OCR Based Thresholding

Yves Rangoni[1] Faisal Shafait[1] Thomas M. Breuel[1,2]

Image Understanding and Pattern Recognition (IUPR) Research Group
[1]German Research Center for Artificial Intelligence (DFKI) GmbH
[2]Technical University of Kaiserslautern
D-67663 Kaiserslautern, Germany
{rangoni,shafait,breuel}@dfki.uni-kl.de

Abstract

In large-scale digitization processes, several common tasks are performed to provide an electronic version of a paper document. One of the first steps is the thresholding of the image, which is necessary for the following procedures to work properly. Many binarization methods have been proposed to solve this problem, but they need to be tuned on the target document corpus to obtain best results. In this paper, we introduce a full automatic thresholding method for printed document analysis. The purpose is to obtain the most suitable binarizer for a given document image according to the quality of the output of an OCR system. Tuning can be done either on a full page or on sample text-lines extracted from a page image. As opposed to existing methods, the tuning is directly goal-directed and does neither depend on subjective visual evaluation nor on non-representative performance criteria. We demonstrate the effectiveness of this approach on a subset of 740 pages from the Google 1000 Books dataset. Results show, that by choosing the right binarizer parameters with the Recognition Driven Thresholding (RDT) method the words-in-dictionary error rate of an OCR system can be reduced by 6%.

1 Introduction

The goal of document binarization is to convert a given greyscale or colour document image into a bi-level representation. The underlying objective is to separate objects, like characters, from the background with the assumption that grey levels of pixels belonging to the two classes are substantially different.

The quality of the binarization is crucial for document recognition because most of the algorithms used during analysis (page orientation, layout analysis, character recognition, etc.) assume a black and white image and rely on the output of the binarizer.

Many approaches for binarizing greyscale or colour documents have been proposed in literature [9]. They can be broadly divided into global and local methods. Global binarization methods try to find a single threshold value for binarizing the whole page. Each pixel in the document image is assigned to page foreground or background based on its grey value. Global methods are computationally inexpensive and they give good results for office scanned documents. However, if the illumination over the document is not uniform, i.e. in the case of camera-captured documents, they fail to correctly binarize the document. Local methods try to overcome this problem by computing thresholds for each pixel individually, using information from the local neighbourhood of the pixel. They are able to achieve good results even on severely degraded documents, but they are often slow since the computation of image features from the local neighbourhood is to be done for each image pixel. Without describing in detail all the key points and drawbacks of each of them, several comparisons survey like [9, 2] suggest that Sauvola's binarization method [8] outperforms the other local thresholding techniques; whereas Otsu's method [7] works best among the global techniques.

Every binarizer depends on parameters which are influencing its performance greatly. They must be set in context of document type and target application like OCR. Correct values are not straightforward to set, especially for most of the local techniques. Usually, subjective evaluations employ humans who tune the parameters according to their perceptual impression. Manual procedures are not suitable for achieving high performance on a large range of heterogeneous documents at low cost. Some techniques have been employed [14, 2] to obtain these right parameters by optimizing a criterion, i.e. an edge detection, which should quantify how suitable the binarization was. There are at least two main drawbacks in such a technique. First, the criterion to optimize does not necessary imply that these settings will be the best for the recognizer; all the proposed methods are not goal-directed. Second, the most advanced ones need also initial settings or assumptions to work. On top of that, the overhead of extra computations is sometimes not justified compared to the gain of performance.

In this paper, we introduce a fully automatic method for finding the best parameters of a binarization technique to optimize the performance of an OCR system. Even if the experimental part is focused on Sauvola's technique, the proposed framework is still fully valid for any kind of binarization method. The next section describes in detail the steps of the *recognition driven thresholding* (RDT) method. Then, experimentations will show how it performs on a public dataset of ancient documents. Finally, conclusions and perspectives will be discussed.

2 Recognition Driven Thresholding

Recognition Driven Thresholding (RDT) is a binarization technique that applies OCR directly to the

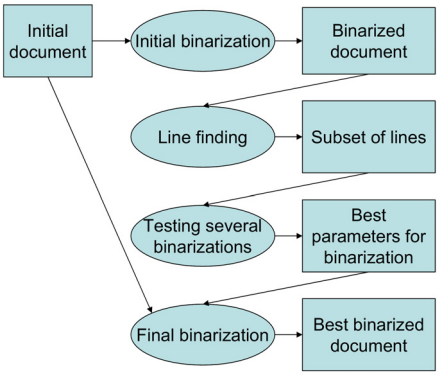

Figure 1: Recognition Driven Thresholding overview

problem of determining good binarization parameters. It means that the OCR is directly used to evaluate the quality and tune the binarization technique. The scheme is to extract some representative lines of the document, then apply different binarization algorithms (and/or different setting for one algorithm), recognize the lines with the targeted OCR, evaluate which binarizer has given the best results and then apply it on the full document (Fig. 1).

2.1 Sauvola's binarization

We make use of the locally adaptive thresholding of Sauvola [8]. It is widely used and performs the best on document binarization [9]. The threshold T for each pixel (i, j) depends on local variance of the pixel neighbourhood. The local mean $m_W(i, j)$ and standard deviation $\sigma_W(i, j)$ are computed on a window of size $W \times W$ around the pixel with a bias K.

$$T_{W,K}(i,j) = m_W(i,j) \cdot \left(1 + K\left(\frac{\sigma_W(i,j)}{128} - 1\right)\right)$$

The main drawback of the method is its need to set the correct parameter values (W, K). Furthermore, these values are not adaptive to individual documents. Sauvola et al. proposed $(15, 0.5)$ to be a good choice. However, Sezgin et al. [9] or Trier et al. [12] have found $(15, 0.2)$ to work better. Another study by Badekas et al. [1] suggested yet another value pair of $(14, 0.34)$.

Even if some settings are valid on average, they are not optimal for each image and depend on the target application like OCR. In addition, the criteria used for determining a good threshold may not guarantee good OCR results. The aim of the proposed method is to optimize the right cost function by minimizing the OCR errors.

2.2 Fast line finding and extraction

In order to evaluate several parameters of the binarizer, the quality of the recognition is tested on a few lines of the document image. In contrary to other methods, the focus is put on the lines of the text and no time is wasted for optimizing a criterion on needless parts of the document, like pictures or drawings for example. We use a fast and robust method, RAST, for extracting a subset of text lines [3]. An interesting property is its capacity of working with a targeted number of lines, and the results are returned in decreasing order of quality.

2.3 Evaluation of the binarization parameters using OCR

After obtaining the best greyscale line images, the next step is to optimize the quality of the recognition on them. The evaluation is goal-directed: we make use of the recognizer to produce the best input (W^*, K^*) for itself. The idea is to test several combinations of binarization parameters, and evaluate the accuracy of the obtained transcription with a line recognizer.

The subset of lines is binarized with different parameters (W, K), and then the OCR is applied. In this work, we calculate the ratio of existing words in the OCR output according to a dictionary D with total length of the text. The objective is to maximize this ratio for all the lines in the subset S:

$$(W^*, K^*) = \underset{W,K}{\operatorname{argmax}} \left(\sum_{line \in S} cost_{W,K}(line) \right) \text{ where}$$

$$cost_{W,K}(line) = length(line)^{-1} \sum_{\substack{word \in line \\ word \in D}} length(word)$$

Flexible matching with edit-distance can be applied to allow one or two errors for long words. Language models, based on trigrams for example, can also be good candidates and can deal with several languages in the same list of trigrams, but it requires building a model and also handling specific cases [5]. Initial experimentations suggested that the exact matching search within a dictionary gives the best results. With a binary search, computing the cost function is negligible even for a large number of words.

The optimization process is made on the space of the binarizer parameters with an exhaustive search. We assumed having no knowledge about the behaviour of (W, K), and the variables are considered as discrete.

3 Experiments and Results

Few large and publicly available datasets with greyscale images of complex documents exist. To evaluate the approach in a challenging application area, we chose a subset of the Google 1000 Books [13] which contains scans of old books. Pages from the inner sections of each English volume have been picked. The original dataset was composed of 770 images, and after removing blank pages or pages without text, the final subset contains 740 documents.

The ground truths given with the G1000 contain many errors, especially for pages where optimizing the binarization produces large and interesting differences. As a consequence, it was not possible to rely on recognition rates, or other edit-distance based methods [6]. We preferred to work with a 'words in dictionary' ratio instead, which is a more robust and general scheme for datasets with ground truths at all.

The open source project OCRopus 0.3.1 [4] has been used to run the experiments using Tesseract 2.0.3 [11] as the character recognition engine. OCRopus has a fast implementation of the Sauvola's technique using integral images to reduce the runtime [10], a tuneable line finder and extractor using RAST.

3.1 Evaluation on full pages

The first experiment is designed to test how well the proposed values in literature make Sauvola's method

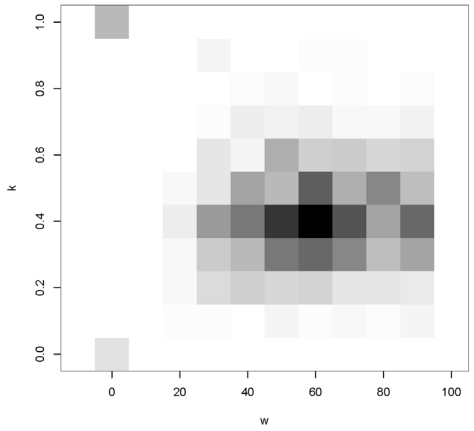

Figure 2: Overview of the winning binarizers for G1000. Each best (W, K) for a page contributes to the darkness of a square at that position. $(60, 0.4)$ seems to be the best unique value for this dataset.

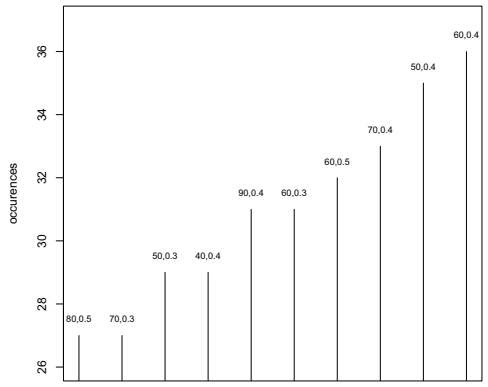

Figure 3: Occurrences in y-axis of the 10 best (W, K) in x-axis. They show higher values for W and K than the literature suggests

work on a heterogeneous corpus. The different W and K are in the ranges $[10, 90]$ and $[0.1, 0.9]$ respectively, and full pages are used for evaluating the binarization. We were also testing a binarizer by range, represented as $(W, K) = (0, 0)$ and the Otsu's binarizer at $(0, 1)$.

It can be seen (Fig. 2,3) that values suggested by different authors are far from being optimal on a new corpus of data. There is no best unique binarizer that can perform well on all kinds of documents. Contrary to what we found in the literature, $(W, K) = (15, 0.2)$ for Sauvola is really not optimal for the Google 1000 books dataset. In some cases, Otsu or the simple binarizer can outperform it. By experience, we have found that $(W, K) = (40, 0.3)$ is suitable for a large range of applications, we can consider here that the first best suitable couple is $(40, 0.4)$. Interesting values of K are in $[0.3, 0.5]$ and 0.4 seems to be a nice trade-off. As reported in the literature, Sauvola is more affected by K than W, and most of the authors set W to a small value around 15 pixels. As we are using a fast implementation of Sauvola, setting W to high values does not matter much. In the top 10, that the smallest value is 40 and higher values are working better. The best unique value for G1000 is $(60, 0.4)$, we can keep it in mind as the "oracle" couple.

The Fig. 4 shows the quality improvements obtained when choosing the right binarizer with the RDT method and a fixed binarizer with $(W, K) = (40, 0.3)$. It can be seen that the auto-threshold allows an increase of the recognition rate: the mean 'words in dictionary ratio' is 53.03% whereas with the fixed values, the ratio is 47.06%, that is to say a difference of 5.97 points. For two documents, the difference was greater than 33 points, 7 cases greater than 20 points and 114 documents with a difference greater than 10 points.

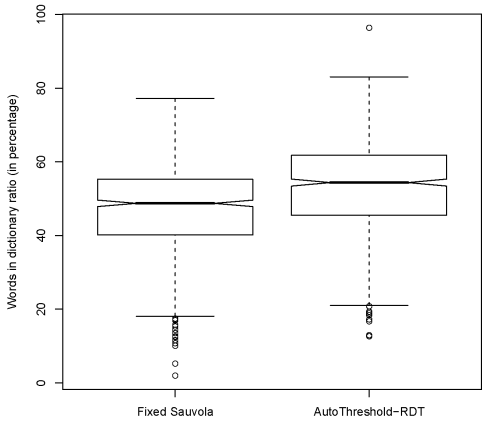

Figure 4: Quality improvement in term of words in dictionary, higher values are better. First boxplot gives scores for fixed Sauvola $(W, K) = (40, 0.3)$. Second boxplot the scores obtained with RDT on page level. RDT obtains a mean of 53% vs. 47% for fixed Sauvola

Note that with the recommended values $(15, 0.2)$, the difference of ratio is now 12.44 points instead of 5.97 and 11 documents have a ratio at least greater than 30 points with the auto-threshold on page-level.

3.2 Evaluation on a subset of lines

The size of the subset of representative lines is set to 10 now. As we know that Sauvola with $(W, K) = (60, 0.4)$ is the best fixed binarizer for our dataset (the oracle), we will compare the results with it. The experimentations show now that the best unique binarizer is $(W, K) = (50, 0.3)$. The global behaviour of making a choice with a small subset of lines tends to choose lower value for W and K (Fig. 5). With a restricted number of lines, several couples can produce more often identical scores. Lowest values are considered.

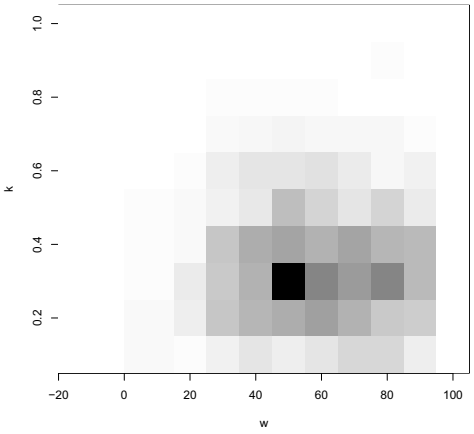

Figure 5: Overview of the winning binarizers with a subset of 10 lines. Lower values are observed due to more ex-aequos, but general behaviour is kept

But choosing other values does not imply making a wrong decision for character recognition. Indeed, even with only 10 lines, the auto-threshold is still better than an oracle (Fig. 6): we have a ratio difference of 0.21 points (47.97% vs. 47.76%). With a subset of 8 lines, there is no difference (0.02 points) and with five lines, the oracle is the winner (-0.82 points).

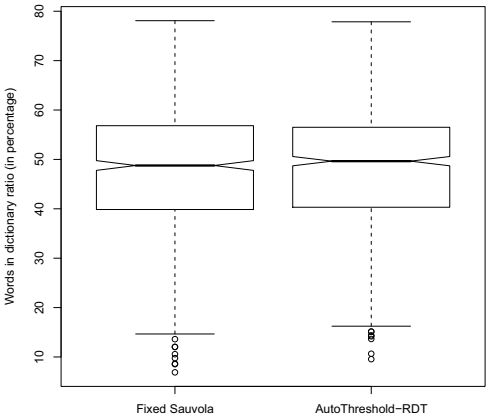

Figure 6: Quality improvement, higher values are better. First boxplot gives results with the "oracle" Sauvola (best values computed for the full page) second boxplot shows the scores obtained with RDT but with 10 lines

When an oracle is not available, RDT is really better than a fixed choice: with 5 lines, we obtained a ratio of 46.93%, 47.78% with 8 lines, and 47.97% with 10 lines, compared to the 40.58% obtained with $(W, K) = (15, 0.2)$. The average gain is 7 points. Note that the proposed values are not so weird since they produce a nicer image with less cuts and missing parts, but what is good for our eyes is not necessary good for the OCR.

3.3 Computation Time

Since the aim of the binarization is page recognition, RDT does not require implementing any new OCR or binarization modules and requires few extra computations. Indeed, 10 lines of a page represent less than 10% of its total surface, and less than 5% for a two-column layout. We are testing 81 combinations of (W, K), but with the use of integral images, lot of computations are redundant for the same W and the binarization cost depends only on W. Binarizing the 81 subsets is roughly equivalent to binarize one time the full page. Finding the lines with RAST is a step that must be done and it is not an extra cost. Only the evaluation (recognition of the text) is time consuming. A subset of 10 lines costs between 10% and 20% of the time for recognizing the full document. As the OCR is here employed to have an evaluation; no dictionary and language modelling have been set for Tesseract and the recognition time is closer to 10%. Finally the total extra cost for the RDT is one binarization and 10 simple text recognitions. Note that testing so many combinations is not necessary; better range of values and interpolations can highly reduce the amount of real evaluations.

4 Conclusions

We have presented a Recognition Driven Thresholding (RDT) method in order to improve the quality of the text recognition. In the literature, many methods have been proposed, but all of them require a specific tuning, as Sauvola's method, to work well on a given dataset. In difference to the previous approaches, that optimize the parameters on a non-text recognition oriented measure, we proposed to directly exploit the OCR engine for the evaluation of the parameter effectiveness. The method uses a small subset of representative lines composing the document, to infer the right choice on the full document.

The RDT is not restricted to Sauvola's method and can be applied to any binarization method with tunable parameters. As we have shown on the Google 1000 books corpus, the RDT gives better results than Sauvola's method with fixed parameters given by an oracle. The RTD takes advantage of highly degraded documents in a heterogeneous corpus. Significant recognition improvement can be done fairly quickly thanks to the use of integral images. A full-range search has been employed, we plan to use optimization algorithms to speed-up the parameter search.

References

[1] E. Badekas and N. Papamarkos: "Automatic Evaluation of Document Binarization Results", *Progress in pattern recognition, image analysis and applications*, vol.3773, pp.1005-1014, 2005

[2] E. Badekas and N. Papamarkos: "Estimation of proper parameter values for document binarization", *International Conference on Computer Graphics and Imaging*, no.10, track 600-037, 2008.

[3] T. M. Breuel: "Robust least square baseline finding using a branch and bound algorithm", *Proc. SPIE Document Recognition and Retrieval IX*, pp.20-27, 2002.

[4] T. M. Breuel: "The OCRopus Open Source OCR System", *Proceedings SPIE DRR XVI*, 2008.

[5] S. F. Chen and J. Goodman: "An empirical study of smoothing techniques for language modeling", *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, pp.310-318, 1996.

[6] Y. Li and D. Lopresti and G. Nagy and A. Tomkins: "Validation of Image Defect Models for Optical Character Recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp.99-108, 1996.

[7] N. Otsu: "A threshold selection method from gray level histograms", *IEEE Trans. Systems, Man and Cybernetics*, vol.9, pp.62-66, 1979.

[8] J. Sauvola and M. Pietikinen: "Adaptive document image binarization", *Pattern Recognition*, vol.33, no.2, pp.225-236, 2000.

[9] M. Sezgin and B. Sankur: "Survey over image thresholding techniques and quantitative performance evaluation", *Journal of Electronic Imaging*, vol.13, pp.146-165, 2004.

[10] F. Shafait and D. Keysers and T. M. Breuel: "Efficient implementation of local adaptive thresholding techniques using integral images", *Document Recognition and Retrieval XV*, vol.6815, 2008.

[11] R. Smith: "An Overview of the Tesseract OCR Engine", *International Conference on Document Analysis and Recognition*, vol.2, no.9, pp.629-633, 2007.

[12] O. D. Trier and A. K. Jain: "Goal-directed evaluation of binarization methods", *Pattern Analysis and Machine Intelligence*, vol.17, no.12, pp.1191-1201, 1995.

[13] L. Vincent: "Google book search: document understanding on a massive scale", *International Conference on Document Image Analysis*, pp.819-823, 2007.

[14] Y. Yitzhaky and E. Peli: "A method for objective edge detection evaluation and detector parameter selection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.25, no.8, pp.1027-1033, 2003.