Difference of Boxes Filters Revisited: Shadow Suppression and Efficient Character Segmentation

Erik Rodner Herbert Süsse Wolfgang Ortmann
Joachim Denzler
Chair for Computer Vision, Friedrich-Schiller University Jena

http://www.inf-cv.uni-jena.de

E-mail: {rodner,nbs,noo,denzler}@informatik.uni-jena.de

Abstract

A robust segmentation is the most important part of an automatic character recognition system (e.g. document processing, license plate recognition etc.). In our contribution we present an efficient segmentation framework using a preprocessing step for shadow suppression combined with a local thresholding technique. The method is based on a combination of difference of boxes filters and a new ternary segmentation, which are both simple low-level image operations.

We also draw parallels to a recently published work on a ganglion cell model and show that our approach is theoretically more substantiated as well as more robust and more efficient in practice. Systematic evaluation of noisy input data as well as results on a large dataset of license plate images [1] *show the robustness and efficiency of our proposed method.*

Our results can be applied easily to any optical character recognition system resulting in an impressive gain of robustness against nonlinear illumination.

1 Introduction

The main characteristic of natural scenes is unpredictable illumination and as a consequence shadows whose existence and location in the image cannot be predicted or modeled in a reasonable way without significant effort. In object recognition shadows can be dealt with by either integrating illumination in the model of the object or by eliminating shadows or shading effects in a preprocessing step.

In the area of character recognition, modeling illumination is a difficult problem, since the appearance of the document does not only depend on the illumination in the scene alone but also on the geometry of pages, dirt or reflection (Figure 1).

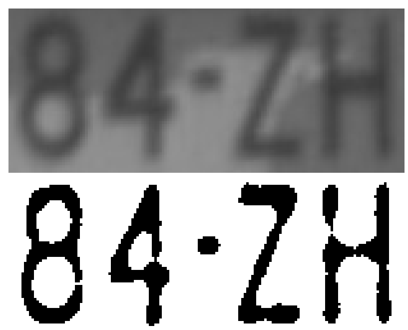

Figure 1. Example of a group of characters, which are difficult to segment due to hard and continuous shading. Segmentation result after applying our segmentation framework.

Additionally, current segmentation approaches have difficulties with low resolution characters. Due to high cost of gathering ground truth data, training sets are restricted to small amount of examples (10-50) per character, which is far too few for estimating complex illumination models. As a consequence, eliminating shadows in a preprocessing step of a segmentation framework seems to be the more promising approach in such applications.

Classical segmentation frameworks simply use global and/or local adaptive methods [5, 6, 3] for binarization, which are often not invariant to shading effects. Vonikakis et al. [12] proposed to use differences of smoothing filters to eliminate shading. The authors motivated their ap-

[1] We would like to thank ROBOT Visual Systems GmbH for financial support and for providing with experimental data for large scale evaluation.

proach by the ganglion cells of the human visual system. We show in the following the simple "theoretical" background of this method which leads to the use of multiple DoB filters (MDoB). We extend their approach to a complete segmentation framework, which uses a new adaptive contour segmentation method to efficiently compute a final segmentation of a MDoB filter output.

The quantitative evaluation of generic segmentation algorithms is a demanding task. State-of-the-art methods often try to develop a similarity measure of segmentation boundaries [8]. These methods always estimate the segmentation quality of the whole image. In the case of character segmentation, a good separation of characters is often sufficient. Independent artefacts can be easily rejected by subsequent classifiers and are therefore not influencing the quality of the character segmentation.

In spite of these well defined requirements, evaluation of character separation is often done by comparing resulting recognition rates of subsequent text classification [10, 1]. This methodology is strongly dependent on the classification methods utilized and which kind of examples are used to train the character classifier. For example the output of an arbitrary segmentation method might provide a good separation of characters, but because of systematic thinning of characters the classifier would perform poorly, if trained with thick characters. Therefore this strategy does not lead to an objective evaluation of methods beyond a special application.

For this reason we introduce a new measure for the comparison of segmentations especially within character recognition systems. On the one hand this can be used for the adaptation of our system to a new application. On the other hand this measure allows us to evaluate our own method with respect to given ground truth data.

We will first describe the basic idea and the simple nature of MDoB filters. In our segmentation framework the filter output is then passed to a locally adaptive contour segmentation, which is explained in section 3. We continue with the optimization of parameters and a new evaluation criterion for character separation in section 4.

Finally we will demonstrate the simplicity and effectiveness of our method within a license plate recognition system and present an in-depth analysis of shading invariance in section 5. A summary of our framework will conclude the paper.

2 Multiple Difference of Boxes

We will initially describe a single difference of boxes filter and its relations to other well known filters used for edge detection. This leads to a simple combination of multiple difference of boxes filters which is the base of our shadow elimination preprocessing.

2.1 DoG and DoB filters

There are a lot of well known standard edge detection techniques within the image processing domain, see e.g. [2]. One of these standard filters is the so called LoG filter or Mexican-Hat. If the output of the LoG filter at a pixel is near zero the pixel is regarded as a candidate for an edge.

LoG filters can be approximated with a difference of two Gaussian filters (DoG filter), which leads to computational benefits and is sufficient in most applications. For an even faster implementation of DoG filters, Rosenfeld in 1971 [9] proposed the replacement with difference of boxes filters (DoB), but also showed that this leads to a huge localization error of edges, compared to the original DoG filter.

Character separation does not need exact localization of edges, but a coarse detection of strokes with a fixed width. This can be done by applying a DoB filter, which is defined by the filter sizes m and M of the small and large box filter. With g_k being a one dimensional gray value image, the shape of a DoB filter is given by:

$$\begin{aligned}\mathrm{DoB}_{m,M}(g) &= \frac{1}{m}\sum_{i=1}^{m} g_i \;-\; \frac{1}{M}\sum_{l=1}^{M} g_l \qquad (1)\\ &= \frac{M-m}{M}\left(\frac{1}{m}\sum_{i=1}^{m} g_i \;-\; \frac{1}{M-m}\sum_{l=m+1}^{M} g_l\right)\end{aligned}$$

A DoB filter is not an approximation of the Mexican-hat filter, because m and M can be arbitrarily specified. The exact shape of a two dimensional DoB-filter is displayed in Fig. 2

The filter output is maximal if we match a similar shape in the image, for example a stroke of a character. Therefore we do not need negative outputs of the filter. We scale only the positive outputs and choose a simple normalization of the outputs, if the local background is dark then we enhance the filter-output.

2.2 Combination of multiple DoB filters

In some situations the stroke width of characters is not yet known a priori. Therefore we use a combination of DoB filter outputs which we will call MDoB filter. One possibility to combine the filter outputs is to calculate the maximum at each pixel. After this operation, peaks in a filtered image correspond to character strokes of different sizes.

2.3 Efficient implementation

Vonikakis et. al. [12] proposed an algorithm which depends on the filter sizes m and M. Their algorithm used DoB filters implicitly, but did not used the benefits of these

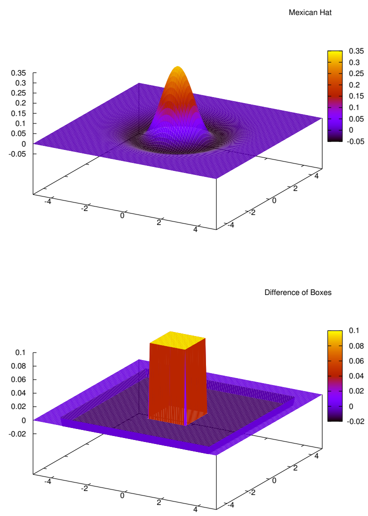

Figure 2. Comparison between Mexican-hat and difference of boxes filter.

filters as a simple combination of box filters. Therefore the implementation of DoB filters can be reduced to the efficient implementation of box filters, which is well known [2] and can be done in linear time in the size of the image independent of filter sizes m and M. This is also reflected by the fact, that DoB filter outputs can be calculated with the use of integral images [11].

3 Local Segmentation

The result of the preprocessing step using MDoB filters can be regarded as a gray value image. This step eliminated shading and noise effects and the resulting image has to be segmented, i.e. each image pixel has to be labeled as "object" or "background". We will present a simple low-level image operation which is robust and sufficient for many situations.

A common approach is global thresholding: Pixels are regarded as object pixels if and only if their values are above a given threshold. After preprocessing with an appropriate MDoB filter even this simple method (global threshold estimated by the approach of Otsu [2]) leads to good results. In comparison to global thresholding, locally adaptive threshold methods provide segmentations, which are more robust to illumination and different object gray values [1, 10].

Local thresholds are always determined by an analysis of the neighborhood $\mathcal{U}(\mathbf{x})$ of a pixel $\mathbf{x}$, which has one drawback: A hard decision between object and background is not possible for all pixels. If the neighborhood contains only pixels from one class (object or background) an estimation of a threshold is impossible. It is therefore essential to define a third label "unknown" for these pixels. Besides finding an algorithm for calculation of a local threshold one has to find a way to deal with "unknown" points.

For the estimation of a local threshold one can use all known algorithms for calculating a threshold in an image, but not all of them are really useful: Models, valid for the whole image may be inappropriate for small areas. On the other hand one has to take into account, that the estimation of the level has to be done for all pixels. Expensive algorithms thus lead to long computation times of the whole segmentation.

An approach that meets the needs for fast execution and good segmentation is the usage of the maximum and minimum of pixel values in the neighborhood:

$$g_{\min}(\mathbf{x}) = \min_{\mathbf{y}\in\mathcal{U}(\mathbf{x})} g(\mathbf{y}) \tag{2}$$

$$g_{\max}(\mathbf{x}) = \max_{\mathbf{y}\in\mathcal{U}(\mathbf{x})} g(\mathbf{y}) \tag{3}$$

In case of a low difference of minimum and maximum in the neighborhood one can assume, that there are only pixels of the same class in $\mathcal{U}(\mathbf{x})$ and small differences are regarded as the result of noise. The pseudo code of the algorithm for each pixel is given in Figure 3.

1 `Calculate` $g_{\max}$ `and` $g_{\min}$

2 `For each pixel` $\mathbf{x}$:

  2.1 `If` $g_{\max}(\mathbf{x}) - g_{\min}(\mathbf{x}) < \gamma$ `then`

    2.2 `label point as unknown`

  2.3 `else`

    2.4 $T = \frac{1}{2}\left(g_{\max}(\mathbf{x}) + g_{\min}(\mathbf{x})\right)$

    2.5 `If` $g(\mathbf{x}) > T$ `then label point as object`

    2.6 `else label point as background`

Figure 3. Algorithm for locally adaptive ternary segmentation.

Figure 4 shows the result of ternary segmentation for an artificial image using a neighborhood of 11 by 11 pixels. This shows, that local segmentation can work well, where global thresholding would fail.
Local segmentation shows some specifics:

a) Small objects or small background areas are detected as usual (An object/background area is called "small"

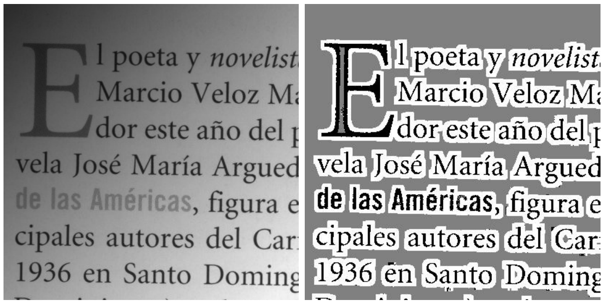

Figure 4. Left: Original image. Right: Result of local segmentation: object pixels (black), background (white), unknown segmentation (gray).

if a background pixel exists in the neighborhood of all object pixels and vice versa.)

b) The detection of the boundary of big objects with good contrast is nearly perfect, while in the interior pixels, which are far away from a background pixel, are labeled as unknown

c) Objects may be partially detected, if there are parts of the object with enough contrast to the background, while others parts disappear because of low contrast.

Contour search is often used for the subsequent process of connected component analysis. Normally this involves following an edge, which is distinguished by object/background pairs of pixels. In case of ternary segmentations, contour search has to stop at unknown pixels. Therefore resulting contours can be either open or closed cycles. With respect to the above mentioned cases this means:

- In cases a) and b) the contour is closed. There is no need for any special care.
- In case c) there are one or more open contours associated with an object. These contours have to be excluded from the segmentation or corrected. For practical issues closing small gaps between start and endpoints can increase the segmentation performance significantly.
- Pixels outside the image can be initialized as "unknown". This means that objects touching the border of an image can be detected as open contours. The user can decide to eliminate these objects or to close the contours.

4 Efficient Parameter Estimation

The multiple difference of boxes (MDoB) filter introduced in section 2 can be used in arbitrary character recognition systems. Each system has its own average character size, stroke width and typical shading effects. It is therefore advantageous to use optimal filter sizes of the MDoB filter. We will first present a new criterion to measure the quality of character segmentations. This criterion is then used as the objective function for the parameter estimation.

4.1 Comparison of Character Segmentations

The comparison of different segmentations, like comparing a ground truth with an automatically generated segmentation, is a highly ill-defined problem and therefore a demanding task. Current research often concentrates on generic segmentations and tries to find a suitable measure comparable to human results [8]. Within character recognition applications we are able to use a more specific formulation. A good segmentation could be defined as the segmentation which is best appropriate to following steps. For example characters have to be well separated and each has to correspond to one specific connected component.

At first let us define an error measure $d_{\mathcal{R}}$ between two connected components A and B of a segmentation. A typical measure, which is widely used for a simple comparison of image regions is

$$d_{\mathcal{R}}(A,B) = \frac{|A \setminus B| + |B \setminus A|}{|A| + |B|}. \tag{4}$$

Given two segmentations $\mathcal{S} = \{\mathcal{S}_k\}_{k=1}^{p}$, $\tilde{\mathcal{S}} = \{\tilde{\mathcal{S}}_k\}_{k=1}^{q}$ as sets of connected components, we have to find an optimal matching, which minimizes the region error $d_{\mathcal{R}}$:

$$d(\tilde{\mathcal{S}}, \mathcal{S}) = \frac{1}{q}\left(\min_{\pi} \sum_{k=1}^{q} d_{\mathcal{R}}(\tilde{\mathcal{S}}_{\pi(k)}, \mathcal{S}_k)\right) \tag{5}$$

with $\pi : \{1, \ldots, q\} \to \{1, \ldots, p\}$ being a total injective map for the case of $p \geq q$ and a simple injective function for $p < q$. In the latter case the sum is added by $(q - p)$ which represents connected components of $\mathcal{S}$ which are missing in $\tilde{\mathcal{S}}$. This optimal matching can be found easily with the Hungarian method [4], which solves the assignment problem.

4.2 Optimization

We will now formulate an optimization framework, which allows us to efficiently estimate or train MDoB parameters from few ground truth segmentations $\mathcal{S}_G^i$. The

License Plate Recognition with **6205** images	
MDoB filters + Local Segmentation	$\mathbf{88.45\%}$
Local Segmentation	73.47%

Table 1. Recognition rates for complete license plates of our license plate recognition system using MDoB filters and local segmentation. Standard global segmentation techniques are not appropriate for this task and are thus not evaluated.

output of our segmentation slightly depends on two parameter sets: filter sizes of the MDoB filters $\theta = \{m_1, M_1, \ldots, m_k, M_k\}$ and parameters of the ternary segmentation method $\eta = \{\gamma,\ \text{size}(\mathcal{U}(x))\}$. To find optimal parameters we simply optimize our measure for segmentation errors as described in equation (5):

$$\epsilon(\theta, \eta) = \sum_i d(\ \tilde{\mathcal{S}}^i(\theta, \eta),\ \mathcal{S}_G^i\ ) \tag{6}$$

with $\tilde{\mathcal{S}}^i(\theta, \eta)$ being the segmentation result of our framework. Optimization of ϵ is rather difficult for several reasons: the segmentation process cannot be described analytically, parameter space is discrete and there are additional constraints (filter sizes have to be odd and positive). Therefore we iteratively add a new component to the MDoB filter and perform cyclic coordinate search [7] at each iteration until convergence.

5 Experiments

We applied our framework to several applications, such as standard document analysis, license plate recognition or id-card recognition. Figure 5 shows some segmentation results of our method. As derived in section 2 linear shading and hard shadow effects are removed. This is also reflected by the recognition rates for complete license plates presented in Table 1. We used our segmentation framework within a license plate recognition system and evaluated the performance using MDoB filters with our local segmentation approach and with local segmentation alone. For testing we used a complex street scene with 6205 images and multiple license plates per image. This huge dataset includes all typical challenges for character separation, such as highlights, shading effects, hard shadows and dirt. Each image was equipped with manually labeled ground truth data and was provided by our industry partner ROBOT Visual Systems GmbH. Training of our character recognition system (which is not a topic of this paper) was done using a fixed set of single character images.

For a second quantitative analysis, images of characters with arbitrary scale and rotation were generated. These images served as ground-truth data and were disturbed with different typical noise types to create input images. After segmentation the result was compared to the ground-truth images using the character separation measure introduced in equation (6). Noise in form of continuous shading and hard shadows (Figure 6 and 7) was used to provide a similar challenge as applications like e.g. document analysis and license plate recognition. The degree (noise ratio) of these two noise types is controlled by a parameter β. Increasing values of β correspond to an increasing noise ratio. Details about the generation of noise are skipped due to lack of space. For different values of β, 2900 generated images were used to estimate the mean error of the character separation measure.

The analysis in Figure 6 and 7 showed the ability of our efficient segmentation framework (70 milliseconds for an image of size 500×500) to obtain good segmentations

Figure 5. Examples for segmentation results within a license plate and id card recognition scenario: Black color corresponds to pixels marked as object pixel during our local segmentation step. Segmentation artefacts can be easily filtered in an additional post-processing.

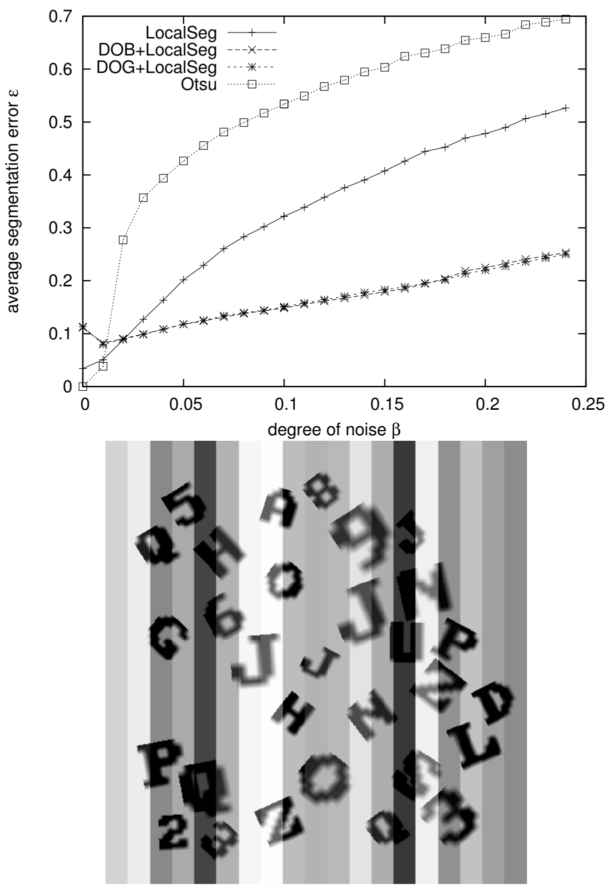

Figure 6. Evaluation of the influence of hard shadows using formula (6)**. Example image of an artifical input image for this experiment.**

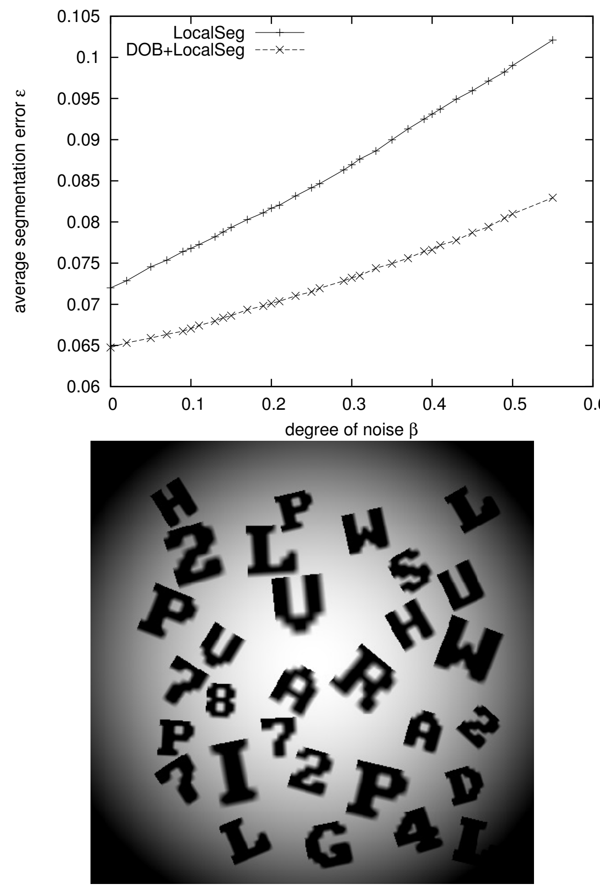

Figure 7. Evaluation of continuous shading influence.

(in the sense of character separation) which are more robust against shading effects than traditional methods or local adaptive methods without preprocessing.

Additionally, the approximation of DoG filters by using DoB filters does not have an influence on the segmentation quality as can be seen in the evaluation in Figure 6.

In some rare cases the preprocessing step leads to small gaps within character segmentation, which is illustrated in Figure 8. This problem can be solved by subsequent morphology-based operations.

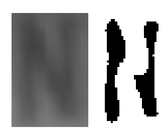

Figure 8. Typical problem with strokes, which are close to each other: (Left) original image, (Right) output of our segmentation framework

6 Conclusions

One important aim of this paper was to show that in spite of using simple low-level image operations one can obtain good segmentation results with high invariance to illumination effects. Therefore our method can serve as a good baseline method for the evaluation of more complex character segmentation methods.

We presented an efficient framework for segmentation within general document analysis systems which is also applicable to license plate recognition or id-card recognition. Our approach is divided in a preprocessing step and a final segmentation. The first step uses a combination of a small number of DoB filter outputs. These simple filters are easy to implement, well known and in combination perfectly suited to correct linear shading effects and removing shadows. Complex and slow approaches, such as the estimation of illumination effects, are thus not necessary. Especially because of its fast computation times our framework fits the needs for current applications of document analysis, such as automatic digitalizing of a huge amount of historic books in libaries.

The possibility of computation with integral images would additionally allow an easy incorporation of these filters as features within traditional object detectors [11]. This could be used to detect and localize characters for e.g. document structure analysis or sign recognition. Furthermore we presented an optimization method, which allows to find optimal parameters of our segmentation framework for a specific application.

References

[1] B. Gatos, I. Pratikakis, and S. J. Perantonis. Adaptive degraded document image binarization. *Pattern Recogn.*, 39(3):317–327, 2006.

[2] R. Gonzalez and R. Woods. *Digital Image Processing*. Addison-Wesley, 2nd edition, 2002.

[3] M. R. Gupta, N. P. Jacobson, and E. K. Garcia. OCR binarization and image pre-processing for searching historical documents. *Pattern Recogn.*, 40(2):389–397, 2007.

[4] H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistic Quarterly*, 2:83–97, 1955.

[5] R. D. Lins, A. R. G. e Silva, and G. P. e Silva. Enhancing document images acquired using portable digital cameras. In M. S. Kamel and A. C. Campilho, editors, *ICIAR*, volume 4633 of *Lecture Notes in Computer Science*, pages 1229–1241. Springer, 2007.

[6] S. J. Lu and C.-L. Tan. Automatic detection of document script and orientation. In *ICDAR '07: Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR 2007) Vol 1*, pages 237–241, Washington, DC, USA, 2007. IEEE Computer Society.

[7] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, August 1999.

[8] C. P. R. Unnikrishnan and M. Hebert. Toward objective evaluation of image segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29:929–944, 2007.

[9] A. Rosenfeld and M. Thurston. Edge and curve detection for visual scene analysis. *IEEE Transaction on Computers*, 20:562–569, 1971.

[10] J. J. Sauvola and M. Pietikinen. Adaptive document image binarization. *Pattern Recognition*, 33(2):225–236, 2000.

[11] P. Viola and M. Jones. Robust real-time object detection. *International Journal of Computer Vision*, 57(2):137–154, May 2004.

[12] V. Vonikakis, I. Andreadis, N. Papamarkos, and A. Gasteratos. Adaptive document binarization - a human vision approach. In A. Ranchordas, H. Araújo, and J. Vitrià, editors, *VISAPP (2)*, pages 104–109. INSTICC - Institute for Systems and Technologies of Information, Control and Communication, 2007.