to reject zones originating from noise can therefore reduce the retrieval error rates.

Our method for page frame detection takes advantage of the structure in a printed document to locate its page frame. This is done in two steps. First, a geometric model is built for the page frame of a scanned document. Then, a geometric matching method is used to find the globally optimal page frame with respect to a defined quality function.

The use of geometric matching for page frame detection has several advantages. Instead of devising carefully crafted rules, the page frame detection problem is solved in a more general framework, thus allowing higher performance on a more diverse collection of documents. Additionally, the use of geometric model for page frame detection makes the presented approach very robust to the amount of noise present in a document image and can find the page frame even if noise overlaps some regions of the page content area.

Part of the work presented in this paper was published in [13] for timely dissemination of this work. This paper is a substantially extended version of the previous conference publication.

The rest of this paper is organized as follows. Section 2 describes in detail the method for page frame detection. In Sect. 3, several error measures to evaluate the performance of a page frame detection algorithm are proposed. Section 4 presents the experimental protocol and discusses the results obtained, followed by the conclusion in Sect. 5.

2 Geometric matching for page frame detection

2.1 Document model

For structured documents, like technical journals and business letters, the structure can be described as a hierarchy, where entities at each level of the hierarchy represent a particular level of information, like zones, text-lines, or connected components. Different hierarchical models representing document structure have been proposed in the past [14–16]. One common shortcoming of these models is that they only represent the contents of the document and do not specify how to represent textual and/or non-textual noise added to the document by the photocopying or scanning process. For instance, in the hierarchical model of Liang et al. [15], all polygonal regions in the page are organized in a hierarchy of zones, text blocks, text-lines, etc. and a reading order is defined between them. However, in the presence of regions consisting of noise, it is not clear how the reading order should be defined among the zones in the page. To address this problem, we extend the definition of the hierarchical document model in this work and another add another level of hierarchy that represents the actual page content area. In this way it is then possible to define a unique reading order of zones within the page content area, ignoring textual and non-textual noise along the page border. The definitions of different levels of the hierarchy are as follows.

- A binary *document image* D is defined as the union of the set of the foreground pixels P_f and the background pixels P_b.
- The set of foreground pixels can then be partitioned into *connected components* $C = \{C_1, \ldots, C_M\}$ such that $C_i \cap C_j = \emptyset \ \forall i \neq j$ and $\bigcup_{i=1}^{M} C_i = P_f$.

- The set of *text-lines* $L = \{L_1, \ldots, L_N\}$ is viewed as a partitioning of the connected components such that $L_i \subseteq C$, $L_i \cap L_j = \emptyset \ \forall i \neq j$ (some connected components may not be included in any text-line).
- The set of *zones* $Z = \{Z_1, \ldots, Z_R\}$ is defined such that each zone $Z_i \subseteq C$ and $Z_i \cap Z_j = \emptyset \ \forall i \neq j$, where each zone consists of only one physical layout structure like text, graphics, or pictures.
- The *page frame* F is defined as the minimum rectangle containing all connected components belonging to the actual document.

Note that other levels of the hierarchy are also possible (e.g. word-level, character-level), but the above-mentioned levels are sufficient to describe a document for the purpose of page frame detection.

In order to extract the document structure at different levels of the hierarchy, the page frame detection system uses a different algorithm at each level. A fast labeling algorithm is used to extract connected components from the document image. The constrained text-line finding algorithm [3] is used to extract text-lines, whereas the Voronoi-diagram based algorithm [17] is used to extract zones from the document. These algorithms were chosen since recent evaluations of page segmentation algorithms [5,18] show that they work well on standard document collections like UW-III. The text-line extraction algorithm was used with a high threshold for the quality of the extracted text-lines to avoid text-lines generated from non-textual noise components. We used the implementations of connected component analysis and text-line extraction algorithms from the OCRopus open source OCR system [19], and the implementation of the Voronoi algorithm from the PSET toolkit [20].

2.2 Page frame model

The page frame of a scanned document is parameterized as a rectangle described by five parameters $\vartheta = \{l, t, r, b, \alpha\}$. The parameters $\{l, t, r, b\}$ represent the left, top, right, and bottom coordinates, respectively, whereas α represents the skew angle of the page frame. The page frame detection system takes skew corrected documents as input; standard skew correction methods [21,22] can be used for this purpose. Hence, the page frame is modeled as an axis-aligned rectangle, described by four parameters $\vartheta = \{l, t, r, b\}$. Given the sets of connected components C, text-lines L, and zones Z, the goal of page frame detection is to find the maximizing set of parameters ϑ with respect to the sets C, L, and Z:

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

where $Q(\vartheta, C, L, Z)$ is the total quality for a given parameter set, and T is the parameter space. The design of the quality function is described in detail in Sect. 2.3, followed by the description of the algorithm for finding the optimal set of parameters in Sect. 2.4.

2.3 Design of quality function

The design of the quality function in Eq. (1) is done by exploiting the text-alignment property of structured documents. In such documents, text-lines are usually printed in justified or left-aligned style. Hence, a large number of connected components are aligned with the page frame of the document. At first glance, it may seem like a good idea to use the number of character bounding boxes touching the page frame as the quality of the page frame. The character bounding boxes could be obtained from C by filtering out noise and non-text components based on their area and aspect ratio. However, such an approach does not work well in practice because:

1. The top and bottom text-lines do not necessarily contain more characters than other text-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, logos, …) at the top or bottom of the page. Hence the parameters t and b cannot be reliably estimated using character level information.
2. The parameters l and r can only be reliably estimated for justified text.

Therefore, instead of using connected component level information, text-lines can be used. The quality function can then be a function of the number of text-lines that touch the page frame from inside. Based on this idea, the parameters ϑ can be decomposed 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 done because of the nature of the problem. First, the parameters ϑ_v are set to their extreme values ($t = 0, b = H$ where H is the page height) and then optimal ϑ_h is searched. This setting ensures that none of the candidate text-lines is lost based on its 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 will be seen later in Sect. 2.5. Hence the optimization problem of Eq. (1) is reduced to

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

The total upper bound of the quality Q can be written as the sum of local quality functions

$$Q(\vartheta_h, L) := \sum_{j=1}^{N} q(\vartheta_h, L_j) \tag{3}$$

An upper and lower bound for local quality function q is computed. Given a line bounding box $\bar{L} = \{x_0, y_0, x_1, y_1\}$, 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, are determined. The local quality function q for a given line and a parameter range ϑ_h can then be defined as

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

where ϵ defines the distance up to which a text-line can contribute to the page frame. Text-lines may have variations in their starting and ending positions within a text column depending on text alignment or paragraph indentation. A value of $\epsilon = 150$ pixels is used in this work in order to cope with such variations for documents scanned at 300-dpi. This quality function alone already works well for single column documents, but for multi-column documents it may report a single text-column (with the highest number of text-lines) as the optimal solution. In order to discourage such solutions, a negative weighting for text-lines on the 'wrong' side of the page frame (that is x_1 of a text-line contributing to parameter l or x_0 of a text-line contributing to parameter r) is introduced in the form of the quality function

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

The overall local quality function is then defined as

$$q(\vartheta_h, (x_0, x_1)) = q_1(\vartheta_h, (x_0, x_1)) + q_2(\vartheta_h, (x_0, x_1)) \quad (6)$$

The quality function in Eq. (6) will yield the optimal parameters for ϑ_h even if there are intermediate text-columns with larger number of text-lines. However, if the first or last column contains very few text-lines, the column can possibly be ignored. The search space for the parameters ϑ_h can be limited to certain regions of the document image to solve this problem. In this work the value of the parameter l was constrained to lie within the first half of the page, whereas the value of the parameter r was limited to the second half of the page.

2.4 Branch-and-bound optimization

The RAST (Recognition by Adaptive Subdivision of Transformation Space) technique [23] is employed to perform the maximization in Eq. (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. During the search, each partition of the search space is described by a Cartesian product of intervals for the parameters, i.e. a set of the form $T = [l_0, l_1] \times [r_0, r_1]$. The upper bound on the quality of the page frame with parameters in the rectangular region T is calculated using interval arithmetic [24]. Given a computation of an upper bound on the quality, the search can be organized as follows (for details see [23,25]):

1. Pick an initial region of parameter values T such that it contains all possible values of parameters that can occur in practice.
2. Maintain a priority queue of regions T_i, where the upper bound on the possible values of the global quality function Q for parameters $\vartheta \in T_i$ is used as the quality.
3. Remove a region T_i from the priority queue; if the upper bound of the quality function associated with the region is too small to be of interest, terminate the algorithm.
4. If the region is small enough to satisfy the accuracy requirements for the dimensions of a region, accept it as a solution.
5. Otherwise, split the region T_i along the dimension furthest from the accuracy constraints and insert the subregions into the queue; then continue the algorithm at Step 3. If different parameters have same accuracy requirements, the dimension furthest from the accuracy requirements is the largest dimension.

This algorithm will return the parameter set that maximizes the quality of the match function in Eq. (2). To make the approach practical and avoid duplicate computations, a match-list representation [23] is used. That is, with each region kept in the priority queue in the algorithm, a list (the match-list) of all and only those text-lines is maintained that have the possibility to contribute with a non-zero local quality to the global quality. These match-lists shrink with decreasing size of the regions T_i. It is easy to see that the upper bound of a parameter space region T_i is also an upper bound for all subsets of T_i. Hence, when a region is split in Step 5, the text-lines in the children that have already failed to contribute to the quality computation in the parent never have to be reconsidered. Thus the match-lists can be reused in the children thereby allowing a very fast computation of quality for the children.

2.5 Parameter refinement

The RAST algorithm returns the optimal parameters for ϑ_h in terms of mean square error with respect to the quality function in Eq. (3). However, if the text is not aligned in the justified style or if different paragraphs have different indentation, parameters ϑ_h returned by the RAST algorithm

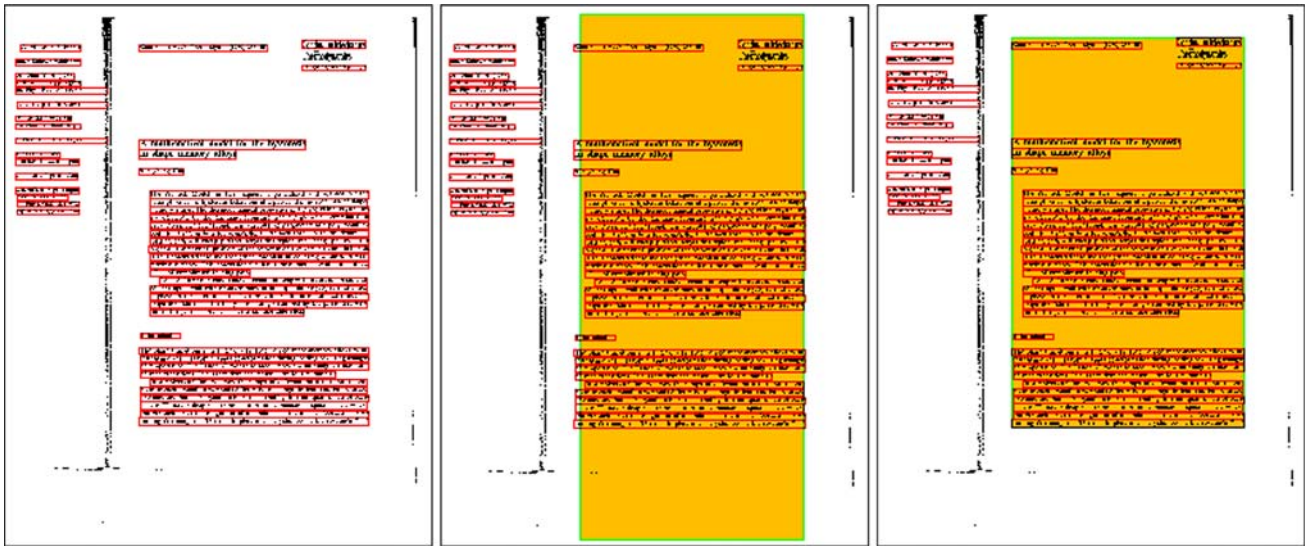


Fig. 3 Example image demonstrating parameter refinement in order to adapt them to text alignment. The detected text-lines are shown in the *leftmost image*. Note that some part of the text is indented more to the right as compared to other text on the page. The page frame corresponding to the optimal parameters with respect to Eq. (6) is shown in the *middle image*. The image on the *right* side shows the initial page frame after adjusting the parameters for text alignment

may cut through some text-lines as shown in Fig. 3. So the parameters are refined to adjust the page frame according to different text alignments. If the bounding box of a text-line overlaps with the page frame by more than half of its area, the page frame parameters ϑ_h are expanded to include the complete text-line, as shown in Fig. 3.

The use of match-lists gives the list of text-lines bounding boxes which contributed positively to the quality function $Q(\vartheta_h, L)$. All these text-lines are sorted with respect to the top of each text-line's bounding box (y_0). This gives an initial estimate for parameters ϑ_v by simply setting $t = \min(y_{0,j}), j = 1, \ldots, N$ and $b = \max(y_{1,j}), j = 1, \ldots, N$. A page frame detected in this way is shown in Fig. 3. Although the detected page frame is correct for most of the documents, it fails in these cases:

1. If there is a non-text zone (images, graphics, logo, …) 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 text-line detection, it will not be included in the detected page frame.

An example illustrating these problems is shown in Fig. 4. In order to estimate the final values for $\vartheta_v = \{t, b\}$, document zones are used as given by the Voronoi algorithm [17]. The Voronoi algorithm performs document cleaning as a part of zoning process and successfully removes most of the non-textual noise. The output of the Voronoi algorithm for an example image is shown in Fig. 4. Textual noise usually appears only along the left or the right side of the document. Based on this observation, filtering is performed 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, all of the remaining zones are included into the page frame. An example result is shown in Fig. 4.

3 Performance measures

To determine the accuracy of the presented page frame detection algorithm, performance measures are needed that not only reflect the accuracy of the algorithm, but also quantify its usefulness in practical document analysis systems. Therefore, the error measures are categorized into two parts.

3.1 Page frame detection accuracy

The goal of the performance measures in this section is to determine the accuracy with which the page frame is located. Previous approaches for marginal noise removal [6–9] use manual inspection to decide whether noise regions have been completely removed or not. Then, the error rate is defined as the percentage of documents on which the noise was not completely removed. While these approaches might be useful for small scale experiments, an automated way of evaluating border noise removal is needed for evaluation on a large sized dataset. In the following, performance measures based on area overlap, connected components classification,

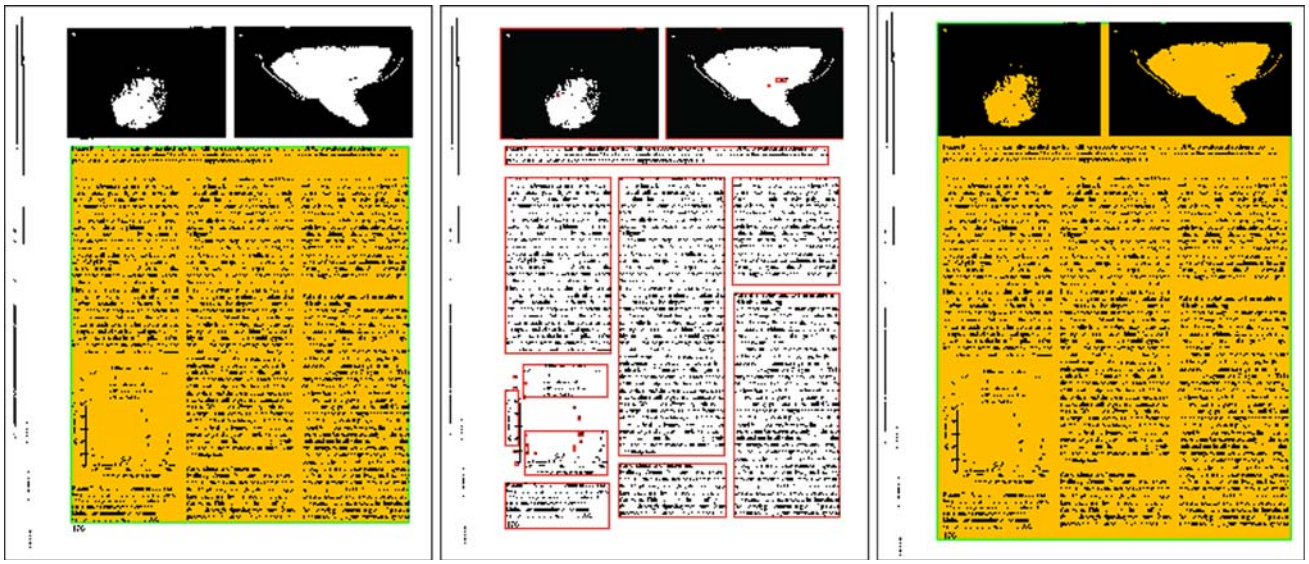


Fig. 4 Example image demonstrating inclusion of non-text zones into the page frame. The initial page frame detected based only on the text-lines is shown on *left*. Note that the detected page frame does not include the images on the *top* and the page number at the *bottom*. The *middle image* shows the zones detected by the Voronoi algorithm. The *rightmost image* shows the final page frame obtained by using zone-level information

and ground-truth zone detection are introduced to evaluate different aspects of the presented page frame detection algorithm.

3.1.1 Area overlap

Let F_g be the ground-truth page frame and F_d be the detected page frame. Then the area overlap between the two page frames can be defined as

$$A = \frac{2|F_g \cap F_d|}{|F_g| + |F_d|} \tag{7}$$

The amount of area overlap A will vary between zero and one depending on the overlap between ground-truth and detected page frames. If the two page frames do not overlap at all $A = 0$, and if the two page frames match perfectly, i.e. $|F_g \cap F_d| = |F_g| = |F_d|$, then $A = 1$. This gives a good measure of how closely the two page frames match. However, the area overlap A does not give any hints about the errors made by the algorithm. Secondly, a small error like including a noise zone near the top or bottom of the page into the page frame may result in a large error in terms of area overlap. To evaluate the page frame detection algorithm in more detail, a performance measure based on connected component classification is defined.

3.1.2 Connected components classification

Defining components detected as lying within the page frame as 'positive', the performance of page frame detection can be measured in terms of four quantities: 'true positive', 'false positive', 'true negative', and 'false negative'. The error rate can then be defined as the ratio of incorrectly classified connected components to the total number of connected components.

The error measure based on classification of connected components gives equal importance to all components, which may not be desired. For instance, if the page number is not included in the detected page frame, the error rate will still be very low because page number comprises a very small fraction (typically about 0.03–0.1%) of the total number of connected components in the page frame. However, the page number carries important information for the understanding of the document. To compensate this shortcoming, a performance measure based on detection of ground-truth zones is introduced.

3.1.3 Ground-truth zone detection

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

- Totally in: Ground-truth zones lying completely inside the computed page frame
- Partially in: Ground-truth zones lying partially inside the computed page frame
- Totally out: Ground-truth zones lying totally outside the computed page frame.

Using this performance measure, the 'false negative' detections are analyzed in more detail. Since, the page numbers are

considered an independent zone, missing page numbers will have a higher impact on the error rates in this performance measure.

3.2 Performance gain in practical applications

In order to demonstrate the usefulness of page frame detection in practical applications, we chose OCR and layout-based document image retrieval applications.

3.2.1 OCR accuracy

The OCR accuracy is determined by the percentage of characters correctly recognized in a document image. Many extra characters (false alarms) may appear in OCR output if textual noise is present in the document. Current commercial OCR systems have their own noise removal techniques to deal with marginal noise. The edit distance [26] between the OCR output and the ground-truth text is used as the error measure for determining OCR accuracy. Edit distance is the minimum number of point mutations (insertion, deletions, and substitutions) required to convert a given string into a target string. The goal of performance measure based on edit distance is to determine whether the performance of existing OCR systems improves if page frame detection is used as a pre-processing step.

3.2.2 Layout-based document image retrieval

In layout-based retrieval, the purpose is to query document image databases by layout, in particular by measuring the similarity of different layouts in comparison to a reference or query layout. Blocks originating from marginal noise result in incorrect matches, thereby increasing the error rates of the retrieval system. Different layout analysis or page segmentation algorithms use different methods to deal with noise in a document image. The goal of this performance measure is to determine the decrease in retrieval error rates when page frame detection is used as a pre-processing step.

4 Experiments and results

The evaluation of the page frame detection algorithm was done on the University of Washington III (UW-III) database [27]. The dataset was divided into 160 training and 1,440 test images. In order to make the results replicable, every tenth image (in alphabetical order) from the dataset was included into the training set. Hence the training set consists of images A00A, A00K, …, W1UA. The training images were used to design the quality function (Sect. 2.3) and to find suitable values for parameters (e.g. ϵ). The post-processing steps (Sect. 2.5) were also introduced based on results on the training images to cope with different layout styles and the presence of non-textual content at the top or bottom of a page image.

The evaluation of our page frame detection system was done on the remaining 1,440 test images. Some examples of page frame detection for documents from the UW-III dataset are shown in Fig. 5. Figure 6 shows an example where marginal noise overlaps with some text-lines at the bottom of the page. The use of page frame detection successfully detects the page contents region and removes the border noise from the image while keeping the page contents intact.

4.1 Page frame detection accuracy

The evaluation of page frame detection on the basis of overlapping area (Eq. 7) showed a page frame detection accuracy of 91%. An inspection of the UW3 ground-truth page frame showed that it does not tightly enclose the page contents area as shown in Fig. 7. Hence, the correct page frame of documents in the test set was computed by finding the bounding box of all ground-truth zones for each document. Testing

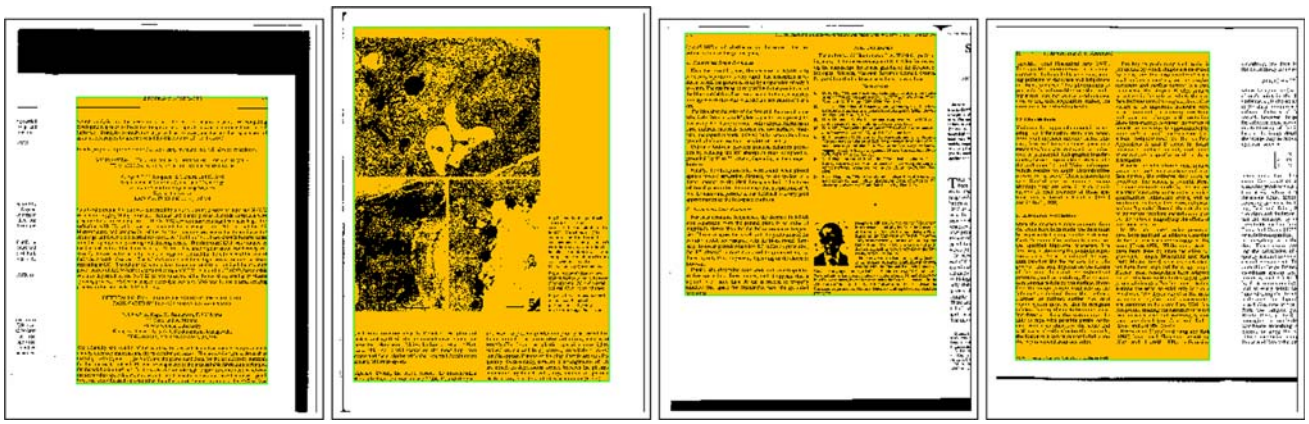


Fig. 5 Some example images showing the detected page frame in *yellow color*

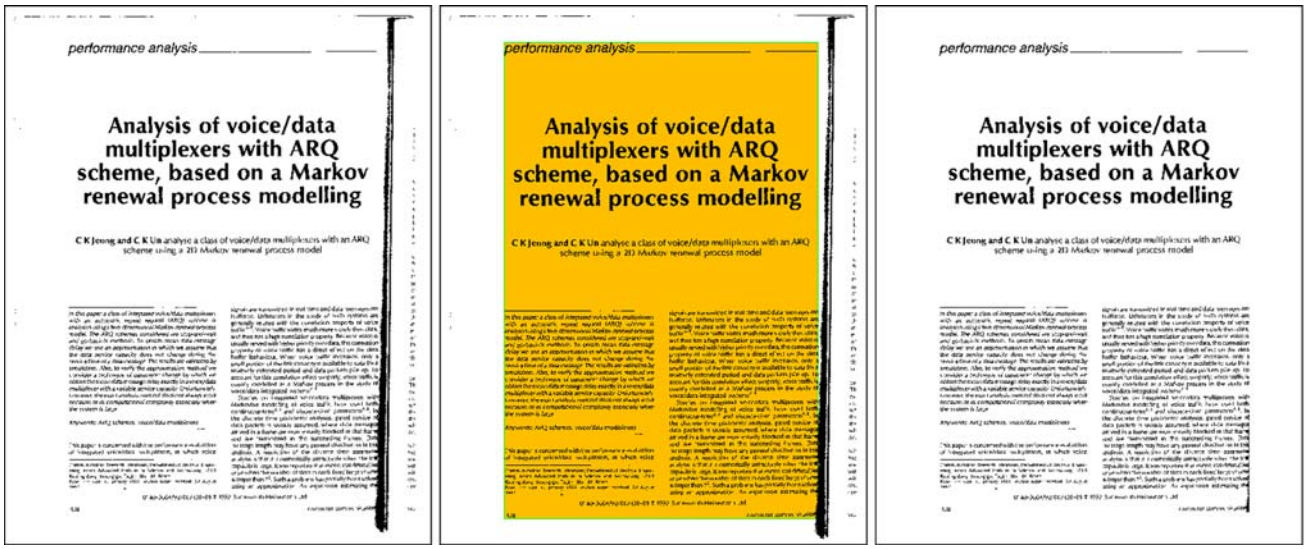


Fig. 6 An example image (A005) showing the page frame detection in case of border noise overlapping the page content area. Image on the *left* shows the original document, the *middle* image shows the detected page frame, and the *right* image shows the cleaned image removing both textual and non-textual noise outside the page content area while keeping the page content area intact

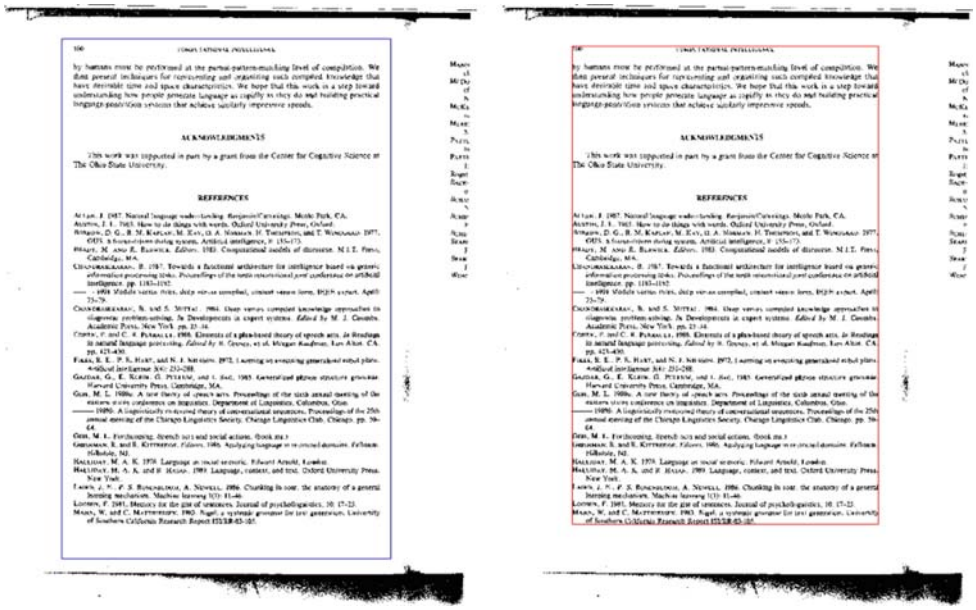


Fig. 7 The *left* image shows a document together with its original ground-truth page frame. The *right* image shows the corrected ground-truth page frame obtained by computing the smallest rectangle including all the ground-truth zones

with the corrected ground-truth page frame gave an overall mean area overlap of 96%. In the following, when mentioning the ground-truth page frame, this corrected ground-truth page frame is meant.

The result for the connected component based measure is given in Table 1. The high percentage of true positives shows that the page frame mostly 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 successfully removed. The results for the *N*th generation photocopies show that the percentage of true negatives goes down to 42.8% which may lead to the conclusion that the computed page frames for this subset are typically bigger than the ground-truth page frame. The total error rate defined as the ratio of 'false' classifications to the total number of connected components is 1.6%. Since the test set contains only 19 images in the *N*Gen category, the total results do not reflect the performance on such severely degraded documents. A detail study of the performance of the proposed page frame detection method on documents with different noise levels is presented later in this section.

The results for the zone based measure are given in Table 2. Compared to the number of missed connected components, it can be seen that the percentage of missed zones is slightly

Table 1 Results for the connected component based evaluation

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
*N*Gen (19)	99.93	0.07	42.8	57.2
All (1440)	99.8	0.2	73.5	26.5
Total (abs.)	4,399,718	8,753	187,446	67,605

The number in brackets gives the number of documents of that class. Error rates in (%)

Table 2 Results for the zone based performance evaluation

Document type	Totally in	Partially in	Totally out
Scans (392)	97.6	0.7	1.7
1Gen (1029)	97.1	1.0	1.9
*N*Gen (19)	97.5	0.0	2.5
All (1440)	97.2	0.9	1.9

Error rates in (%)

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 large number of components, which is typically true for page numbers, headers and footers of documents. These zones have a few components and therefore do not contribute much to the mean false negative errors on the connected component level. In some cases, the text-line finding algorithm merges the text-lines consisting of textual noise to those in the page frame. In such cases, a large portion of textual noise is also included in the page frame.

A box plot of the run times of different steps of the proposed method is shown in Fig. 8. The execution times were computed on an AMD Athlon 1.8 GHz machine running Linux. The worst case running time of the unmodified RAST algorithm is exponential in the problem size [28]. However, in practice such a case rarely appears. For page frame detection, the RAST algorithm took less than 10 ms. per page on the average. Hence if it is integrated with a document analysis system that already computes text lines and zones, page frame detection will not add a significant increase in the computation time. When page frame detection is used as a monolithic system for document image cleanup, the total running time is of interest, which is 3–4 s per page.

A particular advantage of our method is that it is robust to skew between the main page frame and the textual noise. An example image is shown in Fig. 9 where our method successfully finds the main page content area despite the presence of textual noise with a different skew angle. Also note that there is very little non-textual noise in this page. Therefore the assumption by Peerawit et al. [10] that noisy regions can

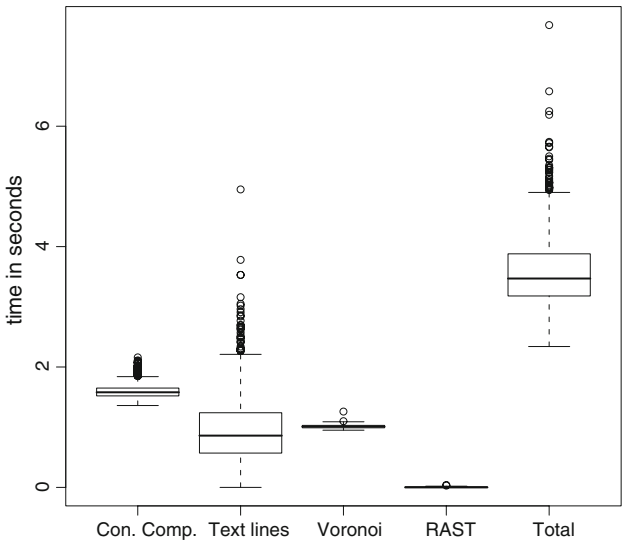


Fig. 8 Run times of the different steps of the algorithm for the test on UW-III

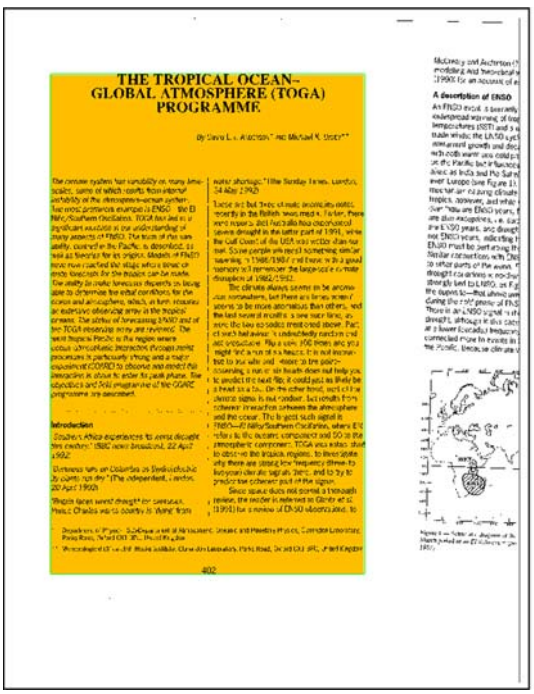


Fig. 9 An example image (S021) showing the page frame detection in case of skew between the main page frame and the textual noise

be separated from text regions based on edge density, will not work here.

In order to quantify the amount of marginal noise in a document image, the noise ratio of a document image is defined as

$$\text{Noise ratio} = \frac{n_{pb}}{n_p} \tag{8}$$

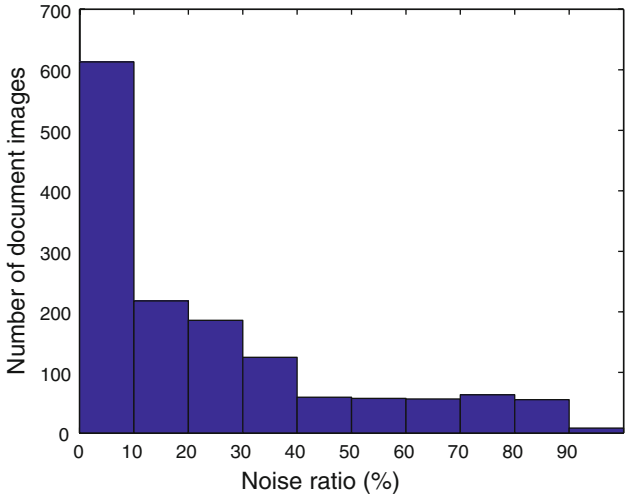


Fig. 10 Histogram of the noise ratio (Eq. 8) of the documents in the test set

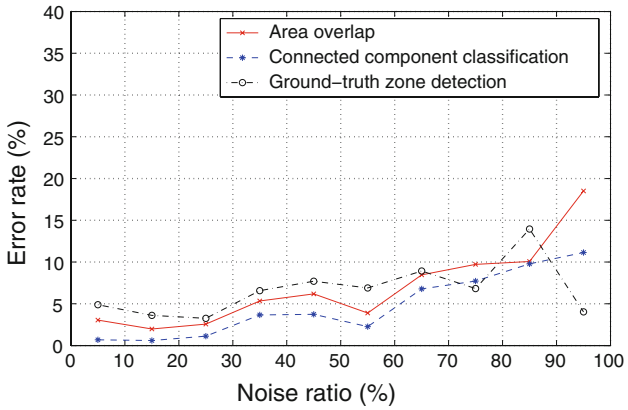


Fig. 11 Performance of the page frame detection on different documents categorized by their noise level. The *three lines* show the three different error measures introduced in Sect. 3.1.1, 3.1.2 and 3.1.3 with increasing noise level

where n_{pb} is the number of foreground pixels outside the ground-truth page frame, and n_p is the total number of foreground pixels in a document image. A histogram of the noise level of the documents in the test set is shown in Fig. 10. Interestingly, there are many documents with noise levels above 50%. The mean error rate obtained for each of these noise level based document categories is plotted in Fig. 11. The plot shows that the algorithm works well even on documents with very high amount of noise. The error rates on all three performance measures used are below 10% for noise levels up to 80%.

Some limitations of the presented page frame detection algorithm were also revealed during the course of evaluation. Although the algorithm works very well for most of the layouts even under large amount of noise, yet for a few layouts the algorithm does not give 100% result even for noise-free documents. This happens for documents with very few text-lines beside the margin of the document and there is no text-line that spans across the main content area and the page margin. In this case, these text-lines lie completely outside the computed parameters ϑ_h (Eq. 2). So the parameter refinement step (Sect. 2.5) fails to include these text-lines into the page frame. To deal with such layouts, the quality function can be modified to include an offset between the page frame parameters and the main content area of the page.

4.2 Performance gain in practical applications

The use of page frame detection in an OCR system showed significant improvement in the OCR results. For this purpose Omnipage 14—a commercial OCR system—was chosen. 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 formatting 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, the UW-III documents are first cleaned using the ground-truth page frame, and then the output of Omnipage on the cleaned images was used 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.

First, OCR was performed on the original images and the Edit distance to the estimated ground-truth text was computed. Then, the computed page frame was used to remove marginal noise from the documents, and the experiments was run again. The results (Table 3) show that the use of page frame detection for marginal noise removal reduced the OCR 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 extra text that were treated previously as part of the document text. There are also some deletion errors, which are a result of the changes in the OCR software's reading order determination.

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

	Del.	Subst.	Ins.	Total errors	Error rate (%)
W/O PFD	34966	29756	140700	205422	4.3
With PFD	19544	9828	53610	82982	1.7

The total number of characters is about 4.8 million

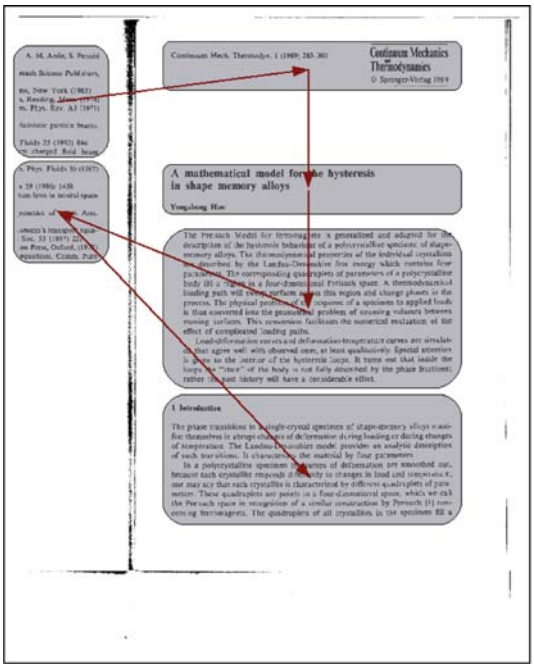


Fig. 12 Result of Omnipage 14 showing the recognized text blocks in the document and their reading order. Note that Omnipage fails to find a correct reading order and inserts the textual noise zone between the text zones from the page contents area

One example is shown in Fig. 12 for which the reading order changed after document cleaning.

The effect of using page frame detection on the performance of a layout based document image retrieval application showed a significant decrease in retrieval error rates. van Beusekom et al. [12] introduced several similarity measures for layout based document image retrieval. They evaluated the performance of these similarity measures on the MARG database. The experiments showed that the best distance measure for this task is the overlapping area combined with the Manhattan distance of the corner points as block distance together with the minimum weight edge cover matching for establishing correspondences between the matched layouts. The documents in the MARG database are categorized with respect to 9 different layout types, 59 publishers, and 161 journals. Given a query document, the target is to retrieve a document of the same class based on layout information only. The error rates are then determined as the percentage of correctly retrieved documents using leave-one-out cross validation.

In this work retrieval experiments were performed both with and without using page frame detection. Since each method for page segmentation has a different way of dealing with noise, four well-known page segmentation algorithms were compared for use in layout based retrieval: the X–Y cut [29], Docstrum [4], whitespace analysis [2], and the Voronoi-diagram based approach [17].

In the first experiment, the document images were used directly for page segmentation without any page frame detection. The blocks extracted from the documents were then used for the purpose of layout based retrieval.

In the second experiment, the document images in the database were cleaned by performing page frame detection and removing all the foreground pixels outside the detected page frame. Following the document cleaning, page segments were extracted from the cleaned images and then retrieval experiment was repeated. The decrease in error rates for each of the three subdivisions of the data set (according to type, publisher, and journal) was used as a performance measure.

The use of page frame for layout-based document image retrieval resulted in lower error rates on all three classes of layouts for each algorithm as shown in Table 4. These results show that the Voronoi-diagram based approach performs better than other algorithms both with and without page frame detection. The use of page frame detection with the Voronoi algorithm lowers the retrieval error rates by 4% for the correct journal, 30% for the correct type, and 20% for the correct publisher. These results clearly demonstrate the usefulness of page frame detection in practical applications.

4.3 Experiments on magazine pages

The UW-III dataset contains scanned pages from journal articles with Manhattan layouts. In order to test the performance of our approach on non-Manhattan layouts that are more representative of layouts in magazines, we chose the ICDAR 2007 page segmentation contest dataset [30]. The dataset contains 23 magazine pages, among which 6 are in the training set and 17 are in the test set. We used all 23 images as test images and used our method without any parameter tuning on these images. Some of the results are shown in Fig. 13. It can be seen that the page frame detection works correctly for most of these pages. In Fig. 13a–c our algorithm

Table 4 Comparison of the error rates (%) for layout-based document image retrieval with and without using page frame detection

Segmentation algorithm	Page frame detection	MARG database classes		
		Journal	Type	Publisher
Voronoi	No	31.0	7.5	7.0
	Yes	29.7	5.3	5.4
X–Y cut	No	36.3	11.7	13.6
	Yes	33.5	8.6	8.0
Docstrum	No	40.9	14.0	14.4
	Yes	32.1	7.4	7.1
Whitespace	No	48.3	20.3	24.6
	Yes	31.2	7.2	6.1

(a) mp00004 (b) mp00248 (c) mp00262

(d) mp00307 (e) mp00253 (f) mp00290

Fig. 13 Example of page frame detection on magazine pages from the ICDAR 2007 page segmentation contest. **a** Image with non-Manhattan layout with left aligned text. **b** Image with large variety of font sizes and a vertical text-line. **c** Image with text-lines extending beyond other lines on the page with some vertical text-lines. **d** Image with a mixture of content type. **e** Image with margin notes. **f** Image with a column containing no text-lines

found the correct page frame without making any error. In Fig. 13d one vertical text-line was missed by our algorithm since it was not detected by the text-line detection algorithm. Another error is shown in Fig. 13e where the last column consisted only of a few lines and hence was excluded from the detected page frame. Note that in Fig. 13d, margin notes were included in the computed page frame because the title lines were spanning across the main text content area and the margin notes area. A similar error is shown in Fig. 13f where the last column consists only of images and hence was not included in the page frame.

To evaluate the performance of our algorithm quantitatively, we used the ground-truth zone detection measure (see Sect. 3.1.3). Since ground-truth information was not available, we counted the number of text-lines in each document and calculated the percentage of totally in, partially in, and totally out text-lines. The results are shown in Table 5. The results show that the page frame detection correctly included

Table 5 Results for the text-line based error measure on magazine pages from the ICDAR 2007 page segmentation competition [30]

Type of line	Totally in	Partially in	Totally out
Horizontal (2393)	98.6	0.0	1.4
Vertical (15)	66.7	0.0	33.3
Inverted (22)	81.8	13.6	4.6
All (2430)	98.23	0.12	1.65

Error rates in (%)

98.23% text-lines from the test images. The partially missed lines were those with inverted text and were on the top of the page. The text-line extraction algorithm detected only parts of these text-lines and hence the remaining part was not included in the detected page frame. The missed errors were due to margin notes and isolated vertical lines.

4.4 Extension to camera-captured documents

In this experiment we extended the proposed approach to work on camera-captured documents. A document captured with a hand-held camera undergoes several distortions like page curl or perspective distortion. In the case of perspective distortion, the page frame can be represented by four straight lines. Under page curl distortion, the top and bottom of the page frame get distorted with respect to the amount of curl on the page, whereas the geometry of the left and right borders is usually not affected. Based on this observation, we focus on adapting the proposed method to find left and right page frame border on camera captured documents as these are needed to remove textual noise from neighboring pages.

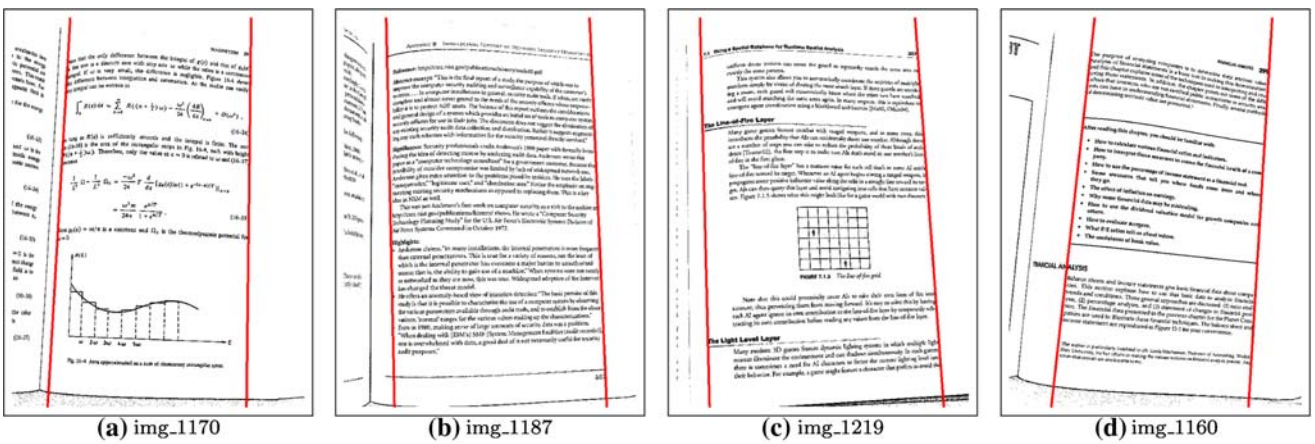
To find the left and right border of the page, the preceding method was modified as follows:

- the *model* is adapted from $\{l, t, r, b\}$ to $\{\theta_l, o_l, \theta_r, o_r\}$, representing the left and right page frame line in normal notation: θ is the angle of the normal vector of the line and o is the length of the normal vector from the line to the origin.
- as *feature points* for the skew-corrected rectangular case, the left and right text line coordinates were used. For this experiment, we extracted curled text-lines from the document images using the approach by Ulges et al. [31]. Since the curled text-line extraction algorithm works better on isolated text regions, we first segmented the page images using the Voronoi algorithm [17]. The text-line extraction algorithm was then run on the segmented image to get text-lines.
- the quality function was modified to use the length of a text-line as its weight:

$$Q(\vartheta_h, L) := \sum_{j=1}^{N} w_j q(\vartheta_h, L_j) \tag{9}$$

where the length of a line was used as its weight. The weight was introduced to make the method robust to small lines originating from speckles. An additional advantage of introducing the weight was that it reduced the effect of small lines coming from the neighboring page in computing the total quality.

- the parameter refinement step was confined to adjusting the page frame for different text alignments. The page frame parameters ϑ_h were extended to include complete



(a) img_1170 **(b)** img_1187 **(c)** img_1219 **(d)** img_1160

Fig. 14 Example of page frame detection for camera-captured document images from CBDAR 2007 document image dewarping contest. **a** Justified text with figures and mathematical formulas, **b** left-aligned text, **c** section headings extending beyond the main text region with center within the text region, **d** section headings extending beyond the main text region with center outside the text region. Our algorithm found the correct page frame in the first three cases. However in (**d**) the section heading was not completely included in the page frame

lines if the middle point of a line was inside the computed page frame.

We evaluated our method on the publicly available CBDAR 2007 document dewarping dataset [32] consisting of 10 training images and 92 test images. The training images were used to develop the algorithm and to test different parameters. For these documents, $\epsilon = 50$ was found to be a good choice. Evaluation based on the percentage of totally in, partially in, and totally out text-lines showed that 95.6% of the text-lines were completely inside, 2.3% text-lines were partially inside, and 2.1% text-lines were completely outside the page frame. Some example results are shown in Fig. 14. Our method correctly found the page frame in Fig. 14a–c despite the large variety of page contents and border noise. In Fig. 14d one text-line was not completely included in the page frame since its center was outside the computed page frame.

5 Conclusion

In this paper we presented an algorithm for page frame detection using a geometric matching method. The presented 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. Several error measures were defined based on area overlap, connected component classification, and ground-truth zone detection accuracy for determining the accuracy of the presented page frame detection algorithm. It was shown that the algorithm performs well on all three performance measures with error rates below 4% in each case. It was also demonstrated that the presented method can handle documents with a very large amount of noise with reasonable accuracy. The error rates on all three performance measures used are below 10% for noise levels up to 80%. 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 and a layout-based document image retrieval system, where it showed a significant decrease in the error rates in both applications. We have released an open source implementation of our page frame detection algorithms as part of the OCRopus open source OCR system.

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

References

1. Cattoni, R., Coianiz, T., Messelodi, S., Modena, C.M.: Geometric layout analysis techniques for document image understanding: a review, Tech. Rep. 9703-09. IRST, Trento (1998)
2. Baird, H.S.: Background structure in document images. In: Bunke, H. Wang, P., Baird, H.S. (eds.) Document Image Analysis. World Scientific, Singapore, pp. 17–34 (1994)
3. Breuel, T.M.: Two geometric algorithms for layout analysis. In: Proceedings of Document Analysis Systems. Lecture Notes in Computer Science, vol. 2423, Princeton, NY, USA, pp. 188–199 (2002)
4. O'Gorman, L.: The document spectrum for page layout analysis. IEEE Trans. Pattern Anal. Mach. Intell. **15**(11), 1162–1173 (1993)
5. Shafait, F., Keysers, D., Breuel, T.M.: Performance evaluation and benchmarking of six page segmentation algorithms. IEEE Trans. Pattern Anal. Mach. Intell. **30**(6), 941–954 (2008)
6. Le, D.X., Thoma, G.R., Wechsler, H.: Automated borders detection and adaptive segmentation for binary document images. In: 13th International Conference on Pattern Recognition, Vienna, Austria, pp. 737–741 (1996)
7. Avila, B.T., Lins, R.D.: Efficient removal of noisy borders from monochromatic documents. In: International Conference on Image Analysis and Recognition, Porto, Portugal, pp. 249–256 (2004)
8. Fan, K.C., Wang, Y.K., Lay, T.R.: Marginal noise removal of document images. Pattern Recognit. **35**(11), 2593–2611 (2002)
9. Cinque, L., Levialdi, S., Lombardi, L., Tanimoto, S.: Segmentation of page images having artifacts of photocopying and scanning. Pattern Recognit. **35**(5), 1167–1177 (2002)
10. Peerawit, W., Kawtrakul, A.: Marginal noise removal from document images using edge density. In: 4th Information and Computer Engineering Postgraduate Workshop, Phuket, Thailand (2004)
11. Stamatopoulos, N., Gatos, B., Kesidis, A.: Automatic borders detection of camera document images. In: 2nd International Workshop on Camera-Based Document Analysis and Recognition, Curitiba, Brazil, pp. 71–78 (2007)
12. van Beusekom, J., Keysers, D., Shafait, F., Breuel, T.M.: Distance measures for layout-based document image retrieval. In: 2nd IEEE International Conference on Document Image Analysis for Libraries, Lyon, France, pp. 232–242 (2006)
13. Shafait, F., van Beusekom, J., Keysers, D., Breuel, T.M.: Page frame detection for marginal noise removal from scanned documents, in: SCIA 2007, Image Analysis, Proceedings. Lecture Notes in Computer Science, vol. 4522, Aalborg, Denmark, pp. 651–660 (2007)
14. Dengel, A., Barth, G., ANASTASIL: Hybrid knowledge-based system for document image analysis. In: Proceedings of International Joint Conference on Artificial Intelligence, Detroit, MI, USA, pp. 1249–1254 (1989)
15. Liang, J., Phillips, I.T., Haralick, R.M.: Performance evaluation of document structure extraction algorithms. Comput. Vis. Image Underst. **84**(1), 144–159 (2001)
16. Das, A.K., Saha, S.K., Chanda, B.: An empirical measure of the performance of a document image segmentation algorithm. Int. J. Document Anal. Recognit. **4**(3), 183–190 (2002)
17. Kise, K., Sato, A., Iwata, M.: Segmentation of page images using the area Voronoi diagram. Comput. Vis. Image Underst. **70**(3), 370–382 (1998)
18. Shafait, F., Keysers, D., Breuel, T.M.: Performance comparison of six algorithms for page segmentation. In: 7th IAPR Workshop on Document Analysis Systems. Lecture Notes in Computer Science, vol. 3872, Nelson, New Zealand, pp. 368–379 (2006)

19. Breuel, T.M.: The OCRopus open source OCR system. In: Proceedings of SPIE Document Recognition and Retrieval XV, San Jose, CA, USA, pp. 0F1–0F15 (2008)
20. Mao, S., Kanungo, T.: Software architecture of PSET: a page segmentation evaluation toolkit. Int. J. Document Anal. Recognit. **4**(3), 205–217 (2002)
21. Okun, O., Pietikainen, M., Sauvola, J.: Robust skew estimation on low-resolution document images. In: 5th International Conference on Document Analysis and Recognition, Bangalore, India, pp. 621–624 (1999)
22. Breuel, T.M.: Robust least square baseline finding using a branch and bound algorithm. In: Proceedings of SPIE Document Recognition and Retrieval IX, San Jose, CA, USA, pp. 20–27 (2002)
23. Breuel, T.M.: A practical, globally optimal algorithm for geometric matching under uncertainty. Electronic Notes Theor. Comput. Sci. **46**, 1–15 (2001)
24. Breuel, T.M.: On the use of interval arithmetic in geometric branch-and-bound algorithms. Pattern Recognit. Lett. **24**(9–10), 1375–1384 (2003)
25. Breuel, T.M.: Implementation techniques for geometric branch-and-bound matching methods. Comput. Vis. Image Underst **90**(3), 258–294 (2003)
26. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions and reversals. Sov. Phys. Dokl. **10**(8), 707–710 (1966)
27. Phillips, I.T.: User's reference manual for the UW english/technical document image database III, Tech. rep. Seattle University, Washington (1996)
28. Breuel, T.M. (1993) Recognition by Adaptive Subdivision of Transformation Space: practical experiences and comparison with the Hough transform. In: IEE Colloquium on 'Hough Transforms' (Digest No.106), pp. 71–74 (1993)
29. Nagy, G., Seth, S., Viswanathan, M.: A prototype document image analysis system for technical journals. Computer **7**(25), 10–22 (1992)
30. Antonacopoulos, A., Gatos, B., Bridson, D.: Page segmentation competition. In: Proceedings of 9th International Conference on Document Analysis and Recognition, Curitiba, Brazil, pp. 1279–1283 (2007)
31. Ulges, A., Lampert, C., Breuel, T.: Document image dewarping using robust estimation of curled text lines. In: Proceedings of Eighth International Conference on Document Analysis and Recognition, pp. 1001–1005 (2005)
32. Shafait, F., Breuel, T.M.: Document image dewarping contest. In: 2nd International Workshop on Camera-Based Document Analysis and Recognition, Curitiba, Brazil, pp. 181–188 (2007)