ELSEVIER

Pattern Recognition Letters 21 (2000) 483–491

Pattern Recognition Letters

www.elsevier.nl/locate/patrec

Context-based filtering of document images

E. Ageenko *, P. Fränti

Department of Computer Science, University of Joensuu, Box 111, FIN-80101 Joensuu, Finland

Received 24 March 1999; received in revised form 5 January 2000

Abstract

Two statistical context-based filters are introduced for the enhancement of binary document images for compression and recognition. The simple context filter unconditionally changes uncommon pixels in low information contexts, whereas the gain–loss filter (GLF) changes the pixels conditionally depending on whether the gain in compression outweighs the loss of information. The filtering methods alleviate the loss in compression performance caused by digitization noise while preserving the image quality measured as the optical character recognition (OCR) accuracy. The GLF reaches approximately the compression limit estimated by the compression of the noiseless digital original. © 2000 Elsevier Science B.V. All rights reserved.

Keywords: Document images; Enhancement; Compression; Filtering; Noise removal; Context-based statistical modeling; Optical character recognition

1. Introduction

The legal requirements and preservation standards oblige to store documents in exact visual forms as they were produced (Harvey, 1993). *Document imaging* refers to the management of paper documents by digitizing, archiving, retrieving, and distributing them electronically (Saffady, 1993; Lynn-George, 1996). The whole process may be set for fully automatic operation without human intervention. Standard solutions for the *digitization* phase exist in the form of scanners and facsimile. The *archiving* phase includes *image enhancement*, *compression*, *optical character recognition* (*OCR*) and *indexing* operations.

The quality of document images may have been degraded during the document life cycle and digitization process. Noise appears on images as additive signal – a set of randomly scattered noise pixels, and content-dependent signal, making the contours of printed objects ragged. Even though the noise level can be low enough and cause only unnoticeable distortion in the image, it may degrade image compression and the accuracy of OCR. *Image enhancement* therefore aims at improving (or at least not defecting) the visual image quality and accuracy of further OCR conversion and to provide better compression performance.

Traditional filtering methods process the image by analyzing local pixel neighborhood defined by a filtering template (Bernstein, 1987; Schonfeld and Goutsias, 1991; Zhang and Danskin, 1996). The filters use a set of predefined rules, or quantitative description of the local neighborhood as in *morphological filters* (Serra, 1982; Heijmans, 1994;

* Corresponding author. Tel.: +358-13-251-5271; fax: +358-13-251-3290.

E-mail address: ageenko@cs.joensuu.fi (E. Ageenko).

0167-8655/00/$ - see front matter © 2000 Elsevier Science B.V. All rights reserved.
PII: S0167-8655(00)00011-8

Koskinen and Astola, 1994; Dougherty and Astola, 1997). *Median filter* inverts the pixel if the majority of the pixel colors in the neighborhood is of the opposite color; and its *soft* generalization, self-dual *rank* operator, inverts a pixel if the number of the opposite color pixels exceeds a predefined threshold value.

These filters are appropriate for additive noise removal (e.g., radar images), but are of limited usage for document images and completely ignore the compression aim. The filtering methods are optimized for the image quality measures such as *mean absolute error* (see e.g., Schonfeld and Goutsias, 1991) and therefore result in smoothed filtered images, which are characterized by the loss of details that can be crucial for image recognition. Better filtering methods can be designed by analyzing statistical dependencies in the image. The goals of image restoration and compression enhancement should be taken into account in the design of the filters.

Document images are stored in compressed raster form in order to reduce the storage size. Using the latest binary image compression standard (JBIG, 1993), images can be compressed approximately to the same size as taken by the file formats of common word processing software. JBIG uses context-based modeling and arithmetic coding. The image is processed in raster scan order and the combination of already coded neighboring pixels defines the context. In each context, the probability distribution of the black and white pixels is adaptively determined, and the pixel is then coded by arithmetic coder.

After the document is stored, its further processing depends on the application. The document may be converted to word-processing compatible format using OCR or become indexed. *Document indexing* stands for the categorizing of the documents by some criteria and usually requires OCR of the whole document or its predefined parts. Searchable ASCII text is usually stored together with the document image to fulfill the needs of the text search on the actual content (Willis, 1992).

Our goal is to design a filtering algorithm, which being applied to document images will improve compression performance and preserve the text readability by human eyes as well as by automatic OCR systems. Two context-based filters meeting these criteria are presented in this paper. Both filters reduce irregularities in the image statistics caused by noise and in this way, improve the compression without degrading the image quality and OCR accuracy.

2. Context-based filtering

Document image can be considered as a message that is generated by some information source. Quantitative measure of information carried by a single image pixel is defined by the *entropy*

$$H(x) = -\log_2 p(x), \tag{1}$$

where x is the symbol and $p(x)$ its probability. The higher the pixel probability, the lower is its entropy. Deterministic source carries no information and may be totally replaced by its model. Entropy of the entire image is calculated as the average entropy of all pixels. It estimates the minimum *bit rate* – that is, the smallest number of bits required to code a single pixel on average, in respect to the model.

The pixels form geometrical structures with appropriate spatial dependencies. The dependencies can be localized to limited neighborhood and described by a *context-based statistical model*. In the model, the pixel probability is conditioned on the context and is calculated by counting the number of black (n_{B}^{C}) and white (n_{W}^{C}) pixels appearing in the context C in the entire image.

A context is defined as a distinct black–white configuration of the pixels in the local context template (see Fig. 1). The entropy $H(C)$ of a con-

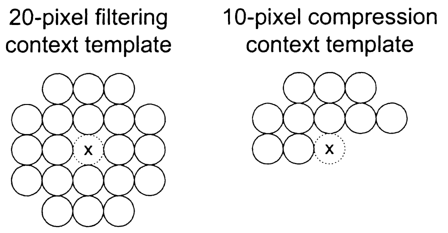
20-pixel filtering context template
10-pixel compression context template
x
x

Fig. 1. Twenty-pixel context template used for filtering and 10-pixel three-line compression context template used in JBIG compression. Location of the current pixel is marked by "x".

text C is defined as the average entropy of all pixels within the context

$$H(C) = -p_{\mathrm{W}}^{C} \cdot \log_2 p_{\mathrm{W}}^{C} - p_{\mathrm{B}}^{C} \cdot \log_2 p_{\mathrm{B}}^{C}, \tag{2}$$

where p_{W}^{C} and p_{B}^{C} are the corresponding probabilities of the white and black pixels in the given context, calculated as

$$p_{\mathrm{W}}^{C} = \frac{n_{\mathrm{W}}^{C}}{n_{\mathrm{W}}^{C} + n_{\mathrm{B}}^{C}}, \quad p_{\mathrm{B}}^{C} = \frac{n_{\mathrm{B}}^{C}}{n_{\mathrm{W}}^{C} + n_{\mathrm{B}}^{C}}. \tag{3}$$

A context with a skew probability distribution has smaller entropy and therefore smaller information content. A single pixel with high entropy value, on the other hand, either carries high information content, or is noise.

Context modeling is commonly used in image compression. The primary aims of filtering and compression, however, are not the same. This implies differences in the choice of the context template and in the collection of statistics. In compression, only preceding pixels known both by the coder and decoder can be utilized in the context template. For filtering purposes, however, directionally limited context template would not be accurate enough. As there are not any limits for referring pixels in all directions, a symmetric filtering template can therefore be applied. The round shape of the template ensures even filtering in all directions.

We have chosen the 20-pixel filtering template, as it is a well-balanced trade-off between filtering performance and reliability of the statistics. Using a larger context template we could utilize spatial dependencies from a wider area and in this way construct an even better statistical model. The number of contexts, on the other hand, increases exponentially with the number of pixels in the template. Therefore, further extension in the context template could lead to less accurate statistics as there would be less samples per context. The huge memory requirement is also a limitation for using very large templates.

The second difference between filtering and compression is that the compression is usually performed by a single pass over the image and the statistics are adaptively determined during the compression. In the filtering, on-line adaptation to the statistics would make the result unreliable until the model adapts to the image. Instead, a two-pass scheme is applied to achieve uniform image filtering: one pass for collecting the statistics and another one for the actual filtering.

2.1. Simple context filter (CF)

A simple context filter determines the statistical content of the image using context-based modeling and inverts the pixels with low probability values using the assumption that they are noise (Dougherty and Astola, 1997). The filtering process consists of two phases.

In the *analyzing phase*, context modeling with 20-pixel filtering template (see Fig. 1) is applied for the input image and the number of black (n_{B}^{C}) and white (n_{W}^{C}) pixels are calculated for each context C. After the analysis, the contexts are categorized as *low information contexts* if the probability of either black or white pixel (p_{B}^{C} or p_{W}^{C}) does not exceed a predefined *threshold* value (e.g., 0.05). The less probable pixels in the low information contexts are classified as *rare* and the most probable as *common* pixels.

The output image is generated in the *filtering phase* when all rare pixels in low information contexts are inverted. The *threshold* value is a trade-off between compression improvement and image degradation caused by filtering. The simple context filter is illustrated in Fig. 2.

2.2. Gain–loss filter (GLF)

In the previous filtering scheme, all pixels having the same context are processed in the same way, although resulting compression improvement may vary from pixel to pixel. Because of the difference in the context templates used for filtering and for compression, it is possible that pixel change does not always result in compression improvement. On the contrary, it may enlarge the code size because of the difference in context templates used in image compression and filtering.

To alleviate this problem, we propose a new GLF. Instead of using a simple probability threshold as in simple context filter, GLF takes into account the possible compression gain (*Gain*)

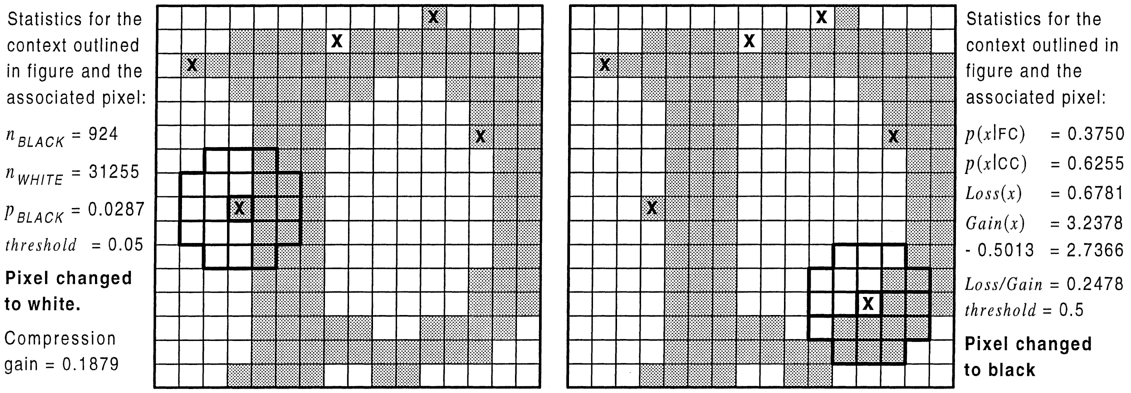

Fig. 2. Examples of the simple context filter (left) and gain–loss filter (right). Original image content is shown in gray and changed pixels are marked with "x".

as well as the error (*Loss*) delivered by changing the pixel color during the filtering (see Fig. 2).

Gain is calculated as the difference in bit-rates delivered by compressing the pixel before (x) and after ($\bar{x}$) its change

$$Gain(x) = H(x|CC) - H\left(\bar{x}|CC\right) = -\log_2 p(x|CC) + \log_2(1 - p(x|CC)), \tag{4}$$

where H is the pixel entropy value estimating the pixel bit-rate, CC is the context obtained using compression context template, see Fig. 1 and the pixel probability is estimated as

$$p(x|CC) = \begin{cases} p_{\mathrm{W}}^{CC} = \frac{n_{\mathrm{W}}^{CC}+\delta}{n_{\mathrm{W}}^{CC}+n_{\mathrm{B}}^{CC}+2\delta} & \text{if } x = \text{white}, \\ p_{\mathrm{B}}^{CC} = 1 - p_{\mathrm{W}}^{CC} & \text{if } x = \text{black}, \end{cases} \tag{5}$$

where $\delta = 0.45$ as in JBIG, and n_{B}^{CC}, n_{W}^{CC} are the numbers of black and white pixels in the image within the context CC. Note that the probability is calculated on the basis of the statistics collected over the whole image, not the statistics at the moment, as in adaptive image compression. Thus, the learning cost effect and possible improvement in compression caused by local adaptation do not affect the filtering.

It is possible, that the color change of the pixel x improves its compression but results in worse prediction and hence worse compression of some neighboring pixels y, whose local context template (denoted as CC_y) includes x. The change of the color of pixel x will result in the change of the context of pixel y. We consider this fact by deriving a more accurate equation for the *Gain*

$$Gain(x) = H(x\ |CC) - H\left(\bar{x}|CC\right) + \sum_{y|x\in CC_y} \left[H\left(y|CC_y(x)\right) - H\left(y\middle|CC_y\left(\bar{x}\right)\right)\right], \tag{6}$$

where y are the pixels whose contexts are affected by changing the color of x and $CC_y(x)$ and $CC_y(\bar{x})$ are the contexts for the pixel y before and after x is inverted, respectively.

Loss is defined by the amount of information that was lost when the pixel x was changed. It is calculated as the entropy of the inverted pixel $\bar{x}$

$$Loss(x) = H(\bar{x}|FC) = -\log_2(1 - p(x|FC)), \tag{7}$$

where FC is the context obtained using the filtering context template, see Fig. 1 and the pixel probability is estimated either as p_{W}^{FC} or p_{B}^{FC} regarding the color of x

$$p(x|FC) = \begin{cases} p_{\mathrm{W}}^{FC} = n_{\mathrm{W}}^{FC}/(n_{\mathrm{W}}^{FC} + n_{\mathrm{B}}^{FC}) & \text{if } x \text{ is white}, \\ p_{\mathrm{B}}^{FC} = 1 - p_{\mathrm{W}}^{FC} & \text{if } x \text{ is black}. \end{cases} \tag{8}$$

Thus, inversion of the pixel to the less probable would result in higher loss (values greater than 1) and vice versa. The changing of one pixel to another with equal probability (0.5) will deliver a unit of loss.

The decision whether a pixel color should be changed is due to the following criterion:

$$\frac{Loss(x)}{Gain(x)} < \text{Threshold}. \tag{9}$$

The GLF requires also two passes: analyzing and filtering. In the analyzing, the image statistics (n_{B}^{CC}, n_{W}^{CC}, n_{B}^{FC}, n_{W}^{FC}) for both context templates are calculated. After the statistics are collected, the *Loss* values are calculated for each context. The output image is generated in the second phase, where for each pixel x, the contexts *CC* and *FC* are obtained, the gain and the loss are calculated and the filtering criterion (9) is checked. If the threshold condition is met, the pixel is inverted and the statistics (n_{B}^{CC}, n_{W}^{CC}) for the compression contexts that are affected by this inversion are updated. This update is done because the coder will process the already filtered image and these updated statistics will be the statistics when the actual coding will be performed.

3. Experimental results

3.1. Test set

The proposed filtering methods were evaluated by applying them to a set of document images. We have used two test sets in our evaluation. The first set consists of artificially generated images and has been used for estimation of the threshold parameters of the filtering methods. The following operations were made to build this test set.

A text document (A4 format) was generated in a word processing system and printed with high-resolution laser printer. The document contains the same text parts formatted in two different fonts: *Times* and *Arial*, both in two different sizes: 10 and 12 points. The typefaces were chosen so that they represent the two commonly used families of fonts (with and without "serif" elements). The total number of symbols in the document is 4712 including spaces.

The printed document was subject to further transformations such as photocopying and faxing. We include four different photocopies of the document: high quality (HQ) copy (sophisticated copy machine with optimal copying parameter setup), bright and dark second copies and low quality (LQ) copy made using all-in-one desktop office machine. An ink-jet printing was also included in the set.

The resulting seven documents were then digitized with an office desktop scanner at the resolution of 300 dpi (dot-per-inch); these images are referred further as *document images*. The fax image was electronically received at the resolution of 200 dpi. Samples from the images are shown in Fig. 5.

The second, a larger scale test set consists of real documents. The documents are taken from conference proceedings and they contain text with a variety of fonts and offprint quality. The documents were digitized at resolutions of 300 and 400 dpi resulting in 56 images of the size 2328 × 3028 and 3112 × 4038 pixels, respectively. These images are referred further as *real document images*. This test set has been used for the final evaluation of the filters.

3.2. Objectives in the evaluation

The two filtering schemes (CF and GLF) were applied to the test images using different *threshold* parameters. The resulting filtered images were evaluated (see filtering samples on Fig. 3). The objectives were to determine the compression performance of the filtered images and the OCR accuracy (*recognition error*) of the images after filtering.

To measure the compression performance we applied the sequential JBIG with default (three-line 10-pixel) context template. We measure the improvement in the compression, i.e., the difference

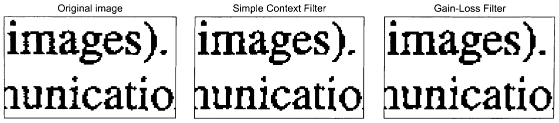

Fig. 3. Example of the filtering with simple context filter with threshold = 0.25 and gain–loss filter with threshold = 0.5.

in the file size before and after filtering. Upper limit for the compression improvement is estimated by the compression of the original noise-free raster image, which has been generated from the *PostScript* file of the original document (from the first test set). The corresponding compressed file size is 58 773 bytes for the noiseless 300-dpi document image and 36 763 bytes for the 200-dpi (fax) image. The maximum compression improvement is therefore estimated as 21.5% for 300-dpi images and 22.4% for 200-dpi fax image.

The Caere OmniPage 5.0 LE OCR software was applied for recognizing the textual content of the digitized images. This software offers a good OCR performance (97.2%) and is widely available (Haskins, 1998). The resulting textual files were compared with the original text and recognition error is measured as the *edit distance* under the unit cost model – i.e., the minimal number of edit operations: symbol changes, insertions and deletions, required for transforming a given text to the original (Sankoff and Kruskal, 1983). All spaces between the words and paragraphs were counted as one *space* symbol.

3.3. Filtering performance

The effect of the threshold parameter on the compression and recognition of the document images is illustrated in Fig. 4. For 300-dpi images,

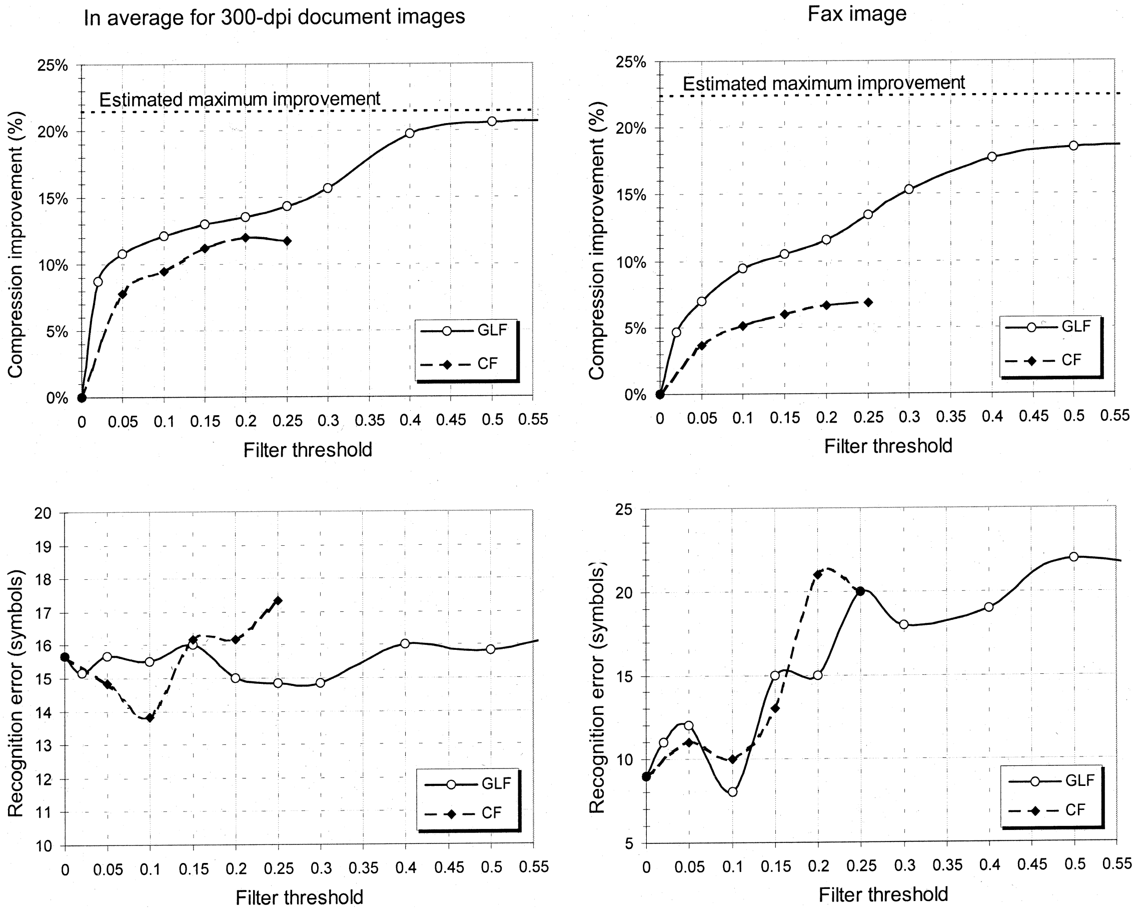

Fig. 4. Compression improvement and recognition error rates of the gain–loss and simple context filters as a function of the threshold. Compression improvement is reported as the difference in the compressed file size before and after filtering.

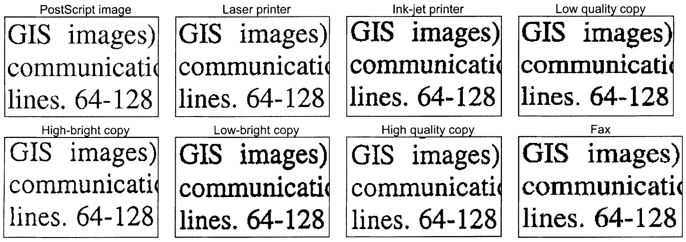
PostScript image
Laser printer
Ink-jet printer
Low quality copy
High-bright copy
Low-bright copy
High quality copy
Fax
GIS images)
communicati
lines. 64-128

Fig. 5. A sample of 12-point text from the test images digitized at 300 dpi (except the fax image, which is at 200 dpi).

the threshold value for GLF can be set to 0.5 without any noticeable effect on the recognition error rate. At this point, the method has almost achieved the estimated maximal improvement (21.5%). CF, on the other hand, starts to decrease the OCR accuracy much sooner and it never reaches the maximal compression improvement. The best result remains around 10% with threshold value 0.1.

For the fax image, CF always weakens the recognition accuracy and the compression improvement remains rather small (below 7%). GLF achieves improvement of about 10% (with threshold value 0.1) without defecting the recognition accuracy. Filtering with higher-threshold values would achieve further compression but it starts to defect the recognition accuracy significantly. The text typed in *Times* at 10 points and digitized at 200 dpi makes the letters too small and coarse to be recognized, and to form distinct statistical description necessary for the filtering.

The compression and the recognition error rates for the original and filtered document images of the first test set are summarized in Table 1. The column "PostScript" stands for the results for the clean, noiseless image that is generated directly from the PostScript file. The typical recognition errors are illustrated in Table 2. They originate mostly from two distinct symbols connected to each other (e.g., Th is mis-recognized as either A or Ml, mn is often mixed up with nm, cl with d, and so on), or due to character similarity (l, /, and i, for example). To sum up, no new serious errors were found with one exception – digit 8 is interpreted as 9 in both original and filtered versions of one document.

The evaluation results of the proposed filtering methods for the second test (real document images) are summarized in Table 3. We have applied the threshold values optimized for the test set 1 (0.1 for CF and 0.3 for GLF) to ensure maximal compression improvement with minimal affect on the recognition error. The experiment shows approximately the same rate of compression improvement as was obtained with the generated images.

3.4. Traditional non-statistical filtering methods

In comparison, non-statistical filters (such as median and morphological filters) are based on the shape, or quantitative analysis of a local neighborhood using a set of predefined rules. They result in much smaller compression improvement and produce more recognition errors. Among

Table 1
Compression performance of the JBIG and recognition errors for the non-filtered and filtered images (using GLF with threshold 0.5)

Image	Measured value	Post-Script	Laser print	Ink print	HQ copy	LQ copy	Dark copy	Bright copy	Fax
Original	Recognition error	6	4	7	9	13	28	33	9
	Compressed size	58 773	68 545	83 832	74 396	72 454	77 825	72 131	47 360
Filtered	Recognition error	5	4	9	12	7	24	39	8
	Compressed size	49 531	54 511	64 537	59 374	58 980	61 809	57 455	42 904

Table 2
Illustration of the recognition errors for two digitized images for the gain–loss filter with threshold = 0.5[a]

Font, size	Low-quality copy		Ink printing	
	Before filter (13)	After filter (7)	Before filter (7)	After filter (9)
Times, 12	unco**m**pressed unco**rn**pressed		and/or and**l**or	and/or and**l**or
	and/or and**l**or	and/or and**l**or		
Times, 10	unco**m**pressed unco**rn**pressed			ava**i**lable ava**f**lable
	2**8** 800 2**9** 800	2**8** 800 2**9** 800	**Th**e ‘**M**e	**Th**e **Al**e
	satel**l**ite satel**f**ite		and/or and**l**or	and/or and**l**or
	and/or and**l**or	and/or and**l**or		
	. ,			
Arial, 10	and/or and**l**or	and/or and**l**or	and/or and**l**or	and/or and**l**or
	“on-the-f**ly**” “on-the-fi**∅**”	“on-the-f**ly**” “on-the-fi**∅**”	“on-the-fly” ‘on-the-fly’	“on-the-fly” ‘on-the-fly’
				lines **f**ines

[a] The total number of errors in the document is shown in parenthesis. Mismatched symbols are typed in bold, and unrecognized symbols are marked as “∅”.

Table 3
Effect of the filtering on the compression and recognition of the real document images (CF is used with threshold 0.1 and GLF with threshold 0.3)

Document resolution	300 dpi			400 dpi		
Filtering method	None	CF	GLF	None	CF	GLF
Total compressed size (Kb)	1974.1	1801.3	1649.9	2678.4	2486.8	2210.2
Average OCR error (symbols)	27.7	29.6	28.4	33.5	33.9	33
Compression improvement (%)		8.8	16.4		7.2	17.5
OCR error increase (%)		6.8	2.4		1.4	−1.3

several different filters, best results for the 300-dpi document images were achieved with *median filter* (10% improvement, 19 errors) and with *rank operator* (11%, 28 errors). For the fax image, only median filter was able to keep the number of errors at a tolerable level (25), delivering 8.8% improvement in compression.

4. Conclusions

Two filtering methods based on context-based statistical modeling of the image were proposed for enhancement of the document images for compression and recognition. The context-based filtering does not depend on the semantic interpretation of the image and is solely based on the statistical properties. Both methods remove the digitization noise from the images and alleviate the loss in the compression performance caused by noise while preserving the image quality and OCR accuracy.

Acknowledgements

The work of Pasi Fränti was supported by a grant from the Academy of Finland.

References

Bernstein, R., 1987. Adaptive nonlinear filters for simultaneous removal of different kinds of noise in images, In: Proc. IEEE Transaction on Circuits and Systems CAS-34(11), pp. 1275–1291.

Dougherty, E.R., Astola, J. (Eds.), 1997. Nonlinear Filters for Image Processing, SPIE Optical Engineering Press.

Harvey, R. (Ed.), 1993. Preservation in Libraries: A Reader. Bowker-Saur, London.

Haskins, D., 1998. Word for Word. PC Magazine, January 20.

Heijmans, H.J.A.M., 1994. Morphological Image Operators. Academic Press, Boston.

JBIG: Progressive Bi-level Image Compression, 1993. ISO/IEC International Standard 11544, ITU Recommendation T. 82.

Koskinen, L., Astola, J., 1994. Soft morphological filters: A robust morphological filtering method. J. Electronic Imaging 3, 60–70.

Lynn-George, J., 1996. Digitization: A literature review and summary of technical processes, applications and issues. Jonh A. Weir Memorial Law Library, University of Alberta, Edmonton, Alberta, Canada. http://www.library.ualberta.ca/library_html/libraries/law/pubs.html.

Saffady, W., 1993. Electronic Document Imaging Systems: Design, Evaluation, and Implementation. Meckler, Westport, Conn.

Sankoff, D., Kruskal, J.B. (Eds.), 1983. Time Warps, String Edits, and Macromolecules: the Theory and Practice of Sequence Comparison. Addyson-Wesley, Reading, MA.

Schonfeld, D., Goutsias, J., 1991. Optimal morphological pattern restoration from noisy binary images, In: Proc. IEEE Trans. Pattern Anal. Machine Intell. Vol. 13, pp. 14–29.

Serra, J., 1982. Image Analysis and Mathematical Morphology. Academic Press, London.

Willis, D., 1992. Imaging: the information access tool of the nineties. In: Proc. 13th National Online Meeting. Medford, NJ, pp. 435–444.

Zhang, Q., Danskin, J.M., 1996. Bitmap reconstruction for document image compression. In: Proc: SPIE Multimedia Storage and Archiving Systems, Vol. 2916, Boston, MA, pp. 188–199.