Color Binarization for Complex Camera-based Images

Celine Thillou and Bernard Gosselin

TCTS Lab, Faculte Polytechnique de Mons, Avenue Copernic, 7000 Mons, Belgium

ABSTRACT

This paper describes a new automatic color thresholding based on wavelet denoising and color clustering with K-means in order to segment text information in a camera-based image. Several parameters bring different information and this paper tries to explain how to use this complementarity. It is mainly based on the discrimination between two kinds of backgrounds: clean or complex. On one hand, this separation is useful to apply a particular algorithm on each of these cases and on the other hand to decrease the computation time for clean cases for which a faster method could be considered. Finally, several experiments were done to discuss results and to conclude that the use of a discrimination between kinds of backgrounds gives better results in terms of Precision and Recall.

Keywords: Thresholding, Color Clustering, K-means, Complex Background, Classifiers

1. INTRODUCTION

Optical Character Recognition (OCR) from camera-based pictures is a quite recent research area and could be considered thanks to recent evolutions of technologies integrating digital cameras and powerful data processing in embedded devices such as personal digital assistants. Actually, new needs and the challenge to get high recognition rates appeared. For example, a mobile reading system for blind or visually impaired system or a mobile translation device for foreign visitors in any country are potential uses of OCR from camera-based pictures. We can also cite the help for car driving with an embedded imaging device to recognize and interpret text information all along the road. A kind of applications is the display of this recognized information on the windshield.

Nevertheless, camera-based images induce numerous degradations not present in scanner-based ones such as blur, uneven lighting, complex backgrounds, perspective distortion, ... Moreover, text needs to be detected because it is not placed in expected areas in the image. It can be everywhere, with different fonts and sizes. Text detection is a preliminary step of this paper and will be no more considered here. Images we process contain already detected text as shown in Figure 1.

We have no a priori information on text and research done on printed scanner-based character recognition is not robust enough for all these constraints. Therefore, commercial OCRs present very low recognition rates or even no results on this kind of images.

An image analysis system includes several image-processing tasks and the thresholding one can affect quite critically the performance of successive steps such as classification of the document into text objects, and the correctness of OCR. Improper thresholding causes blotches, streaks, erasures on the document confounding segmentation, and recognition tasks. The merges, fractures, and other deformations in the character shapes as a consequence of incorrect thresholding are the main reasons of OCR performance deterioration.

First of all, this paper will describe the state of the art of binarization techniques and background evaluation methods. Section 2 will deal with our color thresholding followed by Section 3 on several considerations about the discrimination between backgrounds. To finalize the explanation of methods, results will be given in Section 4 before concluding and describing our future work.

Further author information: E-mail: celine.thillou@tcts.fpms.ac.be, Telephone: 32 65 37 47 17

Figure 1. Samples of images we considered in this paper

2. A STATE OF THE ART

Usually, for color thresholding images, most papers convert the RGB image into a gray-level one and apply different algorithms mainly grouped into two categories: global[1] and local or adaptive.[2] Global methods tempt to binarize the image with a single threshold. By contrast, local ones change the threshold dynamically over the image according to local information. Meanwhile in our context, image processing systems need to process a large number of documents with different styles and without pre-specified parameters, which can be a failure for local methods such as the Sauvola[3] one. Those algorithms have proven their efficiency but they are not robust enough to handle any complex backgrounds.

Nevertheless, several surveys have been published on the subject to extend this first opinion on binarization techniques and to understand their complexity. According to Sezgin,[4] the thresholding methods can be categorized according to the information they are exploiting, such as histogram shape (global methods), measurement space clustering, entropy, object attributes, spatial correlation and local gray-level surfaces (local methods). The choice of a proper algorithm is mainly due to images to analyze.

For camera- or video-based images, Wolf[5] does not use color information and the binarization is based on an improved version of Niblack's method.[2] An image enhancement based on multiple frame integration is used to increase the resolution of characters, in order to get better results. But in our context, no video information is available and text cannot be enhanced so easily. Seeger[6] created a new thresholding technique for camera images, like in our context, by computing a surface of background intensities and by performing adaptive thresholding for simple backgrounds. For very complex color images, those methods are not sufficient and color information could be used to get more clues. In order for people to read text in complex color images, color information is more significant than the contrast between gray-level values.

Wang[7] tried to combine both color and texture information to improve results. This technique works well for images similar to our database but computation time required is very high. There is no consideration on connectivity between components and results are given under visual judgement. With other techniques, and some similar ones, our method fills these failures. Garcia[8] uses a character enhancement based on several frames of video information and a K-means clustering as our method to binarize text information. His method creates four clusters and combination of clusters to get as much text as possible is done on a bunch of criteria concerning characters properties. For him also, text areas are already detected. On the contrary, he does not take into account stroke analysis and results are worse for character segmentation. Moreover he obtained best non-quantified results with hue-saturation-value (HSV) color space. Our results based on the public database (Samples of words of Robust Reading Competition ICDAR 2003[9]) disagree about this last statement as explained in Thillou.[10]

The last algorithm we want to discuss in this section is the Du's one.[11] Color images are composed of three channels (red, green, blue) and entropy-based thresholding are applied on each gray-level channel. Based on a between-class/within-class variance criterion, the three subimages are merged to constitute a binary image. Results seem attractive but text areas are not already detected. As in our case, text information is the main one in the image, this algorithm does not give the same results.

Actually, discrimination between two or more kinds of backgrounds is a not very investigated problem. Some papers try with thresholding algorithms to segment text from background as described in the previous paragraphs. Some other ones tempt to get rough thresholded image and to describe the quality of the image afterwards. Then, therefore, different restoration models can be applied as in Cannon's paper.[12] Five quality measures are defined: Small Speckle Factor, White Speckle Factor, Touching Character Factor, Broken Character Factor and Font Size Factor. According to the results, different restoration algorithms are applied as in Souza[13] with five other criteria. Kanungo[14] defined the quality of images not according to images but to characters themselves.

Figure 2. Applying Otsu (middle) or Sauvola (right) thresholding on original images (left) is a major decision at this point: top: the background is complex, Sauvola is the best algorithm, bottom: the background is clean, Otsu is the best algorithm

Nevertheless, all these restoration models could be studied in a following step because thresholding is already performed. Our goal is to discriminate backgrounds before applying any binarization methods in order to compute the right one and to save computation time.

3. OUR BINARIZATION APPROACH

As explained in previous sections, we use color information in our binarization approach but it is only used after gray-scale denoising and coarse thresholding in order to consider only useful parts and to decrease the required time for color clustering with less pixels. Then a combination of results is either applied or not, according to a parameter of distance and this eventual combination is partial or total in order to take into account non-connectivity of characters.

3.1. Coarse Pre-Processing

An important problem for thresholding methods and especially for "real-world" pictures comes from a non-uniform illumination which introduces noise. This uneven illumination appears as wide noisy areas and is assumed to have a lower frequency spectrum than the one of characters. Based on a wavelet decomposition described in Thillou,[15] the denoising is done by adding no more degradation.

The well-known Otsu[16] method or the Sauvola[3] one is then applied directly on the reconstructed image. The local Sauvola method is used with a fixed window-size of 15. We decide to apply one of both techniques according to the kind of backgrounds, complex or not. They are both considered thanks to their complementarity and their answers to different kinds of backgrounds are shown in Figure 2. Based on samples of words of the public database ICDAR 2003,[9] we tested the complementarity of different direct thresholding algorithms to decide that it is better to use both Otsu and Sauvola. Otsu performs well on 76% images, Sauvola on 48% but 56% of images are different for both algorithms. Several experiments have been done with Otsu and different local thresholding methods such as Niblack,[2] Kittler[17] and so on.

By applying then a zonal mask on the color image as shown in Figure 3 on the left, this step is useful to remove useless parts of the image and to decrease computation time of color clustering. Actually in order to consider only color parts of this first thresholding, the mask is applied by an AND operation on each R,G,B subimage. This task is done in order to constraint the size of the region-of-interest.

3.2. Color clustering

In Wang,[7] color clustering is done using Graph Theoretical clustering without giving the number of clusters because the picture was not pre-processed. Actually, pre-processing with the approximate thresholding does not lose any textual information in our database.

We use the unsupervised K-means clustering with K=3. The three dominant colors are extracted based on the color map of the picture. The color map is obtained by getting all intensity values of each pixel and by removing duplicate values. Color map bins are hierarchically merged according to an Euclidean metric in the color space. These merged bins form color clusters that are iteratively updated by the K-means algorithm using the Euclidean distance metric. Finally, each pixel in the image receives the value of the mean color vector of the cluster it has been assigned to. The robustness of K-means about variability of initial seeds is reached by computing several times the algorithm and by choosing the one with the minimum clustering error. Three clusters are enough for our database thanks to our pre-processing with the mask. A decomposition is shown in Figure 3 on the right.
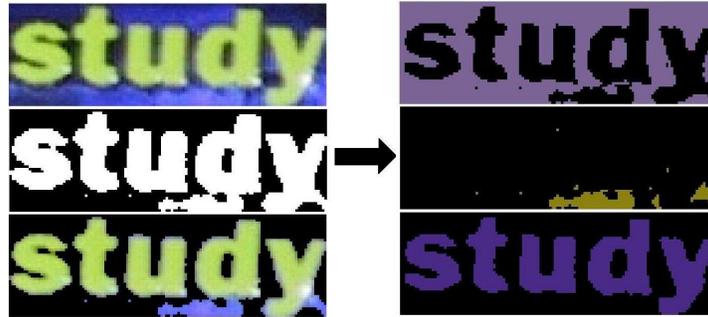
Figure 3. Left: A zonal mask applied on initial color images, right: color clustering in three subimages (background, noise and foreground)

Figure 4. Two different foreground clusters: left: one foreground image and noise, right: two foreground images

3.3. Eventual partial or total combination of clusters

The background color is selected very easily and efficiently (100 % in the ICDAR 2003 database) as being the color with the biggest rate of occurrences in the image edges. Therefore only two pictures are left which correspond, depending on the initial image, to either two foreground pictures or one foreground picture and one noise picture as shown in Figure 4.

In Wang,[7] combination is based on some texture features to remove inconvenient pictures and on a linear discriminant analysis with other criteria. Here, the most probable useful picture is defined with a means of skeletization. Actually, as the first thresholding corresponds in an approximative way to characters, a skeletization is used to get the color of centers of characters as in Thillou.[15]

An Euclidean distance D with both mean color pixel of the cluster and mean color of the skeleton is performed. The cluster with the smallest distance from the skeleton is considered as the cluster with the main textual information.

The combination choice is done according to the distance D between mean color values of the two remaining clusters. If distance is inferior to a certain threshold, color are considered as similar and the second picture seems to be a foreground picture too. Connected components on the first foreground picture are computed to get coordinates of their bounding box in order not to connect components with pixels to add in the combination. Only pixels which can be added will change the first foreground picture. On the contrary, some characters can be broken if they were broken in the first foreground picture. But, in this case, the correction will be facilitated by the fact that characters parts will be closer.

4. DISCRIMINATION BETWEEN COMPLEX OR CLEAN BACKGROUNDS

In Subsection 3.1, we saw that it is better to combine answers from Otsu and Sauvola thresholding during the process of color thresholding. But how to combine them? Different ideas can be tested.

The first one is to apply both of these methods on each document and to combine them or to choose one of them during the recognition step. It induces difficulties in merging data. How to know if a character is right or not or which one to keep? In another way, keeping the most textual result can be a solution but no save on computation time can be done.

Another solution is to apply both of these methods and to remove the result which contains less text or no text, directly after thresholding. Different algorithms used also for validation of text areas in text detection field could be used. Lienhart[18]

uses connected components properties such as mean size, mean inter-distance or spatial variance. Zheng[19] in another way discriminates noise from text with different features and classifiers such as Neural Networks (NN) or Support Vector Machines (SVM). We are currently doing tests on that but nevertheless, both methods have still to be computed and we think that we can save more computation time by doing the discrimination before the color thresholding step.

4.1. Our discrimination approach

The discrimination between clean or complex backgrounds can help character segmentation and recognition if a better algorithm is applied for each image. Moreover, as compared in Thillou,[10] for very clean images as the rightmost subimage in Figure 1, only a denoising step and a global thresholding will be sufficient and could decrease the computation time. We first focused on this paper on the discrimination between two classes in order to apply the proper algorithm, which is between the Otsu or Sauvola thresholding during the coarse pre-processing in our color thresholding.

Our approach is classification-based with tests with different classifiers and different set of features. These experiments were done to understand which feature or classifier works the best but future work could be done for example to optimize sets of features with several types of feature selection. Three sets of features were tried to know if a background is clean or not:

- Full gray-level histogram containing 256 features.

- Smoothed full gray-level histogram in order to be insensible to all slight variations in the histogram. This set contains also 256 features.

- Three values for this third set: the spread width, the maximum value and the number of peaks of the histogram.

We tested these three sets on 5 different classifiers: Linear Discriminant Analysis Classifier (LDA), Quadratic Discriminant Analysis Classifier (QDA), K-Nearest Neighbors (KNN), Full-Connected Multi-Layer Perceptron (MLP) and Support Vector Machines (SVM). All the results are described in Section 5. Best results are get with the third set of features and the SVM classifier.

5. DISCUSSION AND EXPERIMENTAL RESULTS

All compared classifiers are supervised classifiers. Our training database is the TrialTrain of Robust Reading Competition of ICDAR 2003[9] and our test database is the Samples of the Competition.

For LDA and QDA, we use no particular parameters. An Euclidean distance metric is used and a priori probabilities P_i for each of both classes (clean or complex) are fixed to 0.5. For KNN, we did several tests to find the proper K and it is fixed to 2. As for LDA and QDA, we choose the Euclidean distance metric for the algorithm. For MLP, we use a back-propagation neural network with a training based on the hold-out method. The training set is divided into a number of samples for training and another one with fixed size for validation. It is useful to avoid overtraining. In order to get better results for SVM, we did several experiments to tune required parameters. To construct an optimal hyperplane, SVM employs an iterative training algorithm, which is used to minimize an error function. We used C-SVM classification and a radial-basis function (RBF) for the kernel K defined by:

$$K(x, y) = \exp(-\gamma \|x - y\|^2)$$

We use cross-validation to design γ and C for each set of features. To realize all these tests, we used libSVM,[20] available online.

Table 1 describes classification rates for each set of features and each classifier. An error is counted when it is damageable for character segmentation and recognition. Sometimes, both methods (Otsu or Sauvola inside color clustering) give comparable results and for those cases, no error is counted. An example is shown in Figure 5.

Best results are get for the third set of features with SVM classifier. But, results can still be improved with a larger training database and features more optimized by selection. The main goal was to prove the usefulness of the discrimination of background to merge data of different binarization algorithms.

Table 1. Classification rates for each set of features and each classifier

	LDA	QDA	KNN	MLP	SVM
256 features	53%	33%	42%	63%	62%
256 smoothed features	55%	42%	43%	78%	74%
3 features	67%	69%	72%	79%	82%
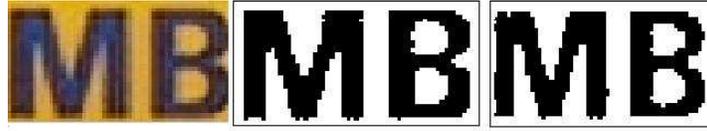
Figure 5. Image (left) where Otsu (middle) and Sauvola (right) gives the same result. No error is counted for classification of backgrounds

Then, we compared the advantage of the discrimination between backgrounds with a measure of Precision and Recall,[21] defined on characters as:

$$\text{Precision} = \frac{\text{Correctly Detected Characters}}{\text{Totally Detected Characters}}$$

$$\text{Recall} = \frac{\text{Correctly Detected Characters}}{\text{Totally Characters}}$$

Inside the algorithm described in Section 3, we compared in Table 2, Otsu thresholding alone, Sauvola one alone and both methods.

Table 2. Comparison of several algorithms or combination of them inside our color thresholding (CT)

	Precision	Recall
Otsu+CT	0.770	0.560
Sauvola+CT	0.714	0.443
Otsu and Sauvola + CT with discrimination	0.811	0.59

As the usefulness of a discrimination step between clean or complex backgrounds was proven, a scheme of our whole color thresholding is proposed in Figure 6.

6. CONCLUSION AND FUTURE WORK

This paper described an automatic color thresholding algorithm divided into several steps: a denoising part, the discrimination of background, the restriction of region-of-interest by applying a global or local rough thresholding algorithm and a zonal mask on the initial image followed by a color clustering. This step performed by K-means is not the first step in order to save computation time and to be more accurate. Three clusters are obtained after this color thresholding and once the background is removed, the two remaining clusters are merged together according to the proximity of their color. If a merge is applied, it can be partial or total with respect to non-connectivity between connected components to improve character segmentation.

A particular focus is done on the discrimination step between two kinds of backgrounds, clean or complex. The usefulness of this discrimination was shown with tests with different set of features and different classifiers. Results can be still improved but they already increased thanks to this particular step.
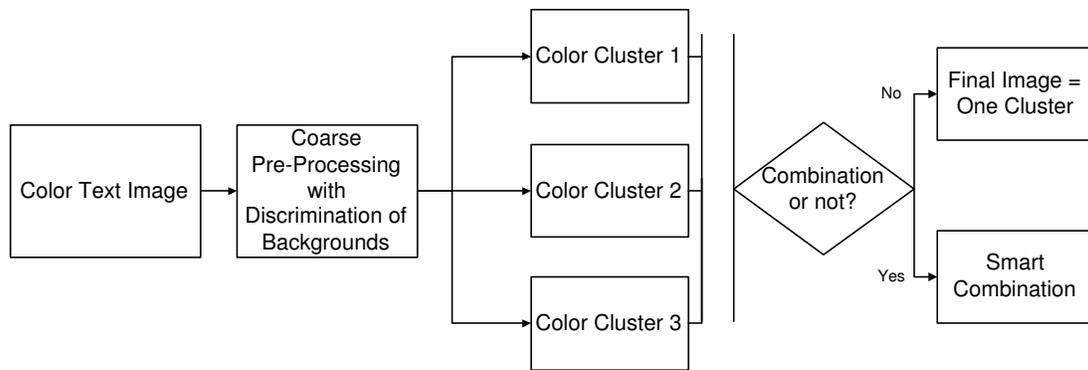
Figure 6. An overview of our whole color thresholding method

Other tests are currently in progress with different distance metrics in K-means for color clustering. Some new results appeared and a new complementarity between two distances exists too. Future works will be to manage all these complementarities with data fusion algorithms to extract text as well as possible. Actually, all information we can get in the thresholding step can improve drastically character segmentation and recognition.

ACKNOWLEDGMENTS

REFERENCES

1. G.Leedham, C.Yan, K.Takru, J.Tan, and L.Mian, "Comparison of some thresholding algorithms for text/background segmentation in difficult document images," *Proceedings of ICDAR*, 2003.
2. W.Niblack in *An introduction to digital image processing*, E. Cliffs, ed., pp. 115–116, Prentice Hall, 1986.
3. J.Sauvola and M.Pietikinen, "Adaptive document image binarization," *Pattern Recognition* **33**, pp. 225–236, 2000.
4. M.Sezgin and B.Sankur, "Survey over image thresholding techniques and quantitative performance evaluation," *Journal of Electronic Imaging* **13**, pp. 146–165, 2004.
5. C.Wolf, J-M.Jolion, and F.Chassaing, "Text localization, enhancement and binarization in multimedia documents," *Proceedings of ICPR* **2**, pp. 1037–1040, 2002.
6. M.Seeger and C.Dance, "Binarising camera images for ocr," *Proceedings of ICDAR*, pp. 54–59, 2001.
7. B.Wang, X-F.Li, F.Liu, and F-Q.Hu, "Color text image binarization based on binary texture analysis," *Proceedings of ICASSP*, pp. 585–588, 2004.
8. C.Garcia and X.Apostolidis, "Text detection and segmentation in complex color images," *Proceedings of ICASSP* **4**, pp. 2326–2330, 2000.
9. "Robust reading competition database." http://algoval.essex.ac.uk/icdar/RobustWord.html, 2004.
10. C.Thillou and B.Gosselin, "Combination of binarization and character segmentation using color information," *Proceedings of ISSPIT*, 2004.
11. Y.Du, C.Chang, and P.Thouin, "Unsupervised approach to color video thresholding," *Proceedings of SPIE Optical Imaging* **43**, pp. 282–289, 2004.
12. M.Cannon, J.Hochberg, and P.Kelly, "Quality assessment and restoration of typewritten document images," *International Journal Document Analysis Recognition* **2**, pp. 80–89, 1999.
13. A.Souza, M.Cheriet, S.Naoi, and C.Y.Suen, "Automatic filter selection using image quality assessment," *Proceedings of ICDAR* **1**, pp. 508–512, 2003.
14. T.Kanungo and Q.Zheng, "Estimating degradation model parameters using neighbourhood pattern distributions: an optimization approach," *IEEE Trans. on Pattern Analysis and Machine Intelligence* **26**, pp. 520–524, 2004.
15. C.Thillou and B.Gosselin, "Robust thresholding based on wavelets and thinning algorithms for degraded camera images," *Proceedings of ACIVS*, 2004.

16. N.Otsu, "A thresholding selection method from gray-level histogram," *IEEE Trans. on Systems, Man, and Cybernetics* **9**, pp. 62–66.

17. J.Kittler and J.Illingworth, "Threshold selection based on a simple image statistic," *Computer Vision Graphics and Image Processing* **30**, pp. 125–147, 1985.

18. R.Lienhart and W.Effelsberg, "Automatic text segmentation and text recognition for video indexing," *Multimedia systems* **1**, pp. 69–81, 2000.

19. Y.Zheng, H.Li, and D.Doermann, "Machine printed text and handwriting identification in noisy document images," *IEEE Trans. on Pattern Analysis and Machine Intelligence* **26**, pp. 337–353, 2004.

20. "libsvm website." http://www.csie.ntu.edu.tw/ cjlin/libsvm/, 2004.

21. M.Junker and R.Hoch, "On the evaluation of document analysis components by recall, precision, and accuracy," *Proceedings of ICDAR*, pp. 713–716, 1999.

22. "Sypole website." http://tcts.fpms.ac.be/projects/sypole/sypole.html, 2004.