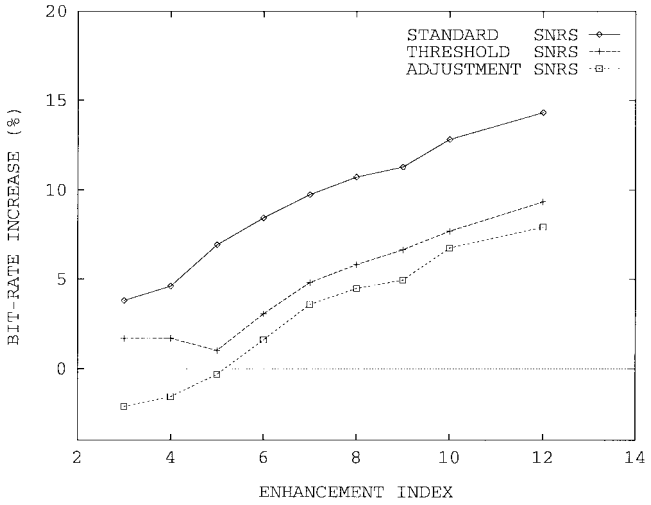

Fig. 4. Percentage increase of SNRS bit rate over equivalent single-layer versus q_e, with index ratio $q_b/q_e = 2.4$.

TABLE II
MEASURED SINGLE-LAYER AND CORRESPONDING SNRS ENHANCEMENT LAYER PSNR LEVELS FOR INDEX RATIO $q_b/q_e = 2.4$

q_e	q_b	Enhancement Layer PSNR [dB]			Single-Layer
		Standard	Threshold	Adjustment	PSNR [dB]
3	7	40.927	41.134	**41.163**	41.063
4	10	39.329	39.483	**39.484**	39.442
5	12	38.252	38.378	**38.377**	38.365
6	14	37.244	37.369	37.379	**37.389**
7	17	36.513	36.671	**36.684**	36.585
8	19	35.765	35.888	**35.914**	**35.914**
9	22	35.160	35.359	**35.350**	35.269
10	24	34.679	34.820	**34.842**	34.788
12	29	33.728	33.954	**33.967**	33.911

enhancement layer. Applying the Lagrangian multiplier technique two approaches to SNRS optimization are considered: one in which enhancement layer coefficients are manipulated and another in which the residual quantization error is optimized via the manipulation of the base layer coefficients. Of these, the direct adjustment of enhancement layer coefficients, is shown to deliver the best results outperforming both standard and optimally thresholded SNRS coders and demonstrating comparable performance to the single-layer coder.

REFERENCES

[1] M. Ghanbari and V. Seferidis, "Efficient H.261-based two-layer video codecs for ATM networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, pp. 171–175, Apr. 1995.
[2] K. Ramchandran and M. Vetterli, "Rate-distortion optimal fast thresholding with complete JPEG/MPEG decoder compatibility," *IEEE Trans. Image Processing*, vol. 3, pp. 700–704, Sept. 1994.
[3] S.-W. Wu and A. Gersho, "Enhanced video compression with standardized bit stream syntax," in *Proc. ICASSP'93*, vol. 1, pp. 103–106.
[4] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, pp. 1445–1453, Sept. 1988.
[5] K. Ramchandran, A. Ortega, and M. Vetterli, "Bit allocation for dependent quantization with applications to multiresolution and MPEG video coders," *IEEE Trans. Image Processing*, vol. 3, pp. 533–545, Sept. 1994.

Subband Domain Coding of Binary Textual Images for Document Archiving

Ömer N. Gerek, A. Enis Çetin, Ahmed H. Tewfik, and Volkan Atalay

***Abstract*—In this work, a subband domain textual image compression method is developed. The document image is first decomposed into subimages using binary subband decompositions. Next, the character locations in the subbands and the symbol library consisting of the character images are encoded. The method is suitable for keyword search in the compressed data. It is observed that very high compression ratios are obtained with this method. Simulation studies are presented.**

***Index Terms*— Binary image coding, binary subband decomposition, document retrieval, textual image compression.**

I. INTRODUCTION

Efficient compression of binary textual images is an important problem in many applications such as document archives and digital libraries. A textual image typically consists of repeated patterns that are mostly characters and punctuation marks. Exploiting the redundancy of character repetitions is the key feature of the document image coding algorithms [2]–[7], which identify the locations of the characters in the image and replace them by pointers into a codebook of characters.

The main steps of a typical textual image compression (TIC) method can be summarized as follows.

1) Find and extract a character in the image.
2) Compare it with the symbol library consisting of the separate character images.
3) If the character exists in the symbol library, take the location only, otherwise add it to the library.
4) Compress the constructed library and the symbol locations.
5) A further step is proposed by Witten *et al.* [6] who encode the residue image so that lossless compression can be achieved.

In this paper, a subband domain textual image compression method is developed. The method consists of two stages as shown in Fig. 1. In the first stage, the document image is decomposed into subimages using a binary subband decomposition structure. The second stage of the algorithm consists of encoding the repetitions of the character images in the subband domain. In Section II, binary subband decomposition methods are reviewed. Encoding of the subband character images is described in Section III and it is experimentally observed that the subband domain method produces higher compression results than regular methods. Furthermore, the computational cost of encoding

Manuscript received March 14, 1997; revised December 14, 1998. This work was supported by the Turkish Scientific and Technical Research Council BDP program and the National Science Foundation under Grant INT-9406954. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Christine Podilchuk.

Ö. N. Gerek was with Bilkent University, TR-06533 Bilkent, Ankara, Turkey. He is now with the Signal Processing Laboratory, Swiss Federal Institute of Technology, CH-1015 Ecublens, Switzerland (e-mail: gerek@ltssun2.epfl.ch), on leave from Anadolu University, Eskişehir, Turkey.

A. E. Çetin is with the Department of Electrical Engineering, Bilkent University, TR-06533 Bilkent, Ankara, Turkey.

A. H. Tewfik is with the Department of Electrical Engineering, University of Minnesota, Minneapolis, MN 55455 USA.

V. Atalay is with the Department of Computer Engineering, Middle East Technical University, Ankara, Turkey.

Publisher Item Identifier S 1057-7149(99)07549-1.

1057–7149/99$10.00 © 1999 IEEE

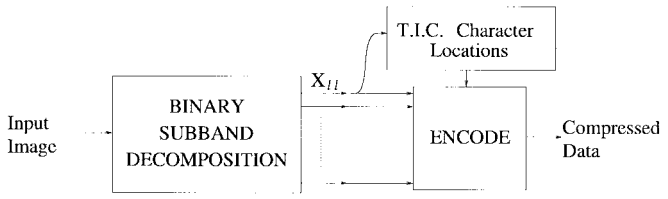

Fig. 1. Flow diagram of the algorithm.

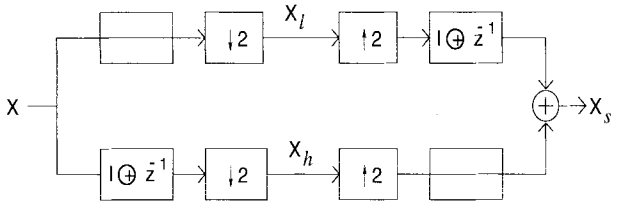

Fig. 2. One stage subband decomposition with "xor."

and keyword search decreases approximately by a factor of 2^{2M} for an M level subband decomposition structure. The pattern matching methods used in this work is presented in Section III-A, and the simulation results are given in Section III-B.

Step 5 of the TIC method is especially important in historical archives where lossless compression is required. In Section IV, lossless coding of documents from Ottoman archives, which contain more than 100 million documents written in Arabic letters, is considered.

The keyword search and document retrieval examples are presented in Section V followed by the conclusions in Section VI.

II. Stage I: Binary Subband Decomposition

The proposed procedure starts with a subband decomposition of the original image. In this section, we consider two classes of approaches for the subband decomposition. These classes are developed to perform binary subband decompositions that accept binary input data and produce binary output data.

The two classes of binary subband decomposition structures are, namely, the binary wavelet transforms (BWT) and the binary nonlinear subband decomposition (BNSD). The BWT was recently developed in [11] and BNSD is introduced here.

As in the regular subband decomposition, the different characteristics of the subband document images enable us to treat each subband image separately. In this way, one can utilize the spatial correlation between the pixels more effectively in individual subband images. Furthermore, the correlation between the bands can also be used to achieve high compression rates as in [9].

1) Binary Wavelet Transformations: The first class of decompositions is based on linear transformations. Recently, wavelet transform concept is extended to finite fields [10], [11]. In [11], the binary images are decomposed into subimages by performing modulo-2 operations. This scheme shares many of the important characteristics of the real wavelet transforms. Typically, the BWT yields an output similar to the thresholded output of a real wavelet transform operating on the image.

The analysis stage of the BWT consists of binary filters followed by decimation by a factor of two. Since all the additions are made in modulo-2, the operations can be replaced by logical "xor" operations.

Let $I = [x[n_1,\ n_2]]_{N\times N}$ be the image matrix whose entries are pixel values. The decomposition is performed by pre- and postmultiplying the image matrix with the analysis matrix T and its transpose T^t

$$I_d = T \times I \times T^t \tag{1}$$

where I_d is the decomposed image which is of the following form:

$$I_d = \begin{bmatrix} I_{ll} & I_{lh} \\ I_{hl} & I_{hh} \end{bmatrix} \tag{2}$$

where I_{ll}, I_{lh}, I_{hl}, and I_{hh} correspond to low–low, low–high, high–low, and high–high images, respectively. The transform matrix T is constructed from a pair of lowpass and highpass filter vectors $\boldsymbol{h}_0 = [h_0[0]h_0[1]\cdots h_0[P-1]]$ and $\boldsymbol{h}_1 = [h_1[0]h_1[1]\cdots h_1[Q-1]]$ as follows:

$$T = \begin{bmatrix} 2-\text{circulant}(\boldsymbol{h}_0) \\ 2-\text{circulant}(\boldsymbol{h}_1) \end{bmatrix} = \begin{bmatrix} \boldsymbol{h}_0 & 0 & 0 & 0 & 0 & \cdots \\ 0 & 0 & \boldsymbol{h}_0 & 0 & 0 & \cdots \\ & & & \ddots & & \\ - & - & - & - & - & - \\ \boldsymbol{h}_1 & 0 & 0 & 0 & 0\cdots & \\ 0 & 0 & \boldsymbol{h}_1 & 0 & 0\cdots & \\ & & & \ddots & & \end{bmatrix}. \tag{3}$$

If the filter size is less than N, then the filter vector is padded with zeros. In this way, the matrix T has two 2-circulant parts.

Due to the structure of the transformation matrix T, the BWT can also be implemented using a binary subband decomposition filter bank. Consider the following choice of analysis filter vectors:

$$h_0 = [1 \quad 0], \qquad h_1 = [1 \quad 1]. \tag{4}$$

In the one-dimensional (1-D) case, the subband signals are obtained as

$$x_j[n] = \sum_{i=0}^{1} x[2n-i]h_j[i], \qquad j = 0,\ 1 \tag{5}$$

where the arithmetic operations are carried out in modulo-2 arithmetic. Note that the modulo-2 summation corresponds to the logical "xor" operation. The lowband signal $x_0[n]$ is nothing but a decimated version of the original signal $x[n]$. The signal $x[n] * h_1[n]$ is zero everywhere except at the zero-to-one or one-to-zero transitions, and $x_1[n]$ is a decimated version of $x[n] * h_1[n]$. This operation corresponds to the analysis part of Fig. 2.

The synthesis operation is performed by the inverse matrices. In the case of the BWT described in (5), the reconstruction can be done by simple FIR filters

$$g_0 = [1 \quad 1], \qquad g_1 = [1 \quad 0]. \tag{6}$$

The reconstruction stage of this BWT is also shown in Fig. 2. For an arbitrary BWT, it may not be always possible to carry out the synthesis operation by a filterbank as in Fig. 2. In those cases, the reconstruction can be performed via binary matrix multiplications instead of synthesis filters [11]. The textual image coding results using the BWT of (5) are presented in Section III-B.

2) Binary Nonlinear Subband Decomposition: The second class of decompositions uses the nonlinear transformations. Recently, nonlinear subband decomposition schemes were developed in [12] and [13]. The regular nonlinear filterbank [13] was proposed to compress images containing sharp edges. Textual images mainly consist of edges and flat regions therefore the edge preserving property of the nonlinear filters make them attractive for textual image compression.

It is shown in [14] that the nonlinear subband decomposition of [13] can be extended to the binary fields, if all arithmetic operations are carried out in modulo-2. The 1-D binary nonlinear filterbank structure is shown in Fig. 3. Order statistics filters $M(\cdot)$ with appropriate regions of support and modulo-2 operations are used in this structure. Extension of the 1-D structure to two dimensions is straightforward.

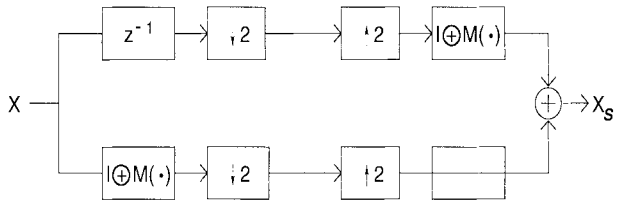

Fig. 3. One stage nonlinear subband decomposition with order statistics filter $M(\cdot)$.

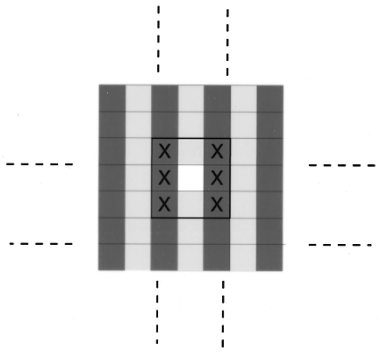

Fig. 4. Region of support for the horizontal direction nonlinear subband decomposition.

A suitable support region for the nonlinear filter $M(\cdot)$ is shown in Fig. 4. The current pixel is represented by the white square, the region of support of the order statistics filter must be on the dark gray pixels. Note that the light gray pixels are downsampled. In our simulations we used a support containing six pixels labeled by crosses in Fig. 4. In this case, the "highpass filtering" operation corresponds to subtracting the median of six elements from the current pixel value.

In order to obtain a binary counterpart of the nonlinear filter, the order statistics filter should select an integer indexed element. In this way, the GF(2) operations can be inserted in the calculations. The computation of the median of a binary list is a very simple operation. If the number of one's (zero's) is greater than the zero's (one's) in the list, then the median is one (zero). The other operations are simply summation and subtraction, which both correspond to the logical "xor."

The nonlinear subband decomposition scheme described in Fig. 3 is a generalized version of the binary filter banks shown in Fig. 2. It can be observed that the filterbank in Fig. 2 can be obtained from Fig. 3 replacing the $M(\cdot)$ with a delay element.

In this work, the Haar wavelet transform (WT) is also considered in simulation studies. The good time localization property of the Haar wavelet makes it suitable for the analysis of textual images which consist of sharp edges. However, the Haar subband images of a binary image have five gray levels after one stage of decomposition hence, Haar transform is computationally more costly than both BWT and BNSD. Coding results using the Haar WT are given in Section III-B.

In Fig. 5, the image of the character "a" is decomposed into subband images. The first image is the original letter "a." The next one is the Haar decomposed image. The image after that is the BWT transformed image of the letter "a," which also corresponds to the decomposition described in Fig. 2. The last image is the image obtained by the order statistical filter described in Fig. 3. It can be observed that the low–low subband images in all decompositions are coarse versions of the original image, and other subimages contain the edge information.

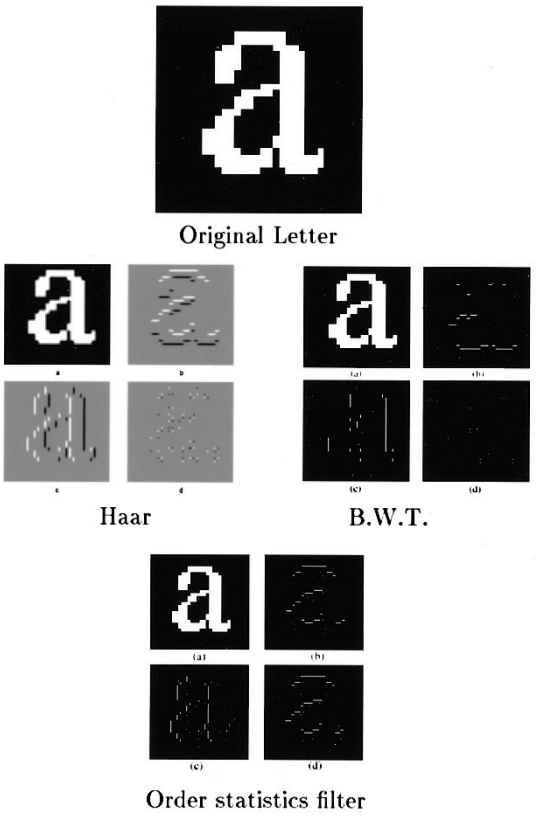

Fig. 5. Three different decompositions of the letter "a."

III. Stage II: Data Compression in the Subband Domain

There are many techniques that can be used for the compression of arbitrary binary images. In this work, we concentrate on a technique that is suitable for content based search in a document library. The technique should yield high compression ratios as well. The technique that we use preserves the identity and location of each symbol in the document. Let us suppose that the document image consists of L symbols

$$T_o = L((p+q) + H_l) + B_0 \tag{7}$$

bits, where p and q are the number of bits for representing the x and y coordinates of a character and H_l and B_o are the number of bits to represent the symbols and the compressed library, respectively. Clearly, the value of H_l is language and context dependent.

In our algorithm, after one level subband decomposition, the total number of bits changes to

$$T_1 = L((p' + q') + H_l) + B_1 \tag{8}$$

where p' and q' are the number of bits for representing the x and y coordinates of a character in the subband domain, and B_1 is the number of bits to represent the new library. Since both the old and the new libraries should contain the same number of symbols, the term H_l remains the same in both (7) and (8). If the locations of the characters are determined in the subband domain then $p' = p/2$ and $q' = q/2$ because the lowband image is a quarter size image. This may result in a one pixel shift of the character boundaries in the coded image. In the simulations, it is observed that human eye does not distinguish one pixel jitter in character locations in 300 dpi scanning resolution. Furthermore, the jitter can be eliminated by recording the locations of the characters in the original image, in which case $p' = p$ and $q' = q$.

TABLE I
TEXTUAL IMAGE COMPRESSION RESULTS: TIMES ROMAN

Times Roman font image size = 2500 × 720								
		Haar Basis, Method 1				Nonlinear filter Method	XOR filter Method	JBIG
	T.I.C.	Direct	with quant.	with bit plane appending	with quant. and bit plane appending	with bit plane appending	with bit plane appending	
Tot. no. of bits for symbol lib.	21768	31392	20280	12296	11904	11336	11888	
Tot. no. of bits for locations	4780	3564	3564	3564	3564	3564	3564	
Tot. no. of bits for symbol sequence	1812	1684	1684	1684	1684	1684	1684	
Tot. no. of lossy bits	28360	36640	25528	17544	17152	16584	17136	
Lossy CR	63.47	49.12	70.51	102.60	104.94	108.53	105.04	
Tot. no. of bits for residue	76017	76017	78118	76017	78118	76017	76017	
Lossless CR	17.25	15.98	17.37	19.24	18.89	19.44	19.32	12.54

TABLE II
TEXTUAL IMAGE COMPRESSION RESULTS: SANS SERIF

Sans-serif font image size = 2533 × 3370					
		Haar Basis, Method 1	Nonlinear filter Method	XOR filter Method	JBIG
	T.I.C.	with bit plane appending	with bit plane appending	with bit plane appending	
Tot. no. of bits for symbol lib.	17431	17208	15992	16848	
Tot. no. of bits for locations	17437	13211	13211	13211	
Tot. no. of bits for symbol sequence	6943	6665	6665	6665	
Tot. no. of lossy bits	41811	37084	35868	36724	
Lossy CR	204.16	230.19	237.99	232.44	
Tot. no. of bits for residue	229568	229568	229568	229568	
Lossless CR	31.45	32.01	32.16	32.06	26.22

The number of bits required for representing the multiresolution library is smaller than the original character library, i.e., $B_1 < B_0$ in all the examples studied in this paper (see Tables I and II).

Since the low–low subimage, $x_{ll}[\underline{n}]$, is a coarse version of the original textual image, the TIC algorithm described in Section I can be applied to $x_{ll}[\underline{n}]$. The procedure starts with determining every character image in $x_{ll}[\underline{n}]$ and comparing it with the images in the symbol library. If a similar image exists in the low–low library, then the character boundary and the location of this character is used for other subband images $x_{lh}[\underline{n}]$, $x_{hl}[\underline{n}]$, $x_{hh}[\underline{n}]$, as well. Various



Fig. 6. Symbol libraries of subband images ll, lh, hl, and hh.

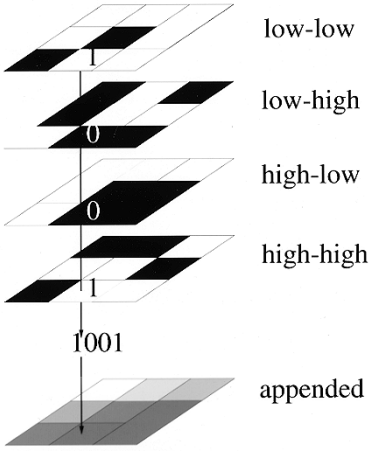

Fig. 7. Visualization of appending the bit-planes of subband images.

matching metrics, as will be described in Section IV can be used for the symbols inside the low–low subimage. If the image does not exist in the library, it is added to the library and the corresponding highband character images are added to their symbol libraries. In this way, four symbol libraries are constructed from the subband images as shown in Fig. 6.

The total number of bits to represent the four symbol libraries (SL) is smaller than the number of bits needed for the SL generated by the direct use of the TIC method. The efficiency of the subband domain textual compression is achieved by making use of the high correlation between subband images corresponding to a textual image. As a property of the binary transforms, the edges of the character images are almost at the same locations in all subbands. The bit planes of four subimages are appended as shown in Fig. 7 and the resulting multilevel image is compressed using an arithmetic coder which encodes the predicted 4 bit pixel value. This method achieves higher compression than individual compression of the subband character libraries.

In the case of the Haar WT, the SL images are not binary. The bit plane appended image has 4×5 gray levels. It is observed that this image can be quantized to 12 levels with almost no visual degradation. These library images are also used in our simulation studies. Since the Haar transform images are multilevel images, both keyword search and compression times are higher compared to the binary transforms because of the increase in the pattern matching time for multilevel images.

Subband decomposition levels can be increased as long as the low–low subimage contains character images suitable for the matching criteria defined in Section IV, i.e., if the scanner resolution is high. For example 2^K subband images are generated for a K level decomposition. In such a case, 2^K character libraries have to be encoded.

The lossless compression is achieved as in [6]. First the subband images are synthesized from the libraries, then the synthesized image

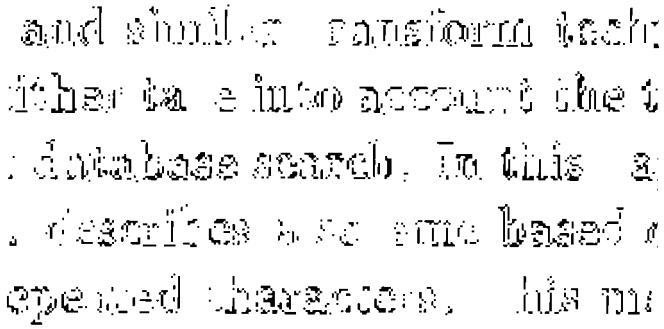

Fig. 8. The residue image for ll subband.

is used for encoding the residue image (Fig. 8). The synthesized image is almost the same as the lossy compressed image obtained by the regular TIC method. Due to this reason, the number of bits for encoding the residue image is comparable to the technique in [6].

A. Pattern Matching Criteria

The pattern matching criteria for extracting and comparing the character images inside the document is an important step that determines the success of the coding algorithm. In conjunction with the multiresolution TIC method, any pattern matching technique can be used in the construction of the symbol library.

In our simulation studies, we used two methods for symbol matching in the low–low subimage. The first one is the local template matching (LTM) criterion, described in [6] and [7]. This matching criterion is a very efficient algorithm that is well tuned for printed text. It detects the differences between imagewise similar looking symbols, such as “c” and “e.” The LTM method uses local characteristics as opposed to global methods which use the overall mismatch between two compared symbols by summing up contributions to the pixelwise matching errors from the whole area occupied by the symbol. The LTM method is a local method which bases its judgment on particular areas of mismatch that involve considerations of adjacent pixels. It looks at individual areas of the error map (the difference between the two symbols) rather than summing all discrepancies across the whole error map. In basic LTM, a match is rejected if any mismatched pixel in the error map has high number of neighbors that correspond to a mismatch. For example, one pixel shifts in the curves of a letter that may be caused by scanning noise will introduce error maps with line patterns, and by selecting the amount of mismatch neighbors as more than three, these errors with line patterns will not be rejected as mismatches. A more sophisticated example of LTM rejects a comparison if:

- a mismatch pixel in the error map has two or more mismatching neighbors, at least two of which are not connected to each other; and
- the corresponding pixel is entirely surrounded by either matching or mismatching pixels.

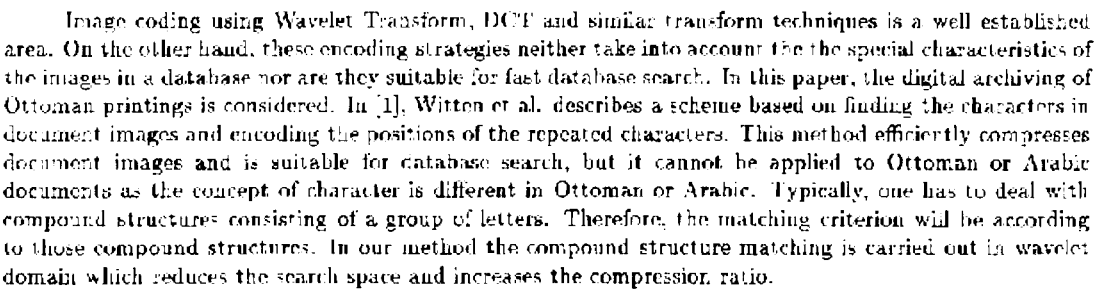
Image coding using Wavelet Transform, DCT and similar transform techniques is a well established area. On the other hand, these encoding strategies neither take into account the the special characteristics of the images in a database nor are they suitable for fast database search. In this paper, the digital archiving of Ottoman printings is considered. In [1], Witten et al. describes a scheme based on finding the characters in document images and encoding the positions of the repeated characters. This method efficiently compresses document images and is suitable for database search, but it cannot be applied to Ottoman or Arabic documents as the concept of character is different in Ottoman or Arabic. Typically, one has to deal with compound structures consisting of a group of letters. Therefore, the matching criterion will be according to those compound structures. In our method the compound structure matching is carried out in wavelet domain which reduces the search space and increases the compression ratio.

Fig. 9. Test document image 1.

Since our scheme uses lower scale textual images for template matching, the above rejection rules are tightened further to reject the match if there are four or more mismatching symbols, and any mismatching symbol cluster occurs in at least a 2×2 neighborhood.

This method also has some cases in which it fails to detect the matches correctly. However by fine-tuning the local parameters appropriately for the type of the document, the errors can be reduced. Furthermore, the method is sensitive to differences in the character sizes. Therefore, two different character sized images corresponding to the same character are treated as different symbols. The template matching precision performances are presented in Section V for the test images.

The second symbol matching method used in our work is based on the self organizing map (SOM) neural network model. This neural network model is used in order to illustrate the effects of using various symbol matching criteria in the multiresolution TIC.

A SOM is an unsupervised neural network that is mainly used for vector quantization and clustering [18]. It consists of a map of output nodes each receiving signals from a vector of input units. There is a weighted connection between each input unit and output node, forming a weight vector associated with each output node. The training algorithm modifies the network weights to extract the statistical properties of the training set and groups similar vectors into similar classes. A spatially localized region of nodes is formed at a location on the output map where the similarity of an input vector and the weight vector is maximum. Regions of the most activated nodes on the output map can be interpreted as clusters in the feature space.

In this study, horizontal projections of the symbol images extracted from the textual image are used as the input vectors to the SOM network. Similar components are mapped to topologically neighboring nodes. As a result, the symbol image with the least Euclidean distance to a winning node is selected as a representative symbol, and it is included in the symbol library. In the subband decomposed images, this task is carried out in the low–low subimage.

The SOM matching differs from the LTM matching in the sense that the representative symbols in the symbol library generated by the SOM are not necessarily the first of the similar symbols found in the raster scanned document image. After the construction of the symbol library using the SOM criterion, the repetition of the symbols are found by running the SOM neural network again, and obtaining the nearest representative node for each symbol in the textual image. In this phase of the coding, the node values are not updated.

The compression results for these template matching methods are presented in Section III-B, and the document retrieval results are presented in Section V.

B. Simulation Studies

Consider the 11-point Times Roman font printed single page document shown in Fig. 9. This document is scanned at 300 dpi and the resultant textual image has a size of 2500×720 pixels. The coding results for this test document using the LTM symbol matching criterion are presented in Table I. This image can be compressed only 12.54 times using the JBIG algorithm. The direct use of the textual image compression procedure [6] yields a lossy compression ratio (CR) of $63.47:1$ and a lossless CR of $17.25:1$. After a single level regular Haar decomposition bit planes of the resulting four symbol libraries are appended and coded. A CR of $102.60:1$ is obtained without introducing any visual degradation for the Times Roman image. By compressing the residual image lossless compression is achieved and the CR is $19.24:1$. In the cases of binary subband decompositions, the lossy and lossless CR's are $108.53:1$ and 19.44 for the binary median filter based subband decomposition, and $105.05:1$ and 19.34 for the filter given in Fig. 3. These results are better than the Haar wavelet case and it is a significant improvement compared to the regular TIC method. Furthermore, the encoding and decoding times for binary decompositions are very small because only logical operations are needed during the analysis, synthesis, and textual coding.

When the SOM method is used for symbol matching, the lossy and lossless CR's are $99.86:1$ and 18.14 for the binary median filter based subband decomposition, and $97.49:1$ and 18.01 for the filter given in Fig. 3.

In Fig. 10, a 12-point sans-serif font textual image with size 2533×3370 is shown. The compression results are higher in this case because the image size is bigger. The effects of subband decomposition over symbol library compression is more or less the same as the Times Roman example. Compression results for the sans-serif image with LTM matching criterion are given in Table II. Similar to the previous case, the SOM matching criterion gives comparable, but smaller compression ratios.

The NIST Special Database 8 [20] Times New Roman test documents are also compressed with the multiresolution TIC method. This test data contains ten documents of random ASCII characters. The average lossless compression ratio is obtained as $39.2:1$ with the LTM criterion. With the SOM criterion, the CR is $36.4:1$. When the regular TIC method is used, the CR is $35.2:1$ with the LTM criterion.

In this study, we use the SOM_PAK Program Package developed by Kohonen *et al.* [19]. Due to the restrictions of this software, only the horizontal projections of the character images are used as input vectors. As a result of this reduction in the representation, some inaccuracies occurred, yielding less compression ratios and inferior retrieval performance. By using more data in the input vectors, such as vertical projections, etc., the performance can be improved.

The compression speed is also improved in our method. With the introduction of binary subband decomposition prior to TIC, it is experimentally observed that the overall coding time has been reduced by a factor of 1.5 on Sun Sparc 10 and Sun Sparc 20 computers.

IV. Compression of Ottoman Script Images

Today, there are more than 30 independent nations within the borders of the Ottoman Empire that lasted more than six centuries until its breakdown in World War I. Almost all of these countries need documents from the archives of the Ottoman Empire, which

is an important task for summarizing the content of a video. It is usually desired to perform this operation by using the readily produced compressed video data. Most of the video streams are already stored in MPEG-I or MPEG-II form and one should avoid full decompression of the data in order to reduce the computational complexity.

This paper describes a method for selecting key frames by using a number of parameters extracted from the MPEG video stream. The parameters are directly extracted from the compressed video stream without decompression. A combination of these parameters are then used in a rule based decision system. There is very little calculation in both extracting the parameters and the in the decision for key frame rule computation. As a result of these, the overall operation is very quickly performed. The experimental results show that this method can select the distinctive frames of a video stream successfully.

The histogram measure depends on the DC values of the DCT coefficients of each block, and is somewhat more tricky. For intra blocks, the DC value is used directly. For inter blocks, the DC value corresponding to that block is interpolated from the DC values of the reference frame blocks according to the given motion vectors. Typically, the motion vectors point to an 8×8 region which has regions inside four 8×8 regular blocks. The motion compensated DC value is calculated by interpolating the DC values of the four regular blocks at the point indicated by the motion vector.

The thresholds have to be chosen according to the general characteristics of the sequence. In our experiments, we used CIF size (352×288) frames for the MPEG sequence. The frame rate is 8 frames/sec. The motion measure and the thresholds should be modified accordingly if the frame rate is altered because the motion characteristics of the sequence changes according to the frame rate. The Group Of Picture (GOP) structure has the sequence

Fig. 10. Test document image 2.



Fig. 11. Two compound structures. One occurs entirely inside other.

contain more than 100 million files. Some of these documents are deteriorating and computer archiving seems to be the only solution.

The Ottoman script is a connected script and it is based on the Arabic alphabet. A typical word consists of compounded letters as in handwritten text. Therefore, a regular TIC for documents containing isolated characters cannot encode Ottoman document images.

As in the case of regular document images the Ottoman script images are first processed by a binary subband decomposition structure and the symbol library is constructed from the low–low subimage. In order to handle connected text the symbol library construction and pattern matching procedures of TIC are modified. In compression of documents with isolated characters, a symbol is marked and extracted from the document image. This symbol is then compared with the previously selected symbols in the symbol library. If it matches to any of the characters in the library, only its location is encoded, otherwise this character image is included in the library. In other words, the extracted single character is only compared to the symbols in the library. In the modified version of the TIC, the document is processed in multiple passes. In Ottoman documents, there are isolated symbols corresponding to single letters as well as long connected symbols which are combination of letters. Usually, the longer symbols include some letters that can also be found separately inside the document. When the smaller isolated symbols are encoded and removed from the document, the longer symbols split into smaller symbols. Each of these smaller symbols may also be contained in other connected symbols. Therefore, they will later cause other connected symbols to split further. The removal of symbols from the document image

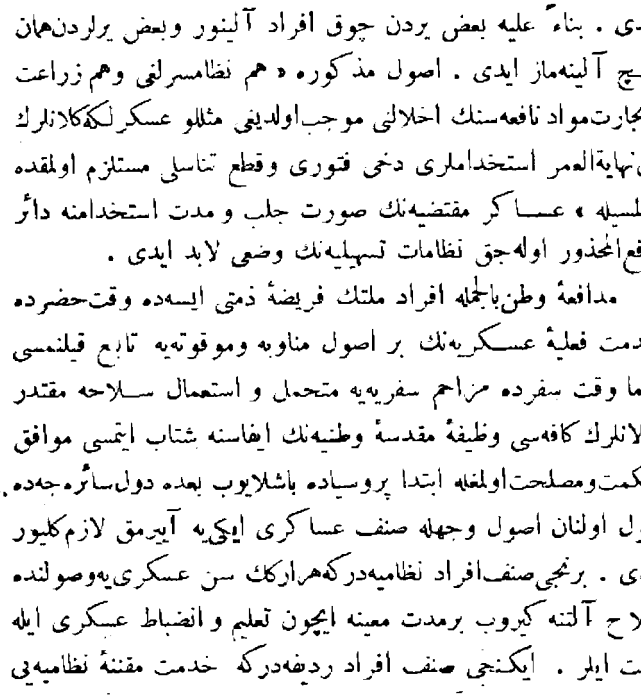

Fig. 12. Sample Ottoman script image.

TABLE III
TEXTUAL IMAGE COMPRESSION RESULTS FOR A SINGLE OTTOMAN SCRIPT IMAGE

Ottoman script image size = 1488×1464				
	T.I.C.	Nonlinear filter Method	XOR filter Method	JBIG
Tot. no. of bits for symbol lib.	14408	11376	11440	
Tot. no. of bits for locations	8616	6144	6144	
Tot. no. of bits for symbol sequence	2520	2520	2520	
Tot. no. of lossy bits	25560	20040	20104	
Lossy CR	85.23	108.70	108.36	
Tot. no. of bits for residue	60088	60240	60184	
Lossless CR	25.43	27.13	27.13	17.79

TABLE IV
TEXTUAL IMAGE COMPRESSION RESULTS FOR 20 PAGES OF OTTOMAN SCRIPT IMAGES

20 page Ottoman script image				
	T.I.C.	Nonlinear filter Method	XOR filter Method	JBIG
Lossy CR	183.73	235.94	235.86	
Lossless CR	30.28	31.36	31.38	17.8

TABLE V
QUERY RESULTS FOR TEN COMPRESSED NIST IMAGES FOR STRINGS "rt," "zE," "3V," "Po," "va," AND "&9" AT CR = 49:1

Image	Correct hit						False alarm					
	"rt"	"zE"	"3V"	"Po"	"va"	"&9"	"rt"	"zE"	"3V"	"Po"	"va"	"&9"
No4	3/3	3/3	5/6	2/2	0/0	2/2	0	0	0	0	0	1
No5	3/3	2/2	2/2	2/2	2/2	3/3	0	0	0	0	1	0
No6	0/0	0/0	0/0	3/3	0/0	0/0	0	0	0	0	0	0
No8	1/1	3/3	2/2	2/2	2/2	1/1	0	0	0	0	0	0
No10	2/2	0/0	2/2	2/2	0/0	1/1	0	0	0	0	0	0
No11	0/0	0/0	0/0	0/0	2/2	1/1	0	0	0	0	0	0
No12	1/1	1/1	0/0	0/0	0/0	0/0	0	0	0	0	0	0
No15	0/0	0/0	0/0	1/1	0/0	0/0	0	0	0	0	0	1
No17	1/1	0/0	1/1	0/0	1/1	0/0	0	0	0	0	0	0
No20	0/0	1/1	0/0	0/0	0/0	0/0	0	0	0	0	0	0

is performed by sliding the symbol image all over the document image and keeping track of the correlation. If the correlation between the symbol and the document image is high at a location, the marked symbol is subtracted from that location of the image, and the location is encoded. When the end of the document is reached during correlation, all the places of the document which contains shapes similar to the marked symbol are removed from the image. In this way, even if the marked symbol is connected to some other symbols inside the document, the similar portions will be extracted, and the longer connected scripts will be broken into smaller pieces. After several such passes, all of the document is encoded.

The extraction of marks from the document image is shown in Fig. 11. The resulting symbol library mainly contains basic compound structures, curves and lines that eventually form the Ottoman script.

An Ottoman document image is shown in Fig. 12. This image is scanned at 300 dpi and its size is 1488×1464. After a single level of binary subband decomposition the image is compressed to 108.70:1. The regular TIC method produces a CR of 85.23:1 and there is no visually noticeable difference between the two coded images. Significant savings can be achieved with the use of multiresolution processing.

Some of the documents may have historical importance that requires lossless compression. By encoding the residue image, the image shown in Fig. 12 can be compressed 27.14 times. Without the binary subband decomposition, the lossless CR is 25.44:1 and JBIG compression of this image gives only a CR = 17.79:1. A 20-page Ottoman document is also compressed in a lossless way. In this case, JBIG remains at the CR of 17.8:1, while the CR of the TIC method increases to 31.35:1 with one level of binary subband decomposition, and it is 30.27:1 without any subband decomposition. Similar to the normal printed text, the longer the document gets, the higher the compression ratio becomes because the same symbol library is used throughout the document. The compression results are summarized in Tables III and IV.

V. DOCUMENT RETRIEVAL AND KEYWORD SEARCH

The issues of document retrieval and keyword search are important for document archiving and they should be considered as an integral part of any document coding method. In most applications, a query in a document library corresponds to a keyword search or a sample image.

The keyword or pattern search algorithm can be carried out over the character library of the low–low subimage and the locations of the characters. If the character images of the word or phrase exist in the symbol library and if they occur consecutively in the location database, then a match is found. In this way, the keyword search time is theoretically reduced by a factor of 2^2 after one level of binary decomposition. Such a speed-up for a query is especially important in large document archives. On a Sun Sparc 10 computer, the query response time for the encoded images with multiresolution TIC is less than half of that required for images encoded by regular TIC.

We use the textual images in the NIST Special Database 8 [20] for testing the keyword search and document retrieval performance of the multiresolution TIC method. Ten pages containing Times New Roman fonts are compressed using our method with the LTM criterion, and the compressed data is tested with arbitrarily selected six key strings consisting of two letter symbols. The queries are made by example images within the documents. Table V gives the number of correct identifications of the words over all the occurrences of these words. The amount of misses and false identifications are presented in individual columns. These experimental results indicate that the number of missing symbols out of all 61 key strings is one for the document of ten pages. The total number of false alarms for these key strings is three. When the TIC is used without subband decomposition, the number of misses is also found to be one, and the number of false alarms is determined as one. When the SOM pattern matching criterion is used, the retrieval accuracy is reduced both for the TIC and the multiresolution TIC at the same amount. With the SOM method, the number of misses for the key strings is five for both the TIC and the multiresolution TIC. The SOM based pattern matching method is used to illustrate the fact that various pattern matching methods can be incorporated to the multiresolution TIC. As described in Section III-A, the coding performance and the retrieval accuracy of the SOM based method can be improved by increasing the input vector size.

The performance of the TIC method in noisy documents is determined by the behavior of the character matching criteria. The retrieval of a character is possible only if the noisy character image can still be identified as the representative symbol in the symbol library. Using the software obtained from the University of Maryland WWW server [21], it is observed that with the speckle noise at level 1/10, half of the characters which should be found in the symbol library start to mismatch the symbol library. Furthermore, the compression ratio falls from 39.2:1 to 24:1. The same effect is observed for the TIC method applied without subband decomposition. At the same noise level, 35% of the characters are mismatched, and the compression ratio falls to 23:1. In such cases, a noise eliminating preprocessing is necessary. The size of the character images is also critical in noise sensitivity. It is more difficult to make symbol matching over small character images, so the number of decomposition levels should be adjusted according to the size of the character images. It is observed that good coding and retrieval results can be obtained by a single level subband decomposition for 12-point document images scanned at 300 dpi.

VI. Conclusions

In this work, a multiscale textual image compression (TIC) method is introduced. The document image is first decomposed into subimages using a binary subband decomposition structure. Next, the locations of the character images in the subband domain are encoded. In this way, higher compression ratios than regular TIC methods can be obtained without introducing any visual degradation.

Performance of various subband decomposition structures are experimentally tested, and it is observed that the binary wavelet transform and the binary nonlinear subband decomposition structures produce comparable coding results.

A feature of the coding method is that the keyword search or the creation of the symbol library can be carried out in the coded bitstream in subbands. Other advantages of the subband scheme include multiresolution image viewing and computational efficiency. As a direct consequence of the subband decomposition, a low-resolution image x_{ll} is obtained and it can be used for fast preview purposes to decrease the bandwidth usage. The compression of the textual image is a computationally costly operation. The multiresolution approach reduces the encoding time as the pattern matching is carried out in low-resolution images.

References

[1] V. K. Govindan and A. P. Shivaprasad, "Character recognition—A review," *Pattern Recognit.*, vol. 23, pp. 671–683, 1990.

[2] R. N. Ascher and G. Nagy, "A means for achieving a high degree of compaction on scan-digitized printed text," *IEEE Trans. Comput.*, vol. C-23, pp. 1174–1179, Nov. 1974.

[3] W. K. Pratt *et al.*, "Combined symbol matching facsimile data compression system," *Proc. IEEE*, vol. 68, pp. 786–796, July 1980.

[4] M. J. J. Holt, "A fast binary template matching algorithm for document image data compression," in *Pattern Recognit.*, J. Kittler, Ed. Berlin, Germany: Springer-Verlag, 1988.

[5] O. Johnsen, J. Segen, and G. L. Cash, "Coding of two-level pictures by pattern matching and substitution," *Bell Syst. Tech. J.*, vol. 62, pp. 2513–2545, May 1983.

[6] I. H. Witten *et al.*, "Textual image compression: Two-stage lossy/lossless encoding of textual images," *Proc. IEEE*, vol. 82, pp. 878–888, June 1994.

[7] I. H. Witten, A. Moffat, and T. C. Bell, *Managing Gigabytes.* San Mateo, CA: Morgan Kaufmann, 1999.

[8] J. W. Woods, Ed., *Subband Image Coding.* Boston, MA: Kluwer, 1991.

[9] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Processing*, vol. 41, pp. 3445–3462, Dec. 1993.

[10] G. Caire, L. Grossman, and H. V. Poor, "Wavelet transforms associated with finite cyclic groups," *IEEE Trans. Inform. Theory*, vol. 39, pp. 1157–1166, July 1993.

[11] M. D. Swanson and A. H. Tewfik, "A binary wavelet decomposition of binary images," *IEEE Trans. Image Processing*, vol. 5, pp. 1637–1651, Dec. 1996.

[12] F. J. Hampson and J. C. Pesquet, "A nonlinear subband decomposition with perfect reconstruction," in *IEEE Int. Symp. Image Processing*, 1996.

[13] O. Egger, W. Li, and M. Kunt, "High compression image coding using an adaptive morphological subband decomposition," *Proc. IEEE*, vol. 83, pp. 272–287, Feb. 1995.

[14] O. N. Gerek, M. N. Gürcan, and A. E. Çetin, "Binary morphological subband decomposition for image coding," in *IEEE Int. Symp. Time-Frequency and Time Scale Analysis*, 1996.

[15] M. N. Gürcan, O. N. Gerek, and A. E. Çetin, "A morphological subband decomposition structure using GF(N) arithmetic," in *Proc. IEEE Int. Conf. Image Processing*, Switzerland, Sept. 1996.

[16] M. D. Swanson, "Issues in image databases: Coding for content-based browsing and retrieval, data hiding, and copyright protection," Ph.D. dissertation proposal, Dept. Elect. Eng., Univ. Minnesota, Minneapolis, MN, June 3, 1996.

[17] M. D. Swanson, S. Hosur, and A. H. Tewfik, "Image coding for content-based retrieval," in *Proc. 1996 SPIE Conf. Visual Communication and Image Processing*, Orlando, FL, 1996, vol. I, pp. 1–12.

[18] T. Kohonen, *Self Organizing Maps.* Berlin, Germany: Springer-Verlag, 1997.

[19] T. Kohonen, J. Hynninen, J. Kangas, and J. Laaksonen, "SOM_PAK: The self-organizing map program package," Tech. Rep. A31, Lab. Comput. Inform. Sci., Helsinki Univ. Technol., Espoo, Finland, 1996.

[20] R. A. Wilkinson, "NIST special database 8: Machine print database," Nat. Inst. Stand. Technol., Adv. Syst. Div., Image Recognit. Grp., Oct. 1, 1992.

[21] D. S. Doermann and S. Yao, "Generating synthetic data for text analysis systems," in *Proc. 4th Symp. Document Analysis and Information Retrieval*, 1995, pp. 449–467.

On Independent Color Space Transformations for the Compression of CMYK Images

Ricardo L. de Queiroz

***Abstract*—Device- and image-independent color space transformations for the compression of CMYK images were studied. A new transformation (to a YYCC color space) was developed and compared to known ones. Several tests were conducted leading to interesting conclusions. Among them, color transformations are not always advantageous over independent compression of CMYK color planes. Another interesting conclusion is that chrominance subsampling is rarely advantageous in this context. Also, it is shown that transformation to YYCC consistently outperforms the transformation to YCbCrK, while being competitive with the image-dependent KLT-based approach.**

I. Introduction

Images for monitor display are commonly stored or transmitted in popular color spaces such as RGB, CIELAB, YUV, YCbCr, etc. [1], [2]. Images to be printed are commonly rendered in a subtractive color space such as CMYK [1], [2]. Sometimes documents contain pictures in other color spaces that are then converted to CMYK for printing. Good color management practice demands color correction for the printing device [3], [4]. We refer to the printing color space as device-dependent-CMYK or simply device-CMYK.

The color correction process not only maps the input gamut to the device's gamut but it also incorporates other factors such as under-color removal (UCR) and tone-reproduction curves (TRC). Those steps are commonly based on a printer model [3], [4] which is empirically derived. CMYK here refers to device-CMYK where the device is known (the dependence on a device is also true for all color spaces derived from CMYK unless a printer-model-based mapping is performed). It is clear that the conversion between color spaces such as CIELAB and device-CMYK is an ill-posed problem, not only because it converts between three and four color bases, but also because it is not a linear process. Thus, invertibility is not guaranteed.

It is often necessary to compress CMYK images for a number of reasons, such as bandwidth or memory reduction. Fig. 1 shows a

Manuscript received February 18, 1998; revised April 12, 1999. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Thrasyvoulos N. Pappas.

The author is with Xerox Corporation, Webster, NY, 14580 USA (e-mail: queiroz@wrc.xerox.com).

Publisher Item Identifier S 1057-7149(99)07553-3.

1057–7149/99$10.00 © 1999 IEEE