the proposed algorithm does not require the *a priori* knowledge of input signals, it is attractive for use in practical applications where the environment is unknown or space varying.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their useful suggestions.

REFERENCES

[1] N. A. M. Verhoeckx *et al.,* "Digital echo cancellation for baseband data transmission," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-27, pp. 768–781, Dec. 1979.
[2] E. H. Satorius and S. T. Alexander, "Channel equalization using adaptive lattice algorithms," *IEEE Trans. Commun.,* vol. COMM-27, pp. 899–905, June 1979.
[3] B. Widrow *et al.,* "Adaptive noise cancelling: Principles and applications," *Proc. IEEE,* vol. 63, pp. 1692–1716, Dec. 1975.
[4] B. Widrow, J. McCool, M. Larimore, and C. Johnson, "Stationary and nonstationary learning characteristics of the LMS adaptive filter," *Proc. IEEE,* vol. 64, pp. 1151–1162, Aug. 1976.
[5] P. A. Maragos, R. W. Schafer, and R. M. Mersereau, "Two-dimensional linear prediction and its application to adaptive predictive coding of images," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-32, pp. 1213–1229, Dec. 1984.
[6] M. Das and N. K. Loh, "New studies on adaptive predictive coding of images using multiplicative autoregressive models," *IEEE Trans. Image Processing,* vol. 1, pp. 106–111, Jan. 1992.
[7] Y.-S. Chung and M. Kanefsky, "On 2-D recursive LMS algorithms using ARMA prediction for ADPCM encoding of images," *IEEE Trans. Image Processing,* vol. 1, pp. 416–422, July 1992.
[8] M. R. Azimi-Sadjadi and S. Bannour, "Two-dimensional recursive parameter identification for adaptive Kalman filtering," *IEEE Trans. Circuits Syst.,* vol. 38, pp. 1077–1081, Sept. 1991.
[9] H. Youlal, M. Janati-I, and M. Najim, "Two-dimensional joint process lattice for adaptive restoration of images," *IEEE Trans. Image Processing,* vol. 1, pp. 366–378, July 1992.
[10] S. B. Kesler and A. S. Elfishawy, "Adaptive change detection in image sequence," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing,* Albuquerque, NM, 1990, pp. 2189–2192.
[11] M. M. Hadhoud and D. W. Thomas, "The two-dimensional adaptive LMS (TDLMS) algorithm," *IEEE Trans. Circuits Syst.,* vol. 35, pp. 485–494, May 1988.
[12] S. Haykin, *Adaptive Filter Theory,* 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1991.
[13] W. B. Mikhael and S. M. Ghosh, "Two-dimensional variable step-size sequential adaptive gradient algorithms with applications," *IEEE Trans. Circuits Syst.,* vol. 38, pp. 1577–1580, Dec. 1991.
[14] A. M. Sequeira and C. W. Therrien, "A new 2-D fast RLS algorithm," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing,* Albuquerque, NM, 1990, pp. 1401–1404.
[15] W. B. Mikhael and S. M. Ghosh, "Two-dimensional block adaptive filtering algorithms," in *Proc. IEEE Int. Symp. Circuits and Systems,* San Diego, CA, May 1992, pp. 1219–1222.
[16] M. R. Azimi-Sadjadi and H. Pan, "Two-dimensional block diagonal LMS adaptive filtering," *IEEE Trans. Signal Processing,* vol. 42, pp. 2420–2429, Sept. 1994.
[17] X. Liu, P. Baylou, and M. Najim, "A new 2D fast lattice RLS algorithm," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing,* San Francisco, CA, Mar. 1992, pp. III.329–III.332.
[18] G. A. Clark, S. K. Mitra, and S. R. Parker, "Block implementation of adaptive digital filters," *IEEE Trans. Circuits Syst.,* vol. CAS-28, pp. 584–592, June 1981.
[19] M. L. Honig and D. G. Messerschmitt, *Adaptive Filters: Structures, Algorithms, and Applications.* Boston, MA: Kluwer, 1984.
[20] W. B. Mikhael and F. H. Wu, "Fast algorithms for block FIR adaptive digital filtering," *IEEE Trans. Circuits Syst.,* vol. CAS-34, pp. 1152–1160, Oct. 1987.
[21] ——, "A fast block FIR adaptive digital filtering algorithm with individual adaptation of parameters," *IEEE Trans. Circuits Syst.,* vol. 36, pp. 1–10, Jan. 1989.
[22] T. Wang and C.-L. Wang, "Comments on 'A fast block FIR adaptive digital filtering algorithm with individual adaptation of parameters,'" *IEEE Trans. Circuits Syst. II: Analog and Digital Signal Processing,* vol. 39, pp. 254–256, Apr. 1992.
[23] ——, "On the optimum design of the block adaptive FIR digital filter," *IEEE Trans. Signal Processing,* vol. 41, pp. 2131–2140, June 1993.
[24] N. J. Bershad, "Analysis of the normalized LMS algorithm with Gaussian inputs," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-34, pp. 793–806, Aug. 1986.
[25] R. R. Bitmead and B. D. O. Anderson, "Performance of adaptive estimation algorithms in dependent random environments," *IEEE Trans. Automat. Contr.,* vol. AC-25, pp. 788–794, Aug. 1980.
[26] J. S. Lim, *Two-Dimensional Signal and Image Processing.* Englewood Cliffs, NJ: Prentice-Hall, 1990.

Nonexpansive Pyramid for Image Coding Using a Nonlinear Filterbank

Ricardo L. de Queiroz, Dinei A. F. Florêncio, and Ronald W. Schafer

***Abstract*— A nonexpansive pyramidal decomposition is proposed for low-complexity image coding. The image is decomposed through a nonlinear filter bank into low- and highpass signals and the recursion of the filterbank over the lowpass signal generates a pyramid resembling that of the octave wavelet transform. The structure itself guarantees perfect reconstruction and we have chosen nonlinear filters for performance reasons. The transformed samples are grouped into square blocks and used to replace the discrete cosine transform (DCT) in the Joint Photographic Expert Group (JPEG) coder. The proposed coder has some advantages over the DCT-based JPEG: computation is greatly reduced, image edges are better encoded, blocking is eliminated, and it allows lossless coding.**

I. INTRODUCTION

Multiresolution techniques provide a convenient way of exploring the several levels of spatial redundancy existing in most images. The Laplacian pyramid [1] and wavelet transforms [2] became quite popular for image processing and coding. The Joint Photographic Expert Group (JPEG) baseline system (DCT-JPEG) [3] is a *de facto* standard for lossy image compression. However, it is based on the discrete cosine transform (DCT), which is somewhat expensive to compute and can also cause ringing and blocking artifacts [3]. In this paper, we present a JPEG-based coder, which uses a nonlinear transform instead of the DCT. The transform is based on a multiresolution filterbank, does not require multiplications or floating point numbers, and allows lossless coding. Comparison to the DCT-JPEG at several bit rates shows the superiority of the proposed coder, both objectively and subjectively. The JPEG standard also includes a dedicated mode (non-DCT-based) for lossless coding [3]. We show that it is also

Manuscript received November 5, 1995; revised May 19, 1997. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. C.-C. Jay Kuo.

R. L. de Queiroz is with Xerox Corporation, Webster, NY 14580 USA (e-mail: queiroz@wrc.xerox.com).

D. A. F. Florêncio is with David Sarnoff Research Center, Princeton, NJ 08544 USA (e-mail: dflorencio@sarnoff.com).

R. W. Schafer is with the School of Electrical Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: floren@eedsp.gatech.edu).

Publisher Item Identifier S 1057-7149(98)01004-5.

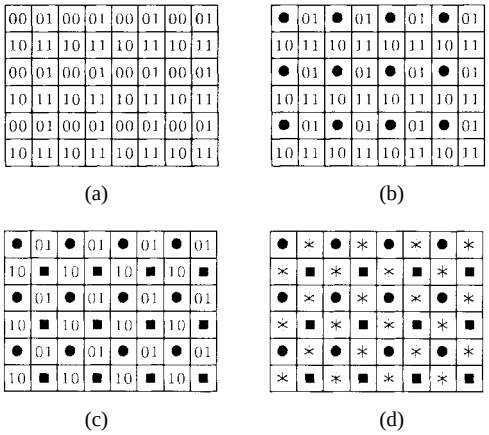


Fig. 1. Interpolation approach. (a) Polyphase components. (b) The samples $x_{00}(\mathbf{n})$ (marked with •) are found by simple subsampling of $x(\mathbf{n})$. (c) The samples $x_{00}(\mathbf{n})$ are used to interpolate (predict) the value of the samples in $x_{11}(\mathbf{n})$ (marked with ■). (d) The samples of $x_{00}(\mathbf{n})$ and $x_{11}(\mathbf{n})$ combined are used to interpolate (predict) the values of $x_{01}(\mathbf{n})$ and $x_{10}(\mathbf{n})$ (both marked $*$). Note that the interpolation step in (c)→(d) is similar to the interpolation step in (b)→(c) if we just tilt (c) by 45°.

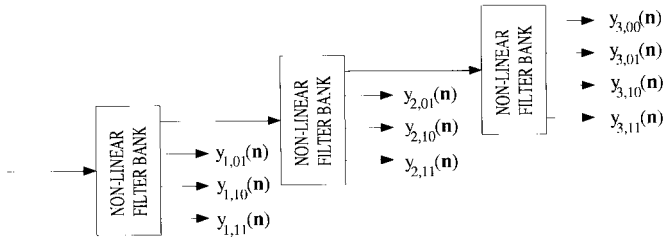


Fig. 2. Generation of a multiresolution pyramid.

outperformed by the proposed nonlinear coder. Besides, since no dedicated lossless mode is required, the coder is also convenient for nearly lossless coding.

II. The Transform

Perfect reconstruction (PR) in critically decimated systems is generally guaranteed by imposing conditions on the filter coefficients. When dealing with nonlinear filters, general conditions are still unknown [4]. For this reason, nonlinear filterbanks were restricted to noncritically decimated cases [5], [6]. Recently, a new approach for critically decimated nonlinear filter banks has been introduced [7], [8], where PR is obtained by imposing restrictions on the filter *structure* instead of on the filter *coefficients*. However, the structure in [7], [8] has been used before with linear filters [9]–[13].

Let the picture elements (pixels or pels) in the input image be denoted by $x(n_1, n_2) = x(\mathbf{n})$. The two-dimensional (2-D) polyphase components of the signal (4-channels) are given by $x_{\mathbf{i}}(\mathbf{m}) = x(2\mathbf{m}+\mathbf{i})$, for $\mathbf{i} = [i_0, i_1]^T$, $i_k \in 0, 1$ as shown in Fig. 1. Using the same notation for the transformed signal $y(\mathbf{n})$, the decomposition (analysis) for one pyramid level can be described as

$$y_{00}(\mathbf{n}) = x_{00}(\mathbf{n}) \tag{1}$$

$$y_{11}(\mathbf{n}) = x_{11}(\mathbf{n}) - F_0[x_{00}(\mathbf{n})] \tag{2}$$

$$y_{01,\,10}(\mathbf{n}) = x_{01,\,10}(\mathbf{n}) - F_1[x_{00}(\mathbf{n}),\, x_{11}(\mathbf{n})] \tag{3}$$

where F_i is any linear or nonlinear function and $x_{01,\,10}(\mathbf{n})$ is the quincunx grid formed by $x_{01}(\mathbf{n})$ and $x_{10}(\mathbf{n})$. It is clear that $x(\mathbf{n})$ can be perfectly reconstructed since we can always find $x_{ij}(\mathbf{n})$ as a function of $y_{ij}(\mathbf{n})$ and of previously reconstructed polyphase components (synthesis). The relative spatial arrangement between the two rectangular grids x_{00} and x_{11} is the same as that between the two quincunx grids $x_{00,\,11}$ and $x_{01,\,10}$. The difference is a rotation of 45°. Therefore, F_1 can be essentially the same as F_0. We further extend the notation to define

$$x_{s,\,i_0 i_1}(n_0,\, n_1) = x(2^s n_0 + 2^{s-1} i_0,\, 2^s n_1 + 2^{s-1} i_1). \tag{4}$$

As in the wavelet and pyramid transforms [1], [2], one can connect the lowpass output of a stage directly to the input of another stage as shown in Fig. 2.

In image coding applications, the subbands are quantized. We can avoid excessive accumulation of quantization error across subbands

procedure $process(y_{k,ij}(\mathbf{n}), \hat{x}_{k,ij}(\mathbf{n}), z(\mathbf{n}), Q_l)$

$\tilde{x}_{k,ij}(\mathbf{n}) = \text{interpolate}(z(\mathbf{n}))$

$y_{k,ij}(\mathbf{n}) = x_{k,ij}(\mathbf{n}) - \tilde{x}_{k,ij}(\mathbf{n})$

encode $Q_l\{y_{k,ij}(\mathbf{n})\}$

$\hat{y}_{k,ij}(\mathbf{n}) = Q_l^{-1}\{Q_l\{y_{k,ij}(\mathbf{n})\}\}$

$\hat{x}_{k,ij}(\mathbf{n}) = \tilde{x}_{k,ij}(\mathbf{n}) + \hat{y}_{k,ij}(\mathbf{n})$

main multistage loop

$l = 1$

$process(y_{S,00}(\mathbf{n}), \hat{x}_{S,00}(\mathbf{n}), \emptyset, Q_1)$

for $k = S : -1 : 1$

$process(y_{k,11}(\mathbf{n}), \hat{x}_{k,00}(\mathbf{n}), Q_{l+1})$

$process(y_{k,01}(\mathbf{n}), \{\hat{x}_{k,00}(\mathbf{n}), \hat{x}_{k,11}(\mathbf{n})\}, Q_{l+2})$

$process(y_{k,10}(\mathbf{n}), \{\hat{x}_{k,00}(\mathbf{n}), \hat{x}_{k,11}(\mathbf{n})\}, Q_{l+2})$

$l = l + 2$

end

Notes:

$\hat{x}_{k-1,00}(\mathbf{n}) = (\hat{x}_{k,00}(\mathbf{n}), \hat{x}_{k,01}(\mathbf{n}), \hat{x}_{k,10}(\mathbf{n}), \hat{x}_{k,11}(\mathbf{n}))$

Q is the quantization operator

Fig. 3. Algorithm for the pyramidal decomposition.

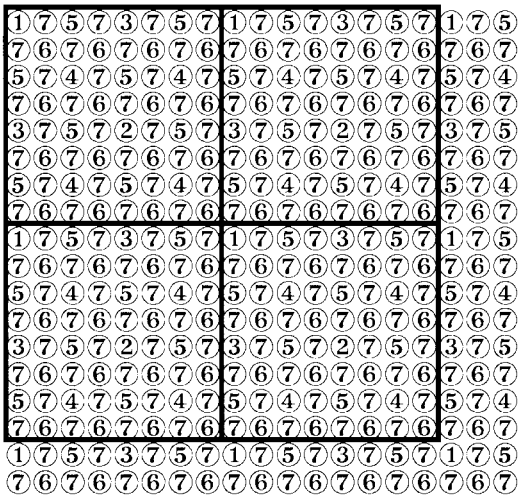


Fig. 4. Illustration of a 3-stage decomposition. Samples labeled "$n+$ 1" are transformed by computing the interpolation error using the four nearest samples among those labeled "1" through "n." We can also group the samples into blocks, as indicated.

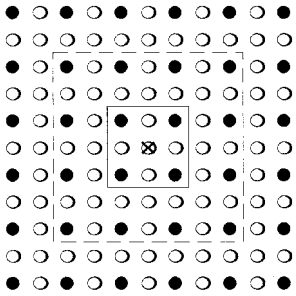


Fig. 5. Typical support region of the interpolation filters.

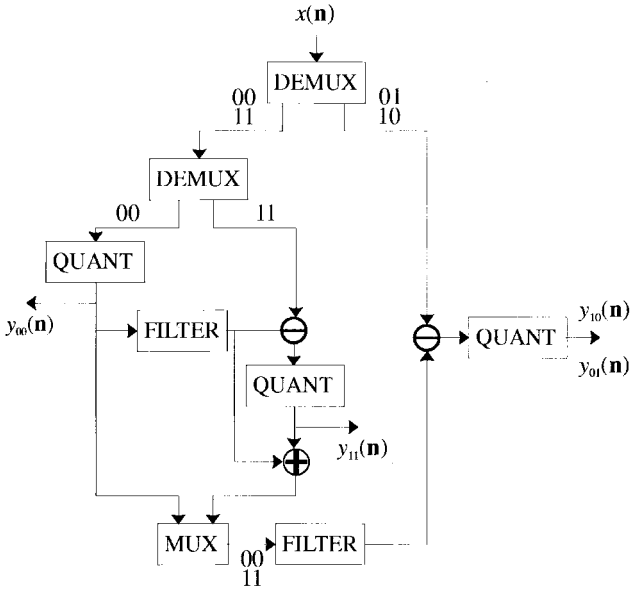


Fig. 6. The structure to generate the filterbank. DEMUX is a 2-D demultiplexer separating input into polyphase components. MUX is a multiplexer to group the components. The branches are labeled to show their respective polyphase components.

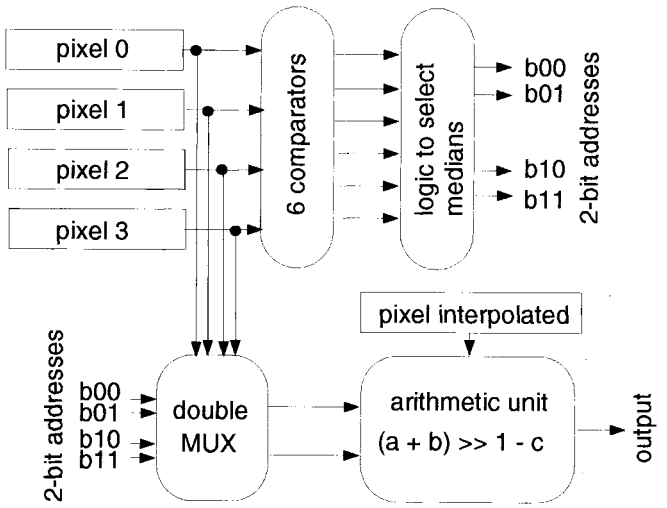


Fig. 7. Implementation of the transform for each pixel (nonlinear filtering and a subtraction). The four input pixels are compared in parallel generating six output bits, which are fed into a simple logic unit that decodes the addresses of the two intermediary samples (medians). Such addresses are used to multiplex the numbers and feed them to the arithmetic unit which will average them and subtract the pixel which is to be interpolated.

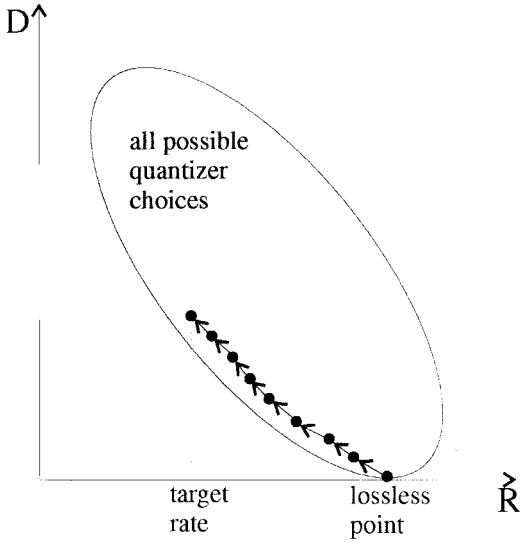


Fig. 8. Recursive procedure for JPEG quantizer table design. We start at a the minimum distortion point and change the quantizers in an organized manner in several trials. The new set of quantizers is chosen for the set corresponding to a new $(R,\ D)$ point whose variation $(\Delta R,\ \Delta D)$ yields lower inclination. The process is continued until the desired rate or distortion is reached. It is straightforward to change the algorithm to start from the point of maximum distortion.

by using a feedback loop (local reconstruction) similar to that used in differential pulse code modulation (DPCM) systems. Furthermore, F_i should be a good interpolator in order to reduce the information sent along the subbands. The description of the analysis process is given in Fig. 3, where Q_n represents the quantization process at the nth step. For example, for uniform quantizers with step size Δ_n, $Q_n\{t\} = round(t/\Delta_n)$ and $Q_n^{-1}\{t\} = t\Delta_n$. For $S = 3$ (a depth-3 decomposition), an example of the sequence of pixels used is given in Fig, 4. In this figure, samples labeled "1" through "n" are used to interpolate samples labeled "$n+$ 1." This pyramid is similar to the one proposed by Endoh and Yamazaki [9].

Note that we can group the samples into $2^S \times 2^S$ blocks (as the 8×8 block in Fig. 4) to replace traditional block transforms.

The choice of the filters boils down to the choice of an interpolation method. In Fig. 5, samples in the grid marked by • are available to interpolate the sample marked with $\otimes$. Typical support regions use four or 16 neighbors. Simple nonlinear interpolation has been shown capable of producing better results than much more complex linear filters [14]. Instead of exploring a complex *ad hoc* design for the filter, we decided to settle on the median filter, which is one of the simplest designs. For four input samples it is defined by *discarding the minimum and maximum values and averaging the remaining two*. The median interpolator is convenient for three main reasons. First, it will likely discard uncommon pixels, e.g., an impulse coinciding with the sampling grid (a linear filter would blur the impulse). Second, it will perfectly interpolate sharp edges present in the image, whenever the edge is horizontal, vertical or at 45° inclination. Last but not least, it has low computational complexity. It can be implemented using only four comparisons, one addition and one division by two (i.e., a shift).

The complexity of the transform in a *per-pixel* basis is the complexity of the interpolation plus a subtraction to find the interpolation error. This is because all subbands are calculated essentially using the same method but involving different pixels. Besides the low-

TABLE I
PREDETERMINED QUANTIZER STEP SIZE SETS

pictorial							graphical						
Δ_1	Δ_2	Δ_3	Δ_4	Δ_5	Δ_6	Δ_7	Δ_1	Δ_2	Δ_3	Δ_4	Δ_5	Δ_6	Δ_7
1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	1	2	2	3	3	4	5
3	3	3	3	4	4	5	1	2	4	4	6	6	6
3	4	4	5	5	6	9	1	4	7	7	9	10	13
4	4	5	6	7	9	11	1	4	10	11	12	12	13
4	5	6	7	9	11	15	2	6	12	12	14	15	18
4	5	7	8	10	13	19	3	12	28	31	31	53	84
5	6	7	10	11	16	26	5	15	35	35	42	42	100
5	6	9	12	15	20	40	12	47	83	83	150	255	255
6	8	10	15	19	27	56	255	255	255	255	255	255	255
6	11	13	17	23	35	62							
8	13	18	26	36	56	105							
10	16	30	43	61	105	255							
255	255	255	255	255	255	255							

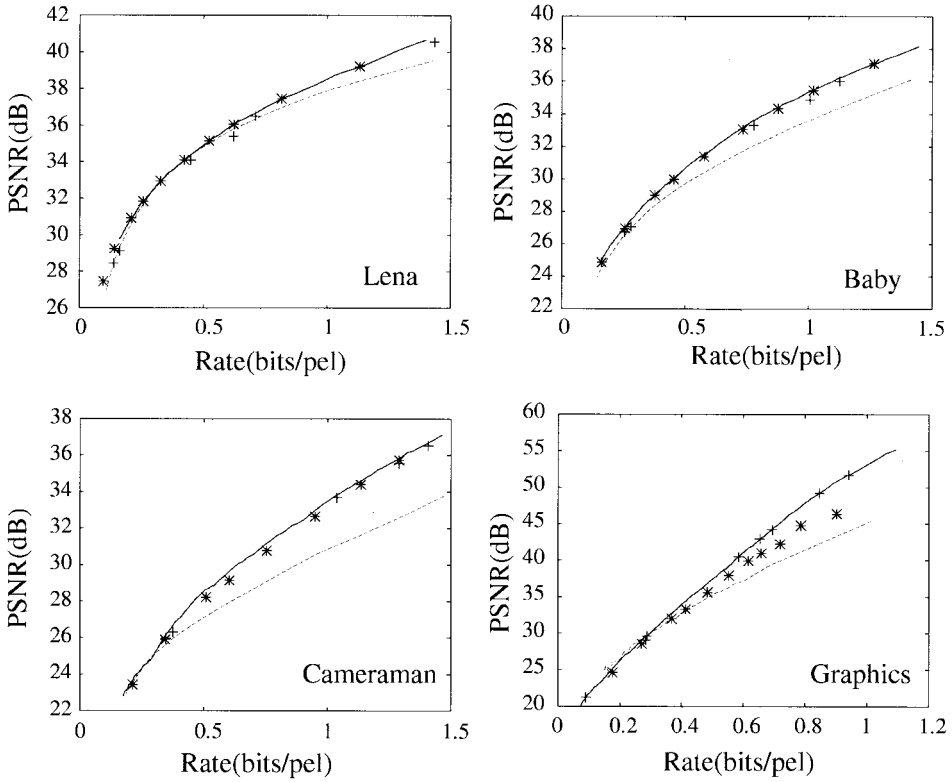


Fig. 9. Plot of PSNR versus bit rate for several images. Dashed line is for regular DCT-based JPEG. Solid line is for recursively designed quantizers. (*) are results found using quantizers from the pictorial set. (+) are results found using quantizers from the graphical set. In all cases, optimized Huffman codes were used. Image cameraman has 256 × 256 pixels while the others have 512 × 512 pixels.

complexity software implementation, the proposed pyramid can be easily implemented in hardware. A block diagram for one stage is given in Fig. 6. The samples are demultiplexed into polyphase components, making memory management easier. Fig. 7 shows a simple scheme to implement the nonlinear filter, which also computes the interpolation error. This diagram demands only four stages of very simple logic units. It is suitable for hardware implementation (e.g., through ASIC or FPGA) for a fast pixel throughput rate because most operations can be parallelized or piped. A key feature of this interpolator is that, unlike most linear transforms, it can be implemented using fixed point arithmetic in software or hardware with B-bit precision, where B is the number of bits per pixel of the input signal. Therefore, the complexity per pixel is far below its linear counterparts such as the cosine and the wavelet transforms.

The pyramid lies in the intersection of hierarchical filterbanks and interpolative predictive systems. It can be viewed as a nonlinear filterbank pyramid, being a particular association of building blocks presented in [4]. However, it can also be viewed as a hierarchical interpolative DPCM, since it applies the same concept used in [10]–[13].

III. JPEG-BASED CODING

We embedded the proposed transform into JPEG. The idea is to replace the DCT coefficients by our pyramid samples. This has been done before by replacing the DCT by the discrete wavelet transform

TABLE II
PERFORMANCE (IN B/PIXEL) OF SOME LOSSLESS COMPRESSORS. FOR LOSSLESS JPEG WE ALWAYS SELECTED THE BEST PREDICTOR

Coders	Lenna	Camer.	Graphics	Baby
NLP-JPEG	4.50	5.23	1.56	4.94
Lossless-JPEG	4.70	5.37	1.94	5.07
S+P Huffman	4.38	5.29	2.51	4.86
GZIP	6.80	6.33	1.19	6.91

(DWT) and using the rest of the JPEG coder [15], [16]. Here, we follow the same principle: using three stages ($S = 3$) and grouping the pyramid samples into blocks as shown in Fig. 4. We, therefore, refer to our coder as NLP-JPEG.

The image is transformed using the nonlinear pyramid with quantizer feedback, where $2S + 1 = 7$ step sizes are selected for uniform quantizers. The lowpass samples are encoded using a 2-D DPCM as

$$\tilde{x}_{S,\,00}(\mathbf{n}) = \frac{1}{2}(\hat{x}_{S,\,00}(\mathbf{n} - \begin{bmatrix}1\\0\end{bmatrix}) + \hat{x}_{S,\,00}(\mathbf{n} - \begin{bmatrix}0\\1\end{bmatrix})) \tag{5}$$

$$y_{S,\,00}(\mathbf{n}) = x_{S,\,00}(\mathbf{n}) - \tilde{x}_{S,\,00}(\mathbf{n}) \tag{6}$$

$$\hat{y}_{S,\,00}(\mathbf{n}) = Q_1^{-1}\{Q_1\{y_{S,\,00}(\mathbf{n})\}\} \tag{7}$$

$$\hat{x}_{S,\,00}(\mathbf{n}) = \hat{y}_{S,\,00}(\mathbf{n}) + \tilde{x}_{S,\,00}(\mathbf{n}) \tag{8}$$

where we encode the value of $Q_1\{y_{S,\,00}(\mathbf{n})\}$. The transformed samples are grouped into blocks of $2^S \times 2^S = 8 \times 8$ samples as in Fig. 4. For each block, the quantized samples are reorganized into a vector. The samples are scanned from those labeled "1" to those labeled "7" in Fig. 4. The quantized samples are encoded using standard JPEG entropy coding based on Huffman codes.

The DCT-JPEG employs 64 uniform quantizers (one for each DCT coefficient), while the proposed one uses only seven for three stages. We propose two approaches for step size selection. We can either select from a predefined set of quantizer steps or design them using image-dependent recursive algorithms.

When compression time is not crucial, we can apply a recursive *optimization* algorithm to find the quantizer steps for a given bit rate. We use a variation of Wu and Gersho's algorithm [17], which is based on a rate-distortion criteria and was developed for DCT-JPEG [17]. The original approach is too complex, but for the NLP-JPEG it is a practical solution. The difference is because the NLP-JPEG runs faster than DCT-JPEG and because the search in 7-dimensional space is more manageable than in a 64-dimensional one. The iterative procedure is illustrated in Fig. 8. Because of the nonlinearity of the transform and because of its recursive nature (quantizer feedback), better results are obtained if we constrain the step sizes to be nondecreasing. On a SPARCStation 20 the quantizer steps for a 512 $\times$ 512-pels image (Lena) are found in 1 min 30 s for 1 b/pel. It takes 3 min 30 s to obtain the quantizer steps for 0.5 b/pel and 6 min for 0.3 b/pel. For decompression, the quantizer design method is irrelevant. If compression time is crucial we recommend using the predefined quantizer sets given in Table I, in which case the NLP-JPEG would run about twice as fast as DCT-JPEG. The predefined quantizer tables are divided into sets of graphical and pictorial data in order to better suit to different image classes. The quantizer sets presented are basic anchor points and, whenever desired, one may *interpolate* between quantizer step values in order to achieve the desired bit rate.

IV. CODER PERFORMANCE

Since $\Delta_n = 1$ leads to lossless coding, we compared the performance of the NLP-JPEG for lossless compression against three dedicated lossless coders: 1) non-DCT lossless JPEG coder; 2) Huffman-based Said–Pearlman lossless coder [18]; and 3) GnuZIP, which is a popular LZW-based compressor. Results are shown in Table II.

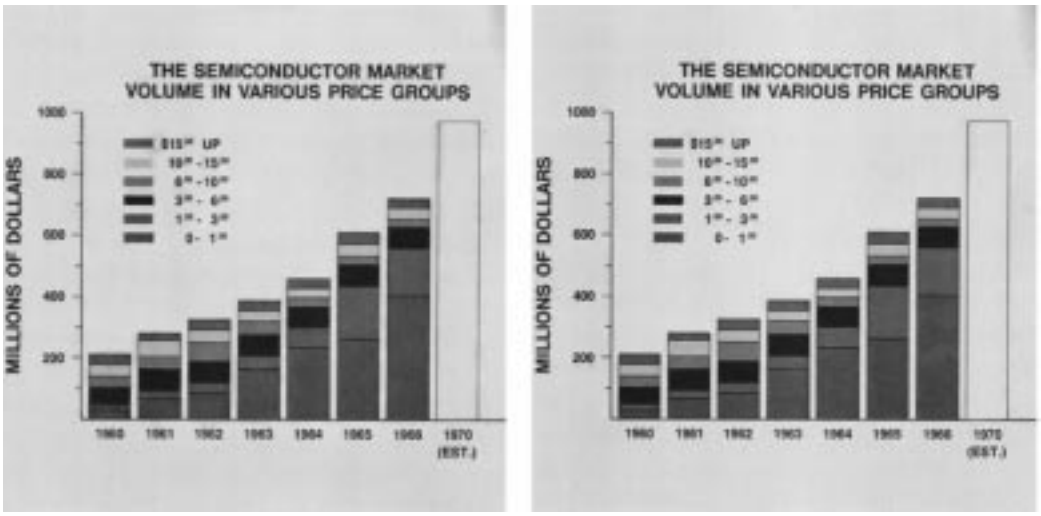
(a)

(b)

(c)

Fig. 10. Test images: (a) Baby (512 $\times$ 512 pels). (b) Graphics (512 $\times$ 512 pels). (c) Cameraman (256 $\times$ 256 pels).

(a) (b)



(c) (d)

(e) (f)

Fig. 11. Baby image encoded at 0.45 b/pixel. (a) NLP-JPEG (PSNR = 30.04 dB). (b) DCT-JPEG (PSNR = 29.22 dB). Graphics image encoded at 0.44 b/pixel. (c) NLP-JPEG (PSNR = 35.35 dB). (d) DCT-JPEG (PSNR = 32.70 dB). Cameraman image. (e) NLP-JPEG (PSNR = 32.48 dB at 0.907 b/pixel). (f) DCT-JPEG (PSNR = 30.26 dB at 0.914 b/pixel).

Tests were carried to compare the performances of NLP- and DCT-JPEG. Fig. 9 shows peak signal-to-noise ratio (PSNR) values for typical images. Resolution is 256 × 256-pels for the cameraman image and 512 × 512 pels for the others. In these plots, we used optimized Huffman codes in JPEG for both the DCT and NLP based schemes. We used three sets of quantizers for the NLP-JPEG: 1) optimized quantizers found by recursive methods; 2) quantizers in the pictorial set; 3) quantizers in the graphical set. If we replace the nonlinear predictor by a linear one (average of the four neighbors), PSNR performance drops typically by 0.1–0.3 dB.

Although, in most cases, both approaches yield relatively close PSNR results, they generate images that look radically different in terms of the artifacts they produce. The DCT-JPEG approach at low bit rates produces the familiar ringing and blocking artifacts. The NLP-JPEG approach has no ringing or blocking and generally encodes edges well, but it does not accurately encode texture regions. Neither coder will reproduce texture regions well at lower bit rates. While the NLP-JPEG removes the texture, DCT-JPEG attempts to produce a diffuse approximation. NLP-JPEG may be advantageous in applications where sharp edges are more important than texture regions.

For the original images shown in Fig. 10, examples of reconstructed images (using both NLP-JPEG and DCT-JPEG) are presented for subjective comparison in Fig. 11.

The NLP-JPEG is aimed as a less-complex replacement for the DCT-JPEG. JPEG was not designed to work at low bit rates. The block sizes and the entropy coding stage make both coders suitable for moderate compression ratios, i.e., 0.2 b/pixel and above.[1] More sophisticated coders do exist that easily outperform DCT-JPEG and NLP-JPEG at a higher computational cost. Examples are the Said–Pearlman (SP) coder [19] and Joshi–Crump–Fisher (JCF) coder [20]. The NLP-JPEG runs about five to six times faster than the SP coder and well over 50 times faster than the JCF coder. Furthermore, these coders require buffering the whole image, which can be difficult for large images such as scanned documents (typically 30 MB per color channel at 600 spots per inch).

Other papers deal with similar pyramidal structures for image coding. In [8], the reported results point to a performance inferior to DCT-JPEG for the bit rates of interest. In [13] high-performance is obtained with a nonprefiltered pyramid, but applying classification and trellis coded quantization (TCQ). It is fairly more complex than NLP-JPEG and requires buffering the image. While [13] reports PSNR gains over DCT-JPEG ranging from 0.5–2 dB, it also reports improvements of up to 6 dB by applying its more sophisticated classification and coding procedures. This may hint at future research directions.

V. Conclusions

We presented a PR critically decimated nonlinear pyramidal structure for image compression based on the cascade of a two-step nonlinear filterbank. Image coding tests were carried using JPEG and replacing the DCT by the proposed pyramidal scheme. The proposed scheme shows comparable or superior performance over DCT-JPEG both objectively and subjectively. It also outperforms the alternative non-DCT-based JPEG algorithm for lossless coding. The most appealing feature of the pyramid is its complexity, as it is far less complex than most popular linear transforms such as the DCT and is suitable for hardware implementation. In a software implementation, the NLP-JPEG runs about twice as fast as the DCT-JPEG.

Future research directions point to giving up some of the simplicity inherent in the NLP-JPEG for an increase in performance. This can be done by applying adaptive classification methods and by using entropy-constrained TCQ [13] to the quincunx pyramid described in Section II.

[1] For default Huffman codes, JPEG uses 6 b per 64-pel block (0.094 b/pixel) just to encode a constant image.

References

[1] P. Burt and J. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Trans. Commun.*, vol. COMM-31, pp. 532–540, Apr. 1983.

[2] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[3] W. B. Pennebaker and J. L. Mitchell, *JPEG: Still Image Compression Standard*. New York: Van Nostrand Reinhold, 1993.

[4] D. A. F. Florêncio and R. W. Schafer, "Perfect reconstructing nonlinear filter banks," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, Atlanta, GA, 1996, pp. 1815–1818.

[5] F. K. Sun and P. Maragos, "Experiments on image compression using morphological pyramids," *Vis. Commun. Image Processing IV*,vol. 1199, pp. 1303–1312, 1989.

[6] A. Toet, "A morphological pyramidal image decomposition," *Pattern Recognit. Lett.*, vol. 9, pp. 255–261, May 1989.

[7] D. F. Florêncio and R. W. Schafer, "A nonexpansive pyramidal morphological image coder," in *Proc. IEEE Int. Conf. Image Processing*, Austin, TX, 1994, vol. 2, pp. 331–334.

[8] O. Egger and W. Li, "Very low bit rate image coding using morphological operators and adaptive decompositions," in *Proc. IEEE Int. Conf. Image Processing*, Austin, TX, 1994, vol. 2, pp. 326–330.

[9] T. Endoh and Y. Yamazaki, "Progressive coding scheme for multi level images," in *Proc. Picture Coding Symp.*, Tokyo, Japan, Apr. 1986, pp. 21–22.

[10] P. Roos, M. Viergever, M. C. Van Dijke, and J. H. Peters, "Reversible image compression of medical images," *IEEE Trans. Med. Imag.*, vol. 7, pp. 328–336, Dec. 1988.

[11] R. L. de Queiroz and J. T. Yabu-Uti, "On a hybrid predictive-interpolative scheme for reducing processing speed in DPCM TV CODEC's," in *Proc. Europ. Signal Processing Conf.*, Barcelona, Spain, Sept. 1990, vol. II, pp. 797–780.

[12] H. Sahinoglou and S. D. Cabrera, "A high-speed pyramid image coding algorithm for a VLSI implementation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 1, pp. 369–374, Dec. 1991.

[13] E. Gifford, B. R. Hunt, and M. Marcellin, "Image coding using adaptive recursive interpolative DPCM," *IEEE Trans. on Image Processing*, vol. 4, pp. 1061–1069, Aug. 1995.

[14] D. F. Florêncio and R. W. Schafer, "Post-sampling aliasing control for images," in *Proc. Int. Conf. Acoustics, Speech and Signal Processing*, Detroit, MI, 1995, vol. II, pp. 893–896.

[15] R. de Queiroz *et al.*, "Wavelet transforms in a JPEG-like image coder," in *Proc. SPIE Conf. VCIP*, 1994, SPIE, vol. 2308, pp. 1662–1673.

[16] J. Bradley, C. Brislawn, and T. Hopper, "The FBI wavelet/scalar quantization standard for gray-scale fingerprint image compression," in *Proc. SPIE VCIP*, Orlando, FL, 1993, vol. 1961, pp. 293–304.

[17] S. Wu and A. Gersho, "Rate-constrained picture-adaptive quantization for JPEG baseline coders," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, Minneapolis, MN, 1993, vol. V, pp. 389–392.

[18] A. Said and W. A. Pearlman, "Reversible image compression via multiresolution representation and predictive coding," in *Proc. SPIE Conf. Vis. Commun. Image Processing*, Cambridge, MA, 1993, SPIE, vol. 2094, pp. 664–674.

[19] A. Said and W. A. Pearlman, "A new, fast, and efficient image CODEC based on set partitioning in hierarchical trees," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 243–250, June 1996.

[20] R. Joshi, V. Crump, and T. R. Fisher, "Image subband coding using arithmetic and trellis coded quantization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, pp. 512–523, Dec. 1995.