Proceedings of the 12th International Conference on Pattern Recognition (12th ICPR), Jerusalem, Israel, October 9–12 1994, vol. 2, pp. 339–344.

Flexible Page Segmentation Using the Background

A. Antonacopoulos and R.T. Ritchings

Department of Computation, University of Manchester Institute of Science and Technology,
(UMIST), P.O. Box 88, Manchester, M60 1QD, U.K.

Abstract

This paper introduces a new method for document page segmentation. This method is based on the analysis of the background white space that surrounds the printed regions on the page. It does not make any assumptions about the shape of the regions as opposed to most earlier approaches which assume that printed regions are rectangular. It is capable of identifying and describing regions of complex shapes more accurately than existing methods. It requires no a priori knowledge. The background white space is covered with tiles and the contour of each region is identified by tracing through these white tiles that encircle it. The method can segment page images with severe skew without skew correction. The white tiles on the image can also be used in subsequent document analysis processes such as the classification of the image regions.

1 Introduction

Page segmentation is the identification of areas of interest in the image of a document page. These areas may contain text or graphic information (line drawings or halftones). Page segmentation is concerned only with *identifying* these areas. It is the goal of *page classification* to label these areas according to their contents.

In the majority of document image analysis and understanding applications, page segmentation is performed as the first step. After the areas of interest have been identified in the page image they can be classified according to their contents, their relative positions can be analysed and a description of the layout structure of the page can be obtained. Then, an attempt can be made to understand and describe the logical structure of the page. Alternatively, the layout structure can be used to direct optical character recognition (OCR) to the text regions of the page image.

The performance of a page segmentation method is fundamental in document image analysis. Its accuracy is of central importance to page classification and, subsequently, document understanding or OCR. Badly identified regions can lead to poor recognition results and increased run time in these later stages. Also, the speed of the method itself is very important. As it deals with a very large amount of data, a slower method adds greatly to the overall run time of an application.

On a document page any number of text paragraphs can be found interspersed with any number of graphic features. Both text paragraphs and graphic features can be of *any shape and in any position* in the page.

The dominant earlier approaches to page segmentation have assumed that all printed areas in the page image are *rectangular.* This is limiting not only in the case where these areas are of different shapes, but also where skew has been introduced during the scanning of the document page. In the case of skew, computationally expensive methods may be used to correct it but, when areas of interest are not rectangular, segmentation fails completely. It is not in the objectives of this paper to review all page segmentation techniques. The discussion will be limited to those needed for comparison with the method presented here. Reviews of earlier techniques can be found in [1], [2] and [3].

In general, most page segmentation methods either perform an aggregation of adjacent connected components according to rules, or attempt to isolate printed areas by identifying the white areas of the image that surround them.

Approaches based on the grouping principle can be slow, especially with large amounts of data [2]. Most of these have also relied upon the assumption of the rectangular areas [4][5][6][7]. An approach that is not bound by this assumption is reported in [8].

On the other hand, the background is simpler to analyse than the printed areas in the image. It can, therefore, be faster to do so. Approaches based on this fact adopt varied strategies. One is to use horizontal and vertical projection profiles and successively partition the image into rectangular areas [9]. Clearly, this fails where the rectangle assumption does not hold. Another approach is to identify maximal white (background) rectangles which

1051-4651/94 $04.00 © 1994 IEEE

surround the printed areas [10]. This approach is designed to deal with *Manhattan* layouts, where all printed areas in the image can be separated by horizontal and vertical straight lines. Therefore, it is not appropriate for layouts where a variety of shapes may be found. Furthermore both of the above approaches need to rectify skew before starting the segmentation process.

A recent approach which does not assume rectangular regions is described in [11]. However, it uses an underlying assumption that printed regions are *"surrounded by straight streams of white spaces"* [12]. This can cause regions with sharply varying shapes to appear fragmented. Furthermore, it is designed mostly as an OCR pre-processor. It performs computations locally on very small parts of the image. Hence, as it is, it does not produce any additional global information which could be greatly utilised by the classification and layout determination stages that follow.

In this paper a new flexible method for page segmentation is presented. It uses the *structure* of the background white space that surrounds the connected components in the image. It is flexible because it does not make any assumptions about the shape of the printed regions. Regions with complex shapes can be successfully extracted and with no need to correct skew. It does not employ any computationally intensive procedures. It needs no a priori knowledge about font size etc. A further advantage is that its analysis of the background produces additional global information which is particularly useful for the subsequent document analysis processes.

In section 2 the new method for page segmentation is described. Each of its steps is detailed in a different subsection. In section 3 some experimental results are presented and the performance of the method is discussed.

2 Segmentation by White Tiles

A document page consists of text regions interspersed with regions of graphic features (e.g. line drawings, halftones). All these regions can be of any shape. Around each region there is a stream of white space which delimits it. On the page, all delimiting streams are connected creating a kind of net whose holes are of different shapes and sizes. The holes correspond to the printed material. The idea put forward in this paper is that by reconstructing this *irregular* net of white spaces the *precise* contours of the printed regions can be identified. Furthermore, the construction of the net readily provides global structural information about the layout. This is essential for subsequent stages of document understanding.

The method proceeds as follows. After pre-processing, the white spaces (background) are covered completely by a series of *tiles* of varying sizes so as to follow their shapes very closely. Then, streams of these white tiles whose sides encircle printed regions are identified as belonging to the net of white streams. The surrounded regions are identified by tracing along the region-bordering edges of the white tiles. A more detailed description of the stages of the method and the problems tackled in each one is given in the following subsections.

2.1 Pre-Processing

Location of streams of white space that delimit printed regions may be obvious to a human observer. However, in a page image there is an abundance of white space apart from that belonging to these streams. There is also white space between text lines of the same paragraph, between words and characters of the same text line and inside characters themselves. The goal of this pre-processing stage is to simplify the effort of subsequent stages in identifying the appropriate streams of white spaces. This is done by blocking or isolating white spaces that are not parts of the delimiting streams.

Text lines in a paragraph are printed horizontally (at least for the documents we are referring to here). Then, they can be joined by *vertical smearing* [4]. This will isolate the horizontal streams of spaces between text lines from the region-delimiting streams. At the same time it will fill the white spaces inside characters. However care must be taken in choosing an appropriate smearing value so that the different printed regions are not merged vertically.

Some authors [11] use a static smearing process in which the value is predetermined. This poses problems as different documents are encountered. Others perform lengthy computations such as the Hough transform [13] to determine the smearing value. The method presented here calculates the smearing value directly from the image data.

In a page image, text lines belonging to the same paragraph are vertically spaced apart by practically the same *inter-line* distance. It can also be observed that different printed areas in the image are vertically spaced apart by a distance larger than the inter-line one. The inter-line distance, however, cannot always be easily determined directly from the image data. This is due to the variability in the presence of characters with ascenders or descenders. On the other hand, characters in the same text line sit on the same *baseline* (e.g. *a* in figure 1). Hence, the baselines of vertically adjacent text lines in the same paragraph will be equally spaced apart. Here, the baseline difference is estimated from the distance of the peaks in the horizontal-projection profiles of a few narrow vertical strips of the image. The method developed works reliably for over 10° of skew (see results at end). For very large skew angles the projections can be calculated over angled strips of the image. However, experiments have shown that, in practice, page images with more than 5° skew are not common.

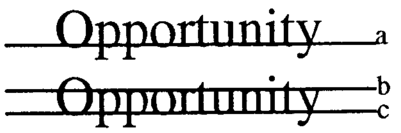

Figure 1. Smearing value determination from baseline difference.

Having determined the baseline difference, the vertical smearing value can now be calculated. Consider the two text lines of figure 1. The baseline difference is the distance between *a* and *c*. If this distance is used as a smearing value there is a possibility of merging different printed regions. The distance between *a* and *b* is chosen to avoid this but at the same time achieving as close a smearing effect as possible. This distance is taken to be 2/3 of the baseline difference.

The effect of smearing on the example image of figure 2 can be seen in figure 3.

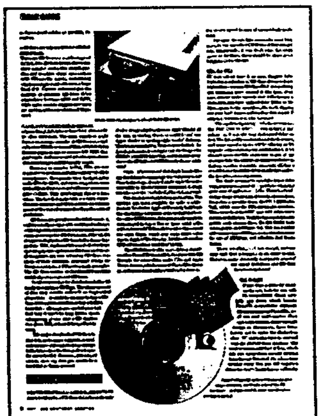

Figure 2. A page with non rectangular layout.

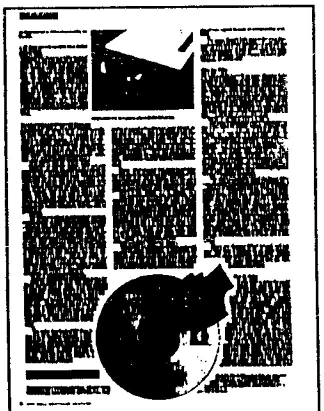

Figure 3. The smeared image of figure 2.

2.2 White Tile Determination

After pre-processing, the next step is to cover all white space in the image by white tiles. A white tile is represented by a rectangle which is vertically and/or horizontally stretched or squashed to fit the longest possible white area in the horizontal direction. No restriction is imposed on the height of the rectangle. If the shape of the white space to be covered varies sharply in the vertical direction, the white tile can be reduced to a horizontal line (i.e. the horizontal sides will coincide).

The algorithm proceeds as follows. Each scan line of the image is considered and each white run is compared to the white runs of the scan line above. If there is no overlapping run above, a new white tile is started. The same happens if there is an overlapping run above but its start and/or its end do not lie close to the corresponding end points of the current run. For example, in figure 4, white run 3 is compared to white run 2 but the difference between their corresponding end points is not acceptable. Hence, a new white tile will be started having run 3 as its first white run.

Otherwise, if there is an overlapping white run above whose end points lie close to the corresponding end points of the current one, the current run is appended to the white tile to which the above run belongs. For example, in figure 4, run 2 was compared to run 1 and it was appended to the white tile to which run 1 belongs, i.e. white tile A. Then, the white tile is augmented to accommodate the new white run.

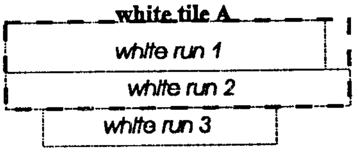

Figure 4. Determination of white tiles.

The new white tile will initially have the width of the starting white run. It can get slightly wider, though, as new runs are appended (see white tile A in figure 4). However, the width of the white tile is not allowed to change by more than a predetermined value. Every time a new run is considered for being appended to the white tile, this value is compared to the accumulated sum of the differences between the corresponding ends of the white runs that belong to the white tile plus the difference between the current run and the one above. If the sum of the differences is larger than the predetermined value, the run is not appended. A new white tile will be started.

The widening of the white tile is done in order to give some flexibility to the fitting process. While a close fit is desired, the computational time will be longer in the subsequent stages if there is a very large number of distinct white tiles. The value of the maximum tolerance of width change is set to be 6 pixels in a 300dpi image. This value corresponds to a width change of half a millimetre on the actual document page. It was experimentally determined that this is sufficient for the white tiles to follow the streams of spaces very closely.

Another point to be made is that because smearing leaves behind a large number of very narrow white areas in the vertical direction, only white runs which are longer than a certain threshold are considered in the determination of white tiles. This threshold is not critical but it should be less than the width of the smallest white tile on the delimiting stream. The threshold used in this method is 1/3 of the baseline difference. In practice, this threshold will suppress inter-character space but preserve some inter-word space. With this arrangement, the white tile determination process produces information which is of particular importance for a page classification approach currently under development.

A number of white tiles which do not actually appear on the image are also created to enable the encircling of printed areas that border the edges of the page image. Each white tile identified is described by a data structure. This holds information about the position of the white tile and about which white tiles are immediately above and below.

The white tiles identified in the smeared image in figure 3 can be seen in figure 5. Overall, the white tilc determination process is fast, using only one sequential pass over the image data. It is also very accurate in covering the streams of white spaces with as few white tiles as possible. At the same time it produces information which facilitates page classification considerably.

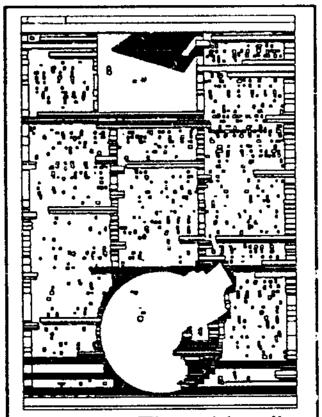

Figure 5. The white tiles on the image.

2.3 Segmentation

At the segmentation stage, the contours of the printed regions in the page are recognised from the white tiles fitted to the streams of white spaces in the image. Each of these contours is a list of white tile edges that border with the corresponding printed region. Each list is cyclic and each element of a list is unique. The objective is to trace the appropriate edges of the white tiles that make up the delimiting white streams of the original image.

The white tile arrangement in the image can be thought of as a graph. The white tiles will be the vertices and the edges of the graph will represent the vertical adjacency between two white tiles. In this graph, then, the problem will be to trace the *minimum* cycles which encircle areas that *do not intersect*. For example, in the graph of figure 6, the wanted cycles will be ABCDAGFEA and AEFGA. In situations like this, standard graph theory is not adequate for identifying the cycles. This is because the cycle ABCDAGFEA contains the vertex A in the middle. By definition, the cycle would not have been recognised as needed. It would end when vertex A is encountered for the second time i.e. ABCDA. It is clear from the white tile arrangement in figure 6 that the white tiles A, B, C, and D do not enclose only one area but *two*.

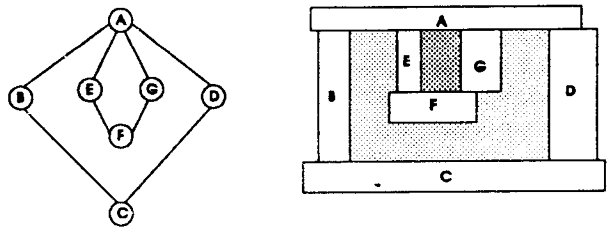

Figure 6. Graph representation of white tile arrangement.

Instead, the approach designed here takes also into account the *edges* of the white tiles that border with the area of interest. In this way, the correct white tile is chosen according to the connectivity of the graph and then the appropriate part of it is added to the list of edges comprising the contour so far.

Tracing cycles in graphs is usually a time consuming task with a lot of computational effort spent in exhaustive search to identify all possible cycles. Moreover, choosing the needed cycles will also add to the overhead.

In contrast, the approach described here is fast. It sequentially recognises only the correct cycles and traces each cycle only once. Furthermore, the search through the white tiles to identify a start of a cycle is performed only once. Once the first white tile is chosen for the tracing to start, there is no need for time consuming searches. All starts of cycles will be identified while tracing other cycles. This makes use of the fact that the streams of white space that delimit regions in the page are connected.

The algorithm proceeds as follows. The white tiles are considered in turn until the first white tile with *potential* start(s) is found. This is a white tile with more than one white tiles below. A potential start is a part of the bottom edge of this white tile that is between two parts covered by the upper edges of the white tiles below. In figure 7, the segments *ab* and *ef* of white tile 1 are potential starts. All potential starts identified throughout the course of the algorithm are placed in a queue. They are not *definite* starts because they may prove to be parts of a cycle that has started elsewhere. This is the case with segment *ef*. When *ef* is reached while tracing the cycle, it will no longer be assumed as a potential start and will be removed from the queue.

From the potential start, the next white tile to be considered is to the left and down. After appending its bordering segment to the cycle, the tracing continues downwards until it reaches a white tile whose contributing segment is a local minimum in the cycle. In figure 7, such will be white tile 7 with the local minima *pn* and *hg*. While going down, there are rules that determine which will be the next white tile whose part(s) will be in the cycle. For example, after white tile 2, which has two below, white tile 7 is chosen. At situations like this, where while going down a white tile with more than one white tiles below is met, its potential start(s) are put in the queue.

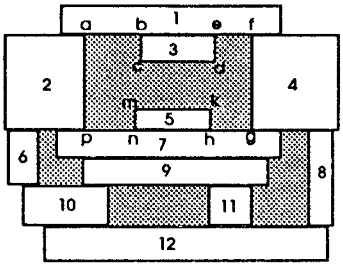

Figure 7. Edges of white tiles that constitute a cycle.

When a local minimum is reached, unless it is at the last white tile at the bottom of the image, it will have at least one white tile below. If there is only one, that one and possibly the ones below it are examined until a white tile is found with more than one white tiles below. The potential starts are then put in the queue. In figure 7, after the local minimum of *pn*, this will lead to white tile 9 and its potential start. It should be said, however, that when the local minimum *hg* is met, the potential start of white tile 9 will not be added to the queue. Every time a potential start is identified while tracing, it is checked upon the already encountered ones (used, discarded or yet to be tried). If it has been met before it is not added to the queue. In this way, no cycle will be traced more than once and no time will be wasted reconsidering potential starts.

After the local minimum, the tracing continues upwards until either a local maximum is reached or the starting segment of the cycle is encountered again and the cycle is closed. As in the case of going down, there are rules which determine which white tile will be the next to contribute part(s) to the cycle. For example, in figure 7, after the local minimum *pn* of white tile 7, the next white tile to be considered will be number 5. When a white tile with more than one white tiles below is met while tracing upwards, its potential start(s) are put in the queue. In figure 7, when white tile 4 is met, the potential start it gives rise to is put in the queue.

When the starting segment is encountered again, the cycle is closed. The next potential start is then retrieved from the queue and another cycle will be traced. In figure 7, after the cycle *(ba,ap,pn,nm,mk,kh,hg,gf,fe,ed,dc,cb)* has been completed, the potential start of white tile 2 will be the next to be tried. The potential start *ef* of white tile 1 will have been discarded as mentioned earlier. Also according to what has already been mentioned, when tracing this new cycle, the potential start of white tile 9 will not be added to the queue as it will already be there.

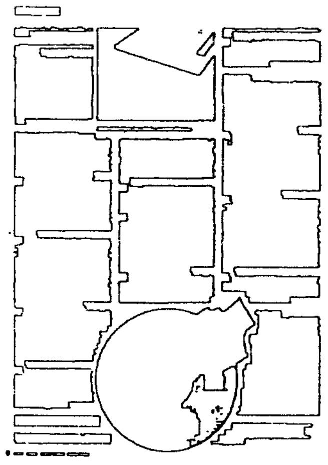

Figure 8. The contours of segmented regions

The contours of the segmented regions of the image in figure 2 can be seen in figure 8. Each of these edge lists follows the shape of the region very closely thus, constituting a quite accurate description of the region. Small segmented regions that belong to the same printed area on the page, can be grouped together if desired. These regions may result mostly from isolated single lines of text and titles.

Overall, the segmentation process is efficient in that the cycles are traced sequentially without the need for exhaustive search. Only the needed cycles are traced and no cycle or part of it is traced more than once. Computation time depends linearly only on the amount of the white tiles covering the region-delimiting streams. Isolated white tiles inside the regions do not affect the process.

3. Results and Discussion

The method described in this paper has been tried successfully on several documents containing printed regions of rectangular as well as of a variety of shapes. Another example of segmentation is that of figure 10 which is the result of the method applied to the image of figure 9. The method has also been tested successfully on page images which were scanned with skew. The segmentation result of figure 12 describes the printed regions of the page of figure 11. It should be noted that the method was applied with no extra processing for skew estimation and the smearing value was adequately determined from the *vertical* strips.

The general idea of segmentation by white tiles can also be applied to pages in which the text lines are printed vertically as in many Japanese documents. In this case the smearing process would have to be horizontal and the white tiles would be fitted with the longest sides in the vertical direction.

The main benefits of segmentation by white tiles are accuracy and speed. It can handle regions of non rectangular shapes, where other methods [9][10] will fail. It also produces more accurate descriptions of the printed areas on the document page. This is an improvement on the method described in [11] where non rectangular regions may appear as a set of disjoint rectangular blocks. Segmentation using white tiles is fast, in that it does not employ any time consuming computations. No skew correction is needed and it does not rely on successive grouping of components or regions. With the use of white tiles a great reduction in the amount of data is achieved. Finally, the cycle tracing algorithm sequentially identifies precisely the needed contours with neither backtracking nor exhaustive search.

A further advantage of segmentation by white tiles is that the white tiles it identifies in the image can be used as the basis for the stages that follow in the document analysis and understanding processes. Work is currently being carried out to show this.

References

[1] Nadler M., "A Survey of Document Segmentation and Coding Techniques", *Computer Vision, Graphics and Image Processing*, 28, 1984, pp. 240-262.

[2] Srihari S.N. and G.W. Zack, "Document Image Analysis", *Proceedings of the 8th International Conference on Pattern Recognition*, Paris, France, 1986, pp. 434-436.

[3] Tang Y.Y., C.Y. Syen, C.D. Yan and M. Cheriet, "Document Analysis and Understanding: A Brief Syrvey", *Proceedings of the First International Conference on Document Analysis and Recognition*, 1, Saint Malo, France, Sept. 30-Oct. 2, 1991, pp. 17-31.

[4] Wahl F.M., K.Y. Wong and R.G. Casey, "Block Segmentation and Text Extraction in Mixed Text/Image Documents", *Computer Graphics & Image Processing*, 20, 1982, pp. 375-390.

[5] Ciardello G., M.T. Degrandi, M.P. Roccotelli, G. Scafuro and M.R. Spada, "An Experimental System for Office Document Handling and Text Recognition", *Proceedings of the 9th International Conference on Pattern Recognition*, Rome, Italy, Nov. 1988, pp. 739-743.

[6] Lam S.W., D. Wang and S.N. Srihari, "Reading Newspaper Text", *Pattern Recognition, 1, Proceedings of the 10th International Conference on Pattern Recognition*, 16-21 June 1990, Atlantic City, New Jersey, U.S.A., pp. 703-705.

[7] Amano T., A. Yamashita, N. Itoh, Y. Kobayashi, S. Katoh, K. Toyokawa and H. Takahashi, "DRS: A Workstation-Based Document Recognition System for Text Entry", *IEEE Computer*, July 1992, pp. 67-71.

[8] Saitoh T. and T. Pavlidis, "Page Segmentation without Rectangle Assumption", *Proceedings of the 11th International Conference on Pattern Recognition*, 31 Aug. - 3 Sept. 1992, 2, Le Hague, Netherlands, pp. 277–280.

[9] Nagy G., S. Seth and S.D. Stoddard, "Document Analysis with an Expert System", *Pattern Recognition in Practice II*, North-Holland, 1986, pp. 149-159.

[10] Baird H.S., "Background Structure in Document Images", *Advances in Structural and Syntactic Pattern Recognition*, H. Bunke (ed.), World Scientific, 1992, pp. 253-269.

[11] Pavlidis T. and J. Zhou, "Page Segmentation and Classification", *CVGIP: Graphical Models and Image Processing*, 54, no. 6, November 1992, pp. 484-496.

[12] Pavlidis T. and J. Zhou, "Page Segmentation by White Streams", *Proceedings of the First International Conference on Document Analysis and Recognition*, 2, Saint Malo, France, Sept. 30-Oct. 2, 1991, pp. 945-953.

[13] Fisher J.L., S.C. Hinds and D.P. D'Amato, "A Rule-Based System for Document Image Segmentation", *Pattern Recognition, 1, Proceedings of the 10th International Conference on Pattern Recognition*, 16-21 June 1990, Atlantic City, New Jersey, U.S.A., pp. 567-572.

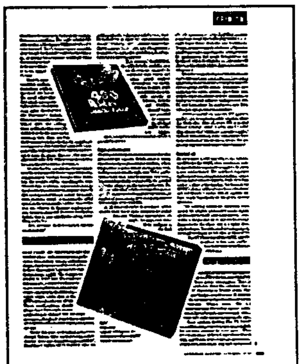

Figure 9. Original.

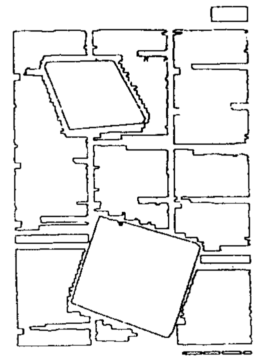

Figure 10. Result.

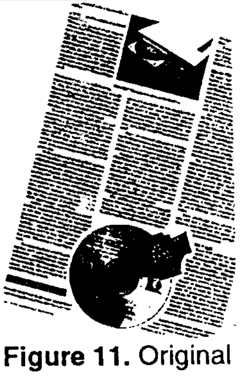

Figure 11. Original image with 15° skew.

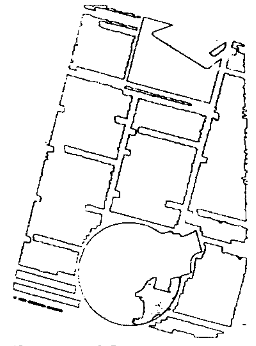

Figure 12. Segmentation result.