

PROHIST: An Environment for Image Processing of Historical Documents

Carlos A.B.Mello
Department of Computing Systems,
University of Pernambuco
Recife, PE, Brazil
+55 81 2119-3842
carlos@dsc.upe.br

Adriano L.I.Oliveira
Department of Computing Systems,
University of Pernambuco
Recife, PE, Brazil
+55 81 2119-3842
adriano@dsc.upe.br

Ángel Sanchez
Superior School of Experimental
Science and Technology, Rey Juan
Carlos University, Madrid, Spain
+34 91 664-7452
angel.sanchez@urjc.es

ABSTRACT

There is an increasing interest in historical documents nowadays. Due to the fragility of the paper as main media of storing information through the ages, it is necessary to develop means to retrieve this information in a more robust way. At the same time, it must be possible to make this information available to everyone world around. Digital media comes as the more effective way to achieve both these objectives. They are more appropriate for preservation purposes and digital information is easily accessible through the Internet. This paper describes the advances in creating algorithms for image processing of historical documents achieved in the ProHist project. These advances come from thresholding, text segmentation and automatic indexing of the documents. Each one of these steps works together as part of a complete environment for publishing historical documents in a digital library.

Categories and Subject Descriptors

I.7.4 [Document and Text Processing]: Electronic Publishing

General Terms

Algorithms, Experimentation.

Keywords

Document processing, Historical documents, thresholding, segmentation, handwritten digit recognition.

1. INTRODUCTION

The advances in multimedia databases and in the Internet broadcasting allow a bigger number of people to access larger repositories through the world. When these databases are comprised of documents, the interests come in two levels: the ones which are interested in the contents of the documents and the ones interested in the visualization of the files. The information contained in a document can be accessed by visualization of its image or by accessing a textual representation of it. In both cases, there are several issues that must be observed. In spite of the

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

WDL'07, October 21–24, 2007, Gamado, Brazil.

Copyright 2007 ACM 1-58113-000-0/00/0004...\$5.00.

common use of broadband Internet access nowadays, the visualization of thousand of files is not a simple task. Even in JPEG file format the archive consumes Giga Bytes of space.

In order to access the textual information, OCR tools can be used to translate textual images into text files. Nonetheless, unfortunately, this is not an easy solution when the document is handwritten. Automatic character recognition of handwritten documents is still a challenge.

This paper reports the research currently being developed in the PROHIST Project [1] for preservation and broadcasting of historical documents. This work presents the results applied to the letters and documents from a file which is composed of almost 6,500 documents from the end of the 19th century onwards, totalizing more than 30,000 pages. Documents are digitized in true color with 200 dpi resolution and stored in JPEG file format with 1% loss for preservation purposes. Even in this format each image of a document reaches, in average, 400 KBytes.

Herein, we describe aspects of image thresholding, segmentation and handwritten digit recognition and the new developments done in the scope of historical documents. New algorithms and results for image binarization, text line segmentation and handwritten digit recognition are presented. At this point, our interest is in an automatic indexation of the complete archive.

This research is being developed jointly by University of Pernambuco, Brazil, and Rey Juan Carlos University, Spain. More information can be found in references [6]-[9].

Next Section presents the new thresholding algorithm developed in the project. Next, in Section 3, document segmentation is described as it is used in our project. Section 4 presents the studies in digit recognition and the last Section concludes the paper.

2. IMAGE THRESHOLDING

Thresholding [10] is an important part of several image processing or pattern recognition applications. In our case, we search for means to classify the foreground (the ink) or the background (the paper). After this, the pixels classified as ink are turned to black and the pixels classified as paper are turned to white. The main problem comes when the images have low contrast. This means that there exists a gray level interval of pixel intensities where it is hard to determine whether a given pixel belongs to the foreground or the background region. To obtain a perfect separation, an appropriate threshold value must be defined. The gray levels below this value are classified as ink and

the gray levels above are part of the paper. The threshold value is considered correct if all the essential information of the image is preserved.

Several image features can be used for thresholding. Entropy is one of them and it has a major importance in our work.

Entropy [12] is a measure of information content. In Information Theory, it is assumed that there are n possible symbols, s , which occur with probability $p(s)$. The entropy associated with the source S of symbols is:

$$H(S) = -\sum_{i=0}^n p[s_i] \log(p[s_i]) \quad (1)$$

where the entropy can be measured in bits/symbols.

Five entropy-based segmentation algorithms are well known in literature: Pun, Kapur *et al*, Li-Lee, Wu-Lu and Renyi.

Pun's algorithm analyses the entropy of black pixels, Hb , and the entropy of the white pixels, Hw , bounded by the threshold t :

$$Hb = -\sum_{i=0}^t p[s_i] \log(p[s_i]) \quad (2)$$

$$Hw = -\sum_{i=t+1}^{255} p[s_i] \log(p[s_i])$$

The algorithm suggests that t is such that maximizes the function $H = Hb + Hw$. Some aspects of Pun's algorithm are further used in our proposal.

Tsallis entropy [13] has been considered a new information measure. According to Tsallis, an universal definition of entropy is given by:

$$H_{\alpha}(S) = \frac{1 - \sum_i p(i)^{\alpha}}{\alpha - 1} \quad (3)$$

where $p(i)$ is a probability as in the classical definition of entropy and α is a real parameter.

Tsallis entropy (Eq. 3) can be broken into two parts as follows:

$$H_{\alpha}(S) = H_{b\alpha}(A) + H_{w\alpha}(B)$$

where

$$H_{b\alpha}(A) = \frac{X_b}{\alpha - 1} - \frac{1}{\alpha - 1} \sum_{i=0}^t p(i)^{\alpha}$$

and

$$H_{w\alpha}(B) = \frac{X_w}{\alpha - 1} - \frac{1}{\alpha - 1} \sum_{i=t+1}^{255} p(i)^{\alpha}$$

with $X_b + X_w = 1$. In our application, t is the most frequent color in the image and it works as an initial threshold. As most part of a document belongs to the paper, it can be considered that this most frequent color is part of the background. $H_{b\alpha}$ is the measure of the pixels below the color t and $H_{w\alpha}$ is the measure of the colors above the threshold t . The variable t is also used to define the values of X_b and X_w , as X_b is the percentage of colors below t and X_w is the percentage of colors above t .

Although it is not fixed by Tsallis, we considered after several experiments α equal to 0.3 for our application.

At first, the documents are classified in one of three groups according to the value of Shannon entropy (H) but with the logarithmic basis taken as the product of the dimensions of the

image. As defined in [3], changes in the logarithmic basis do not alter the definition of the entropy.

- Class 1 ($H \leq 0.26$): documents with few parts of text or documents where the ink has faded;
- Class 2 ($0.26 < H < 0.30$): common documents with around 10% of text elements;
- Class 3 ($H \geq 0.30$): documents with more black elements than it should have; this includes documents with a black border or documents with back-to-front interference.

These boundaries between classes were defined analyzing a set of 500 images representatives of the complete archive.

Also, the values of Hb and Hw (Eq. 2) are evaluated, using the most frequent color (t) as the separation point.

For each of these classes, an analysis must be made to process the images that belong to them as can be seen next. The final threshold value, th , is defined by:

$$th = mb * H_{b\alpha} + mw * H_{w\alpha}$$

where mb and mw are multiplicative constants that are defined for each class as we show next.

For class 1 documents, we have:

- If ($Hw \geq 0.1$), then $mb=2.5$ and $mw=4.5$ (typewritten documents with dark ink and bright paper);
- If ($0.08 < Hw < 0.1$), then $mb=6$ and $\alpha=0.35$ (documents with the ink faded);
- If ($Hw \leq 0.08$), then $mb=4$ (documents with dark ink and paper).

The algorithm defines $mb = 2.2$ and $mw = 3$ for the documents of class 2. Some darkened documents need another treatment. If a document belongs to class 2 and $Hw > 0.1$, then the value of mw decreases by half (i.e., $mw = 1.5$), unless the most frequent color is greater than 200 (brighten documents) for which $mw = 9$.

For the last class, in general, mb is fixed as 1 and $mw = 2$. Some cases, however, must be considered when the documents have brightened paper again. In this class, brighten paper documents are the ones with most frequent color (t) greater than 185:

- If ($t \geq 185$) then
 - If ($0.071 < hw < 0.096$) then $mw = 9$;
 - If ($0.096 \leq hw < 0.2$) then $mw = 6$.

Figure 1 presents the results of the binarization of sample documents from each class.

To evaluate quantitatively the results found, we analyzed the values of precision, recall, accuracy and specificity defined in [5] by:

- Precision (P) = TP/(TP + FP)
- Recall (R) = TP/(TP + FN)
- Accuracy (A) = (TP + TN)/(TP + TN + FP + FN)
- Specificity (S) = TN/(FP + TN)

Where TP stands for true positive; FP is false positive; TN is true negative, and FN is false negative. The comparison is made using a gold standard image created manually. For each document, a perfect bi-level image (the ideal image) was created by a visual choice of the best thresholding value.

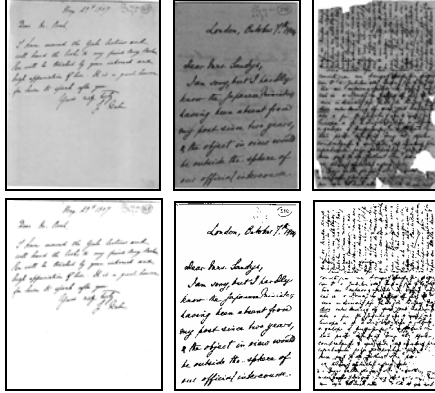


Figure 1. (top) Sample documents of each class (class 1: left, class 2: center and class 3: right) and (bottom) their binarization by the new algorithm

A good algorithm must have all these measures tending to 1 at the same time. Table 1 presents the average result for these four measures applied to a set of 200 documents binarized by the new proposed algorithm and classical algorithms [11] in comparison with their ideal images. As one can see, the algorithm achieved the higher values for the four measures.

Table 1. Average values of precision, recall, accuracy and specificity in a set of 200 bi-level documents generated by the new proposal and classical methods compared with their ideal version generated manually

Algorithm	P	R	A	S
New Algorithm	0.82	0.88	0.97	0.98
Brink	0.91	0.69	0.95	0.98
C-Means	0.88	0.79	0.93	0.99
Fisher	0.95	0.51	0.73	0.99
Huang	0.88	0.80	0.94	0.99
Kapur	0.88	0.79	0.93	0.98
Kittler	0.94	0.73	0.96	0.99
Mean Grey Level	0.95	0.71	0.96	0.99
Otsu	0.81	0.81	0.97	0.98
Pun	0.94	0.69	0.93	0.99
Renyi	0.88	0.77	0.93	0.99
Two Peaks	0.87	0.82	0.95	0.98
Wu-Lu	0.94	0.71	0.95	0.99
Yager	0.99	0.17	0.39	0.91
Ye-Deniilsson	0.87	0.77	0.93	0.99

3. TEXT LINE SEGMENTATION AND WORD EXTRACTION

Line segmentation is an important stage in the OCR of handwritten texts. In the particular application of historical documents, the complexity increases due to additional problems related to the degradation of the original documents. Moreover, the variability of handwriting makes it difficult to extract and separate the lines using a general method. In particular, the text lines are not equally spaced, they get skewed, different lines touch each other and artefacts due to scanning noise appear in the digital document.

Our approach to detect and separate lines of text in a document image I (where I is the bi-level image generated by our algorithm) consists in the following steps:

1. Eliminate document margins of I obtaining the image I' .

2. Compute and smooth the black/white transition count map M from I' .
3. Search and separate touching lines obtaining a set of line regions L from M .
4. Extract and simplify the “axis” a_i of each line l_i in L through a skeletonization algorithm.
5. Assign a unique line number to each of the pixels in the written parts of the image document I' .

Next, we shortly describe each of the involved algorithm stages for automatic text line segmentation.

- **Margin elimination:** We estimate that a digitized ink component is part of a margin (and it is not considered in the image document I') if its height is similar to the image height and it is a vertical line (for the case of vertical margins) or its width is similar to the image width and it is a horizontal line (for the horizontal margins).
- **Transition count map M computation:** This map is a more compact representation of the pre-processed image I' . The value of each pixel in M is set to the number of transitions (from white to black or vice versa) in the binary image of the document which appear in a horizontal window centered at each pixel location. The result is a “blotted” binary image where dark regions are highly probable to contain the text lines. For the following stages, this image is smoothed using a median filter. To compute M , we have implemented a modified version of the approach proposed in [3].
- **Extraction of the set of line regions L :** Due to the interpersonal characteristics of handwriting (*i.e.* size and shape of the letters, spaces between lines, slanted writing, etc), it is difficult to formalize the concept of text line. We use the previous smoothed transition map M to define a set L of line regions obtained by separating the possible touching characters of different lines in the original document. For this goal, we analyze the shape of the line regions to determine the candidate “cutting points” between lines used to split the touching lines and determine the set of separated line regions.
- **Compute and simplify the set of line axis A :** We use the line axis as a mechanism to determine the correct position of each line. This axis also gives the global shape and orientation of each line (text lines are not always exactly parallel to each other). This is efficiently performed by computing the skeleton of each separated line region using Zhang and Suen’s algorithm [16]. Next, each line skeleton is improved by joining incorrect broken axis of the same text line, removing branching and holes in line axis, etc. At this stage, false line axis corresponding to noise are also detected and removed.
- **Assign a line number to each text pixel in the image I' :** The information provided by the set A of line axis is carefully superposed or combined with the connected components of the text image. We search exactly which line axis a_i in A “cuts” each text component. These intersecting text pixels are used as seeds for a region growing algorithm that labels the connected pixels as belonging to the same text line. Several problems can appear at this step (*i.e.* a text component could cut to more than one line axis) and they are solved by specific heuristics.

Figure 2 illustrates the proposed text line segmentation method.

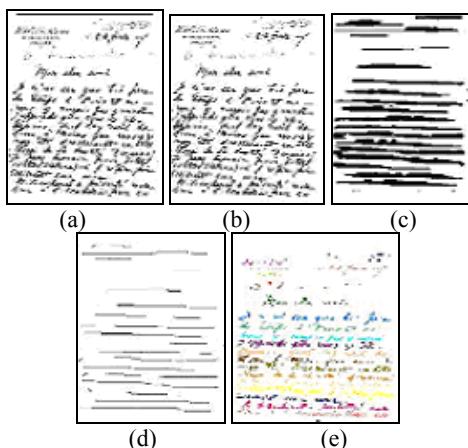


Figure 2. (a) Scanned thresholded image, (b) margin removal, (c) image with line regions, (d) line axis extraction, and (e) segmented text lines (each different colour represent a line).

Once the text lines have been segmented, the component words corresponding to each line are extracted and enclosed into bounding rectangles. These rectangles can be later used by a classifier to recognize the manuscript text in the document. A word image is composed of discrete characters, connected characters, or a combination of the two [4]. The goal is to merge these subunits into a unique meaningful unit which is a word. The proposed word extraction method using the segmented lines is based on the idea that two connected components which form a word are less spaced than two different words in a line. Our method has two main steps:

1. Apply appropriate mathematical morphology dilation to the separated components of each line to achieve these components touch. Then, these new overlapped subunits are labelled as belonging to the same word.
2. Associate all the supposed punctuation marks (i.e. commas, points, etc) to the corresponding words. Using heuristics based on distance and position, we detect the corresponding bounding boxes of these elements that are later merged.

The proposed approach was tested for a database of 15 different images corresponding to thresholded handwritten scanned document images from the complete archive. Average percentages of 82% and 89% were obtained for correct line segmentation and correct word extraction, respectively.

4. HANDWRITTEN OPTICAL DIGIT RECOGNITION

At this stage, our purpose is handwritten digit recognition in order to have automatic indexing. The documents can be easily indexed if a system can sort them automatically by date. We made experiments on digits recognition employing two different feature extraction methods, namely, (a) principal components (PCA) and (b) undersampled bitmaps. The classifiers considered here are (1) k-nearest-neighbors (kNN), (2) radial basis functions neural networks (RBFNs) and (3) support vector machines (SVMs). All these methods and classifiers are detailed in [14].

After document segmentation according to the previous Section, we have gathered 200 handwritten digits from our file of historical documents to form our dataset. The extraction of the

digits themselves is made manually. The automation of this step is object of further phases of the project.

This digit dataset is balanced; that is, it is composed of 20 images of each digit. The area of each number is approximately of the same size although some areas must intersect each other.

The digits were normalized to bitmaps of 32 by 32 pixels. Next, a feature extraction method, either PCA or undersampled bitmaps was applied. In the case of PCA, we have retained the number of eigenvectors which accounted for 95% of the variance of the original data. This has produced input vectors with 130 components. For undersampled bitmaps, our input vectors had 64 components. After the application of these feature extraction methods we have normalized the inputs from 0 to 1. The simulations were carried out using Weka tool [15].

10-fold cross-validation (CV) was used to assess the generalization performance as well as to compare the feature extraction and classifiers considered in this work. In CV, a given dataset is divided into ten subsets. A classifier is trained using a subset formed by joining nine of these subsets and it is tested by using the one left aside. This is done ten times; each one employing a different subset as the test set and computing the test set error, E_i . Finally, the cross-validation error is computed as the mean over the errors E_i , $1 \leq i \leq 10$. It is important to emphasize that all the simulations reported here used stratified CV, whereby the subsets are formed by using the same frequency distribution of patterns of the original dataset [15].

Our first set of experiments considered kNN as the classifier. The results obtained in these experiments are shown in Table 2. Notice that the parameter k (the number of neighbors to be considered for classification) has a noticeable influence in the performance. More importantly, for all values of k the use of undersampled bitmaps as the feature extraction method remarkably outperforms the use of PCA. The same observations can be made in the case of the simulations using RBFNs, whose results are also shown in Table 2.

Table 2. Performance of the kNN (as a function of parameter k) and RBF neural network (as a function of the number of hidden RBFs classifiers) and of the feature extraction method. The table reports the stratified 10-fold cross-validation errors.

k	kNN		RBF		
	PCA	Under-sampled bitmaps	RBF Units	PCA	Under-sampled bitmaps
1	78%	9%	2	36.5%	17%
3	87.5%	11%	10	50%	18%
5	88%	13%	20	31.5%	14%
7	90%	15%	50	31.5%	14%
9	89.5%	18.5%	70	31.5%	14%

Our experiments using SVMs employed a grid search considering both the complexity (C) and the kernel parameter. The experiments used both RBF and the polynomial kernel. Our grid search utilized a large number of combinations of values of these parameters. In this subsection we report on results obtained using some of these parameters, particularly those which achieved the best results. Tables 3 shows the results obtained using RBF and polynomial kernels. In both cases, the undersampled bitmaps

method yields much better accuracy than PCA. There is a strong influence of the values of the parameters on the accuracy.

Table 3. Performance of the SVM classifier with RBF kernel as a function of the complexity parameter (C), the kernel parameter (γ or the exponent d) and of the feature extraction method. It reports the stratified 10-fold CV errors.

C	γ	PCA	Under-sampled bitmaps	d	PCA	Under-sampled bitmaps
1	0.1	25.5%	7.5%	1	24%	6%
1	0.01	46.5%	24.5%	3	24%	17%
1	0.001	46.5%	54%	9	32.5%	41.5%
10	0.1	20%	5.5%	1	24%	6%
10	0.01	21.5%	7.5%	3	24%	16.5%
10	0.001	48.5%	23.5%	9	32.5%	41%
100	0.1	20%	5.5%	1	24%	6%
100	0.01	22.5%	5.5%	3	24%	16.5%
100	0	21%	7.5%	9	32.5%	41%

Table 4 summarizes the results by showing, for each classifier, the results obtained by the best set of parameters. These results clearly indicate that: (1) the undersampled bitmaps feature extraction method produces much better results than PCA in combination with any of the classifiers considered, (2) SVM remarkably outperforms both kNN and RBFN in our problem, and (3) the kernel function used by SVM (RBF or polynomial – labeled as *pol. kernel* in Table 5) had insignificant influence on performance.

Table 4. Comparison of classification schemes. This table shows the best results obtained by each classifier.

Classifier	PCA	Undersampled bitmaps
kNN	78%	9%
RBFN	31.5%	14%
SVM (RBF kernel)	20%	5.5%
SVM (pol. kernel)	24%	6%

5. CONCLUSIONS

This paper presents the ProHist project for creating methods for preservation and publishing images of historical documents. We related the advances we have achieved in several aspects of image processing of these type of documents.

Specifically, we presented a new algorithm for image thresholding which is the main step for a system for automatic character recognition. Our new algorithm is based on Tsallis entropy and it was compared with several well-known classical thresholding algorithms. The comparison was made using the values of precision, recall, accuracy and specificity.

After binarization, the bi-level images are segmented for text detection. The document is segmented into its text areas which is a very difficult task when we deal with handwritten documents.

The text areas containing digit elements are dealt in the recognition phase for automatic indexing of the documents. Currently, the digits are extracted manually. We proceeded with the analysis of the best recognition method. Our simulations have shown that the best results were obtained by using undersampled bitmaps as our feature extraction method. This method obtained much better accuracy than PCA when used with any of the

classifiers considered here. The best combination was with SVMs, which yielded 5.5% 10-fold cross-validation error.

6. ACKNOWLEDGEMENTS

This research has been partially supported by CNPq, UPE, Universidad Rey Juan Carlos and Agencia Española de Cooperación Internacional (AECI) under contract no. A/2948/05.

7. REFERENCES

- [1] PROHIST: <http://recpad.dsc.upe.br/prohist>
- [2] Kapur, J.N.: *Measures of Information and their Applications*, J.Wiley & Sons, 1994.
- [3] Kennard, D.J. and Barrett, W.A.: *Separating Lines of Text in Free-Form Handwritten Historical Documents*, Proc. Second International Conference on Document Image Analysis for Libraries, France, 2006.
- [4] Manmatha, R. and Rothfeder, J.L., *A Scale Space Approach for Automatically Segmenting Words from Historical Handwritten Documents*, IEEE TPAMI, v. 27, no. 8, 2005.
- [5] McMillan, N.A. and Creelman, C.D.: *Detection Theory*. LEA Publishing, 2005.
- [6] Mello, C.A.B.: *New Tsallis Entropy-Based Thresholding Algorithm for Images of Historical Documents*. ACM Document Engineering, Winipeg, Canada, 2007.
- [7] Mello, C.A.B. et al.: *Image Thresholding of Historical Documents: Application to the Joaquim Nabuco's File*, Digital Cultural Heritage Conference - Eva Vienna, p. 115-122, Austria, 2006.
- [8] Mello, C.A.B., and Costa, A.H.M.: *Image Thresholding of Historical Documents Using Entropy and ROC Curves*, Lec. Notes in Computer Science, v. 3773, p. 905-916, 2005.
- [9] Oliveira, A.L.I., et al.: *Optical Digit Recognition for Images of Handwritten Historical Documents*, Brazilian Symposium of Neural Networks, p.29, Brazil, 2006.
- [10] Parker, J.R.: *Algorithms for Image Processing and Computer Vision*. John Wiley & Sons, 1997.
- [11] Sezgin, M. and Sankur, B.: *Survey over image thresholding techniques and quantitative performance evaluation*, J. of Electronic Imaging, no.13, v.1, pp. 146-165, 2004.
- [12] Shannon, C.: *A Mathematical Theory of Communication*, Bell System Tec.Journal, v.27, pp. 370-423, 623-656, 1948.
- [13] Tsallis, C.: *Possible Generalization of Boltzmann-Gibbs statistics*, J. Stat.Physics, v.52, nos. 1-2, pp. 479-487, 1988.
- [14] Webb, A. *Statistical Pattern Recognition*. John Wiley & Sons, second edition, 2002.
- [15] Witten, I.H. and Frank, E.: *Data Mining: Practical Machine Learning Tools and Techniques*, second ed., Morgan Kaufmann, San Francisco, 2005.
- [16] Zhang, T.Y., and Suen, C.Y.: *A Fast Parallel Algorithm for Thinning Digital Patterns*, Communications of the ACM, v. 27, no. 3, 1984.