

# Page segmentation and classification using fast feature extraction and connectivity analysis

Jaakko Sauvola and Matti Pietikäinen  
Department of Electrical Engineering  
University of Oulu  
FIN-90570 OULU, FINLAND  
e-mail: jjs@ee.oulu.fi

## Abstract

Page segmentation and classification are important parts of the document analysis process. The aim is to extract and classify different parts of the page. This paper proposes an approach in which these two phases are combined. The integration process includes fast feature extraction with rule-based classification and label propagation using connectivity analysis providing classified areas in three categories: background, text and picture.

**Keywords:** segmentation, classification, fast feature extraction, rule-based reasoning, connectivity analysis.

## 1. Introduction

There is an ever growing demand for electronic and computerized document management, storage and circulation. This has been a problem for governments and corporations in the past, but nowadays also considerable consuming masses, ordinary people, have grown to demand these basic tasks. Document structure analysis and understanding are the main processes in reaching this goal: ease of use and availability of documents. The latest OCR software products include efficient storage and retrieval facilities.

In order to achieve the best possible results with OCR and storing, the contents of the document have to be examined. Many different methods have been proposed to do just this task: Pavlidis and Zhou [1] present a method based on smeared run-length codes that may be used to divide the document. Watanabe et al. propose a method for document structure recognition using individually characterized document knowledge [2], Jain and Battacharjee a method based on gabor filters [3], Antonacopoulos and Ritchings a method using document background for page segmentation [4] and Etemad et al. a method for page segmentation using decision integration and wavelet packets [5], respectively, to name but a few.

## 2. System overview

We have developed a method that is an important part in a chain of procedures meant to improve the quality of a document before OCR (Figure 1), [6]. The document is first thresholded with adaptive thresholding based on two-dimensional signal correlation proposed in [7] and modified by the present author [6]. Then the skew error is determined with a new method based on texture direction analysis [8]. After skew correction the image can be enhanced if needed [6].

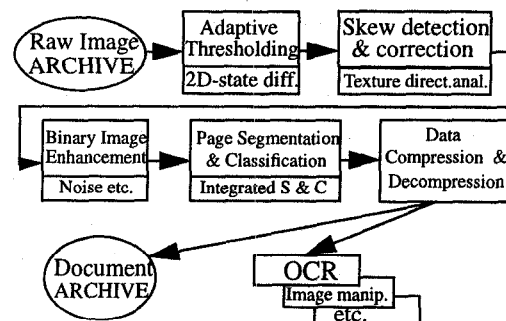


Figure 1. Document analysis, storing and text recognition system.

Finally, the contents of the page are searched and classified with a segmentation and classification procedure, PCS (Page, Classification, Segmentation), considered in this paper. The results of the PCS procedure are compressed into an archive for further use which can be recognition of text or document image processing, for example. The tests have shown that the result of the OCR improves with this enhancement procedure [6].

The PCS procedure combines the segmentation and classification parts. The method is divided into steps, designed such that one can add or take away features easily and therefore affect the properties and aim of the process. For this reason our method can easily be adapted to various situations. The first step is to divide the document into small windows in which the desired features are calculated. We

used four different features described in Section 3. After extracting the features from each window, a move to a higher abstraction level is made: all the operations from now on are performed in a window map area, instead of performing them at the pixel level. This greatly simplifies the processing and reduces the amount of computing. A set of rules is used to decide the label of each window. Then, an iterative connectivity analysis process based on carefully designed 3x3 and 4x4 masks is applied to the pre-classified window map. The aim of the masks is to propagate the labels such that dominating classes are formed in the window map area. Because the iteration is performed on the window map, the calculation time is not an issue, even if many different masks are used. The masks have been developed to form rectangular shaped areas around classified text or picture objects. After this process is completed, the formed areas are extracted from the image by finding the corners of these boundaries. This process is planned such that it tolerates quite a severe skew, although it is not necessary when using the skew detection/correction procedure described above. All the operations are performed quickly because of the reduced amount of data: for example, if the image area is 1000x1000 pixels and the window size is 20x20 pixels, we can drop the basic calculations down to 0.25% (1 million cells compared to 2500 cells) compared to the amount of calculations at pixel level! This feature enables the addition of different types of operations.

### 3. Feature extraction

The image is first divided into small  $n \times n$  pixel windows, where  $n$  is determined according to scanning resolution and the size of the image. Ten to twenty were used for values of  $n$  in our tests. The purpose of this division is to gain 'atomic'-like components for fast feature extraction purposes and connectivity manipulation. After the whole image has been windowed, one can proceed to feature extraction.

The created windows must be small enough to distinguish different areas (text, column gaps etc.) from each other, and yet big enough to gain reliable feature characteristics. In our experiments, we used four simple features to classify each window to text or picture. These characteristics were black/white-ratio inside the single window, average black (thresholded) run-length and vertical cross-correlation between neighbouring pixels and between the first and every fifth (relative) pixel.

Every one of these features provides useful information for generating a decision procedure dependent on them. The b/w-ratio gives an estimate of the amount of black pixels compared to the white ones. When a picture is concerned, this ratio gets closer to 100%. This ratio can be judged by analyzing various thresholded pictures. It does

no harm to set a single window inside the picture to text or background due to the structure of the propagation process after this pre-classification. In our tests, we used a value of 85% or higher for pictures, 2-85% for text and under 2% for background (due to noise factor elimination). The value 85% was selected in order to allow noise or scanning errors to be eliminated. The same reason was the main argument for using 2% for the background value discrimination from text: to put up with noise and errors. With average black run-length, we can distinguish text from background. Values 0 or 1 were used for the background, a value of more than 1 for text and for a picture it is not defined. Since the run-length is averaged inside the window, the general assumption can be made to rule in the current window: if the average is 0 or 1 it can be ruled to background. From the average of 0 or 1, the assumption can be made that the window has no black pixels (0) or very few (1), located randomly. In the latter case, these pixels can be ruled to noise. If the average is more than 1, the assumption can be made that the window includes text or text-like components. At the higher end, the difference limit between text and picture is more complex to distinguish and therefore this judgement is left unused. The other features can be used instead.

Cross-correlation can divide fast changes in vertical pixel lines in order to determine whether one vertical pair of lines accumulated within a window includes text-styled layout or is more homogeneous, when it can be either background or a picture. The limit between text and background/picture is not very stable and thus the limit must be set according to the window and image size (scanning resolution). We used a scaled value 0.97 as a general limiter, which allows the noise factor to be taken into consideration. The tolerance was set at +/-0.03 to take into account the environment mentioned above. This tolerance gives a bigger window and image has more room to have noise or errors and adaptively tightens the limit when smaller window and lower resolution images are used.

#### a) Black/white-ratio scaled between 0-1

B/w-ratio is calculated from each  $n \times n$  pixel window. The amount of black pixels (ABP) is counted, compared to the maximum amount of pixels ( $n \times n$ ) and scaled in the range 0-1:

$$\frac{b}{w} = \frac{ABP}{n^2} \quad (1)$$

#### b) Average horizontal black run-length

The average horizontal black run-length is calculated for

each horizontal scan-line inside the window. Each window is processed separately. This means that if the line starts or ends with a black pixel and if a window before or after the present window ends or starts with a black pixel, these pixels are ignored. Only the pixels inside the current window are processed. A table is formed in order to keep count of the lengths. After the whole window has been processed the average is counted from the table. This number acts as the average black run-length.

#### c) Signal cross-correlation

Signal cross-correlation is calculated vertically within one and five pixels distance. Pixels get the value 0 or 1 depending on their color (black=1 and white=0). A normalized cross-correlation is used between scanlines  $y$  and  $y+a$ , where  $y$  is a vertical location and  $a$  is added to this location. We get:

$$C_r(a, y) = 1 - \frac{2}{M} \sum_{k=0}^{M-1} p(y, k) \text{ XOR } p(y+a, k), \quad (2)$$

where

- $M$  is the width of the window,
- $p(y, k)$  is the  $k$ th binarized pixel in the vertical line  $y$ ,
- $a$  is a horizontal distance value set for vertical pair of pixels.

If pixels have different values,  $C_r(a, y)$  gets value -1. If the pixels get the same value,  $C_r(a, y)$  gets value 1. These values are accumulated to each window separately. Cross-correlation inside the window is calculated in the  $x$ - $y$  plane such that  $x$ 's value increases by one always when a vertical pair of lines with  $y$  and  $y+a$  is calculated.

## 4. Classification rules

After extracting all the desired features, the classification rules are formed. As discussed earlier, the contents of the page are classified into three different classes: background, text and picture. The addition of new classes is easy, just by selecting features accordingly, calculating them in a feature extraction block and finally making the new class in this stage.

Following classification rules are applied for the selected three classes:

#### a) Background

- B/w gets a value less than 0.02.
- The average black run-length is 0 or 1.

- The cross-correlation between one pixel distance is 0.97 or greater.
- The cross-correlation between five pixels distance is 0.97 or greater.

#### b) Text

- B/w gets value greater or equal than 0.02 and less or equal than 0.85.
- The average black run-length is greater than 1.
- The cross-correlation between one pixel distance is less than 0.97.
- The cross-correlation between five pixels distance is less than 0.97.

#### c) Picture

- B/w gets value greater than 0.85.
- The average black run-length is not defined.
- The cross-correlation between one pixel distance is 0.97 or greater.
- The cross-correlation between five pixels distance is 0.97 or greater.

These values have been found with tests using various types of test windows. If one would like to define more features, as discussed earlier, it is easy to adapt the sequence to new ones simply by adding a new rule and setting new limits.

If a window cannot be classified to any of the above classes, we have ruled it to the class 'background'. The window in question will be dealt with later with the help of connectivity analysis, where background windows get the general 'don't care' label in the label extraction.

## 5. Determination of window connectivities

After the windows are classified the connectivity analysis is next. We do not use the original picture data any more until at the very end when the window information is linked to the picture data. From this stage on, only the classified windows are used. This manoeuvring reduces the computation time and simplifies the dataset at hand for further classification. The purpose for connectivity analysis is to expand or reduce text and picture labelled window areas so that they form unified rectangular shapes recognized as one solid area within the original image. For this purpose, several different basic 3x3 and 4x4 (windows) masks have been developed. The masks are applied iteratively. We used 20 to 200 iterations in our experiments. The big range within iterations can be explained with the solidity factor: the more iterations, the more solid the produced areas are. The optimum limit can be set according to image and window size: the bigger the image and the smaller the window,

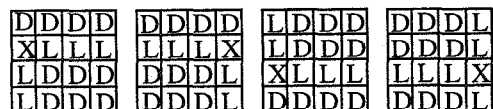
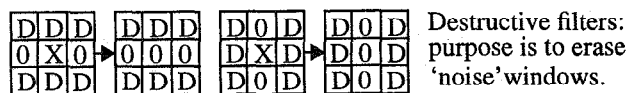
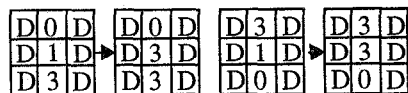
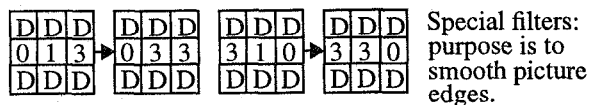
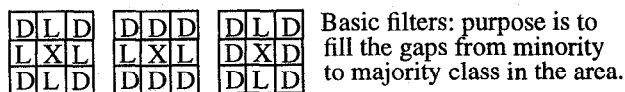
the more iterations are needed.

The set of masks are constructed partly from basic masks and partly from a few masks specially designed for this purpose. Some of the masks are presented in Figure 2.

Basic 3x3 filters are applied first one by one together with 4x4 filters (Figure 2 top-most 3x3 and at the bottom 4x4 filters). After this set is iterated the special filters (Figure 2) are applied in a similar manner.

This process is easy to adapt to new situations, since one only has to add a new mask in the iteration phase.

Because all the masking can be done by comparison (boolean operations), no time demanding multiplication is needed. Each mask is applied to the window map at a time as described above, and changes are made into the originally empty window (temporary stock) map. After each iteration step, the changed window labels are added from the temporary map to the original map. This enables changes to take effect before next iteration loop.



Special filters: purpose is to expand found corners in order to form unified rectangles.

In the masks,

- X = window to be examined,
- D = don't care,
- L = window labeled other than X,
- 0 = background label,
- 1 = text label,
- 3 = picture label.

Figure 2. The mask set.

Text and picture areas are processed separately. The very last operation is done with masks that erase windows that are left alone in the background (Figure 2, destructive filters). Such windows are considered to be 'noise' that exists due to scanning errors or poor document quality. After extracting and erasing processes have been completed, the original window map area should include uniform rectangular shaped areas, even if all the found and classified components are not the same shape (frames can

be seen around each classified area, see e.g. Fig. 4). The next part is to find these areas and to determine geometric characteristics of each area.

## 6. Determination of text area corners

Because each formed area gets a label that describes its contents, the text areas are easy to search from the window map. We can use this map to find the edges of the classified areas. This is done by matching with the 3x3-masks specially designed for this purpose shown in Figure 3.

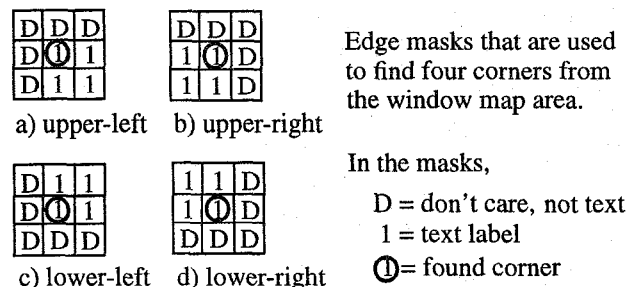


Figure 3. Corner masks for mask-matching.

The found corners are stored in a new map, called a corner map. From the corner map all the upper-left corners, identified with a label 'a' are searched. Each of these labels can be linked to the original picture, from where the x-y-coordinates of the upper-left corner of the area can be calculated. After the upper-left corners have been found, the area width and height must be determined. This is done by linking the upper-left corner found to the nearest right and bottom labels. The dimensions can be calculated directly from the corners found and the upper-left corner by subtracting.

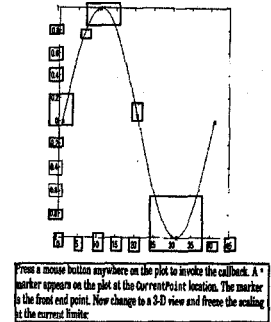
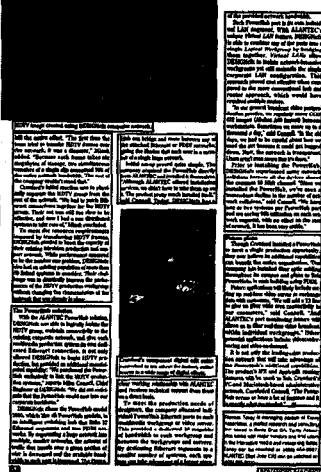
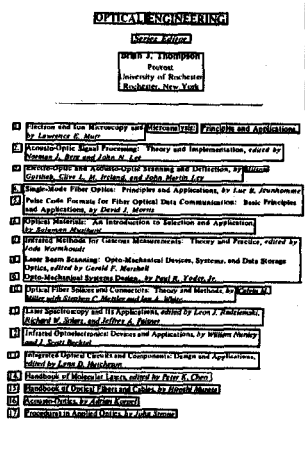
After forming rectangle coordinates, the areas left inside the rectangles can be extracted from the original picture for further processing, including OCR text recognition or data storage.

## 7. Experimental Results

The proposed segmentation and classification procedure was tested with several images, some of which are shown in the Figure 4. The tests were implemented in a SUN SPARCstation 10 in a Khoros environment.

In our experiments, we were able to segment and classify complex text and picture regions accurately. Since the aim was to extract text regions for the OCR, we focused the masks to do just that.

The percentage of the extracted text was 99-100% of all text in test images. This percentage includes complex text regions and text in pictures.



The text blocks have been extracted, the picture blocks and the background is ignored in the text extraction process.

Pictures can be extracted by defining them as targets for extraction in the produced command file.

Figure 4. Sample images. A result of segmentation and classification.

### 8. Conclusion

Page segmentation and classification is an important part of the document analysis process. In this paper, we proposed a new method which integrates both parts as one. This is achieved by fast feature extraction at a pixel level and by moving immediately to a higher abstraction level using a window map which is linked to the original picture data at the end of the process. This mapping technique reduces the amount of data and simplifies the classification process, which is done by propagating the window labels. As a result we get a document that is classified with background, text and picture labels. These areas can be used separately for further processing or data storing. The advantages of this procedure include reduced calculation time, with increased versatility of applied features, decision classes and propagation masks. The adaptation to new situations can be achieved very easily, since each part of this procedure is modifiable. The move to a higher abstraction level (windows) simplifies the source data at hand, which makes it easy to develop masks for different purposes.

### 9. Acknowledgements

The authors wish to thank Professor Theo Pavlidis for e-mail discussions and fruitful comments during the course of investigations.

### References

- [1] Pavlidis T. and Zhou J. (1992) Page Segmentation and Classification. In: CVGIP: Graphical Models and Image Processing, November 1992, Vol. 54, pp. 484-496.
- [2] Watanabe T., Qin L. and Sugie N. (1993) Structure Recognition Methods for Various Types of Documents. In: Machine Vision and Applications, Vol. 6, pp. 163-176.
- [3] Jain A.K. and Bhattacharjee S. (1992) Text Segmentation Using Gabor Filters for Automatic Document Processing. In: Machine Vision and Applications, Vol. 5, pp. 169-184.
- [4] Antonacopoulos A. and Ritchings R.T. (1994) Flexible Page Segmentation Using the Background. In 12th International Conference on Pattern Recognition, Vol. 2, Jerusalem, Israel, pp. 339-344.
- [5] Etemad K., Doermann D. and Chellappa R. (1994) Page Segmentation Using Decision Integration and Wavelet Packets. In 12th International Conference on Pattern Recognition, Vol. 2, Jerusalem, Israel, pp. 345-349.
- [6] Sauvola J. (1994) Document Structure Analysis and Methods, A Masters thesis, University of Oulu, Department of Electrical Engineering, Finland (in Finnish).
- [7] Yang J., Chen Y. and Hsu W. (1994) Adaptive Thresholding Algorithm and It's Hardware Implementation. In: Pattern Recognition Letters, Vol. 15, pp. 141-150.
- [8] Sauvola J. and Pietikäinen M. (1995) Skew Angle Detection Using Texture Direction Analysis. In: 9th Scandinavian Conference on Image Analysis, Uppsala, Sweden.