A NON-LINEAR FILTER BANK FOR IMAGE CODING

Ricardo L. de Queiroz

Xerox Corporation
800 Phillips Rd. M/S 128-27E
Webster, NY, 14580
queiroz@wrc.xerox.com

Dinei A. F. Florêncio

School of Electrical Engineering
Georgia Institute of Technology
Atlanta, GA 30332
floren@eedsp.gatech.edu

ABSTRACT

A new non-expansive pyramidal decomposition is proposed. The signal is decomposed through a non-linear filter bank into low- and high-pass signals and the recursion of the filter bank over the low-pass signal generates a pyramid resembling that of the octave wavelet transform. The transformed samples were grouped into square blocks and used to replace the DCT in the JPEG coder. The proposed coder shows several advantages: computation is greatly reduced compared to the DCT, image edges are better encoded, blocking is eliminated, and it allows lossless coding.

1. INTRODUCTION

The Laplacian pyramid [1] became quite popular for image processing and coding despite the fact that it imposes an expansion of the number of samples, limiting the performance of the coder. Expansiveness can be eliminated by directly applying an association of filter banks [2], which has been shown to be equivalent to the discrete wavelet transform [2]. The JPEG baseline system (referred here simply as JPEG) [3] is a *de facto* standard for lossy compression of gray-level or color images. However, it is based on the discrete cosine transform (DCT) which is somewhat expensive to compute. In this paper we present a JPEG-based coder which uses a non-linear transform instead of the DCT. The transform is an enhanced version of the pyramidal structure presented in [4] and it does not require multiplications, nor floating point numbers of any kind.

Perfect reconstruction (PR) in critically decimated systems is generally guaranteed by imposing conditions on the filter coefficients. When dealing with non-linear filters, no such general conditions exist. For this reason, non-linear filter banks were restricted to non-critically decimated cases [6, 7, 8]. Recently, a new approach for critically decimated non-linear filter banks has been introduced [9, 4], where PR is obtained by imposing restrictions in the filter *structure* instead of on the filter *coefficients*. We use here a particularization of a general framework [5].

0-7803-2972-4/96$5.00©1996 IEEE

2. THE TRANSFORM

2.1. One stage

Let the picture elements (pixels or pels) in the input image be denoted by $x(n_1, n_2)$. With the usual notation for multidimensional signals [10], we define the vector $\mathbf{n} = [n_1, n_2]^T$ and denote the signal by $x(\mathbf{n})$. We define the polyphase components of the signal as $x_{\mathbf{i}}(\mathbf{m}) = x(\mathbf{Mm} + \mathbf{i})$, for $\mathbf{M} = \begin{bmatrix} M_1 & 0 \\ 0 & M_2 \end{bmatrix}$ and for $\mathbf{i} = [i_0, i_1]^T$, $0 \leq i_k < M_k$. We are concerned with 2D signals and with the case $M_1 = M_2 = 2$, so that $\mathbf{i}$ can assume the values representing one out of four polyphase components: (0,0), (0,1), (1,0), (1,1), i.e., they obey the following grid:

```
(00)(01)(00)(01)(00)(01)(00)(01)(00)(01)
(10)(11)(10)(11)(10)(11)(10)(11)(10)(11)
(00)(01)(00)(01)(00)(01)(00)(01)(00)(01)
(10)(11)(10)(11)(10)(11)(10)(11)(10)(11)
```

We also represent the transformed signal into its polyphase components (subbands) as $y_{\mathbf{i}}(\mathbf{n})$, such that similar notation applies to samples of $y(\mathbf{n})$ as well.

The decomposition for one level of the specific structure that we used to construct the pyramid can be described as:

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

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

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

where F_i is any linear or non-linear function and $x_{01,10}(\mathbf{n})$ is the quincunx grid formed by $x_{01}(\mathbf{n})$ and $x_{10}(\mathbf{n})$. The signal can be perfectly reconstructed by making:

$$x_{00}(\mathbf{n}) = y_{00}(\mathbf{n}) \quad (4)$$

$$x_{11}(\mathbf{n}) = y_{11}(\mathbf{n}) + F_0(x_{00}(\mathbf{n})) \quad (5)$$

$$x_{01,10}(\mathbf{n}) = y_{01,10}(\mathbf{n}) + F_1(x_{00}(\mathbf{n}), x_{11}(\mathbf{n})) \quad (6)$$

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 degrees. Therefore, F_1 can be essentially the same as F_0 [5].

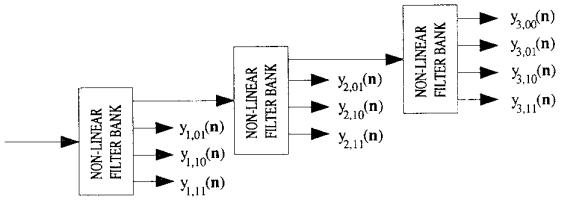

Figure 1: Cascading stages to form a multiresolution pyramid.

2.2. The pyramid

As usual in the filter banks literature, we call the subband decomposition process *analysis* and the reconstruction process *synthesis*. We further extend the notation to define

$$x_{s,i_0 i_1}(n_0, n_1) = x(2^s n_0 + 2^{s-1} i_0, 2^s n_1 + 2^{s-1} i_1). \quad (7)$$

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

In image coding applications the subbands are quantized. Given the sequential nature of the decomposition process, we can avoid excessive accumulation of quantization error across subbands by using a feedback loop (local reconstruction) similar to that used in DPCM systems. Furthermore, for maximum compression, F_i should be a good interpolator in order to minimize the information sent along the subbands. A description of the analysis process[1] is given by:

$l = 1$
$y_{S,00}(\mathbf{n}) = x_{S,00}(\mathbf{n})$
$\hat{y}_{S,00}(\mathbf{n}) = Q_l^{-1}\{Q_l\{y_{S,00}(\mathbf{n})\}\}$
$\hat{x}_{S,00}(\mathbf{n}) = \hat{y}_{S,00}(\mathbf{n})$
for $k = S : -1 : 1$
  $\tilde{x}_{k,11}(\mathbf{n}) = \text{interpolate}(\hat{x}_{k,00}(\mathbf{n}))$
  $y_{k,11}(\mathbf{n}) = x_{k,11}(\mathbf{n}) - \tilde{x}_{k,11}(\mathbf{n})$
  $\hat{y}_{k,11}(\mathbf{n}) = Q_{l+1}^{-1}\{Q_{l+1}\{y_{k,11}(\mathbf{n})\}\}$
  $\hat{x}_{k,11}(\mathbf{n}) = \tilde{x}_{k,11}(\mathbf{n}) + \hat{y}_{k,11}(\mathbf{n})$
  $\tilde{x}_{k,01}(\mathbf{n}) = \text{interpolate}(\hat{x}_{k,00}(\mathbf{n}), \hat{x}_{k,11}(\mathbf{n}))$
  $y_{k,01}(\mathbf{n}) = x_{k,01}(\mathbf{n}) - \tilde{x}_{k,01}(\mathbf{n})$
  $\hat{y}_{k,01}(\mathbf{n}) = Q_{l+2}^{-1}\{Q_{l+2}\{y_{k,01}(\mathbf{n})\}\}$
  $\hat{x}_{k,01}(\mathbf{n}) = \tilde{x}_{k,01}(\mathbf{n}) + \hat{y}_{k,01}(\mathbf{n})$
  $\tilde{x}_{k,10}(\mathbf{n}) = \text{interpolate}(\hat{x}_{k,00}(\mathbf{n}), \hat{x}_{k,11}(\mathbf{n}))$
  $y_{k,10}(\mathbf{n}) = x_{k,10}(\mathbf{n}) - \tilde{x}_{k,10}(\mathbf{n})$
  $\hat{y}_{k,10}(\mathbf{n}) = Q_{l+2}^{-1}\{Q_{l+2}\{y_{k,10}(\mathbf{n})\}\}$
  $\hat{x}_{k,10}(\mathbf{n}) = \tilde{x}_{k,10}(\mathbf{n}) + \hat{y}_{k,10}(\mathbf{n})$
  $l = l + 2$
end

Note that at each iteration

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

[1] more details can be found in [5]

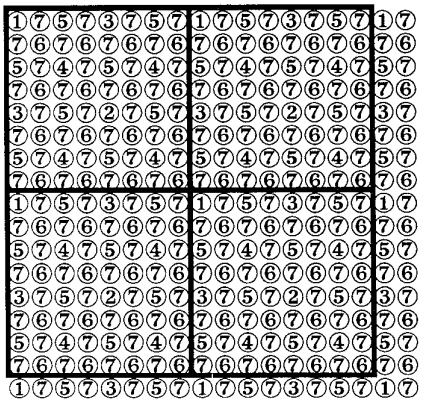

Figure 2: Illustration of a 3-stage decomposition. Samples labeled "1" through "n" are used to interpolate the samples labeled "n+1". We can also group the samples into blocks, as indicated.

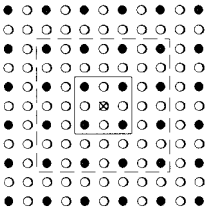

Figure 3: Typical support region of the interpolation filters. Samples marked with • are used to interpolate the sample marked with ⊗.

and Q_n represents the quantization process at the n-th step and Q_n^{-1} is the inverse operation. 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. 2. In this figure, samples labeled "1" through "n" are used to interpolate samples labeled "n+1". Note also that we can group the samples into $2^S \times 2^S$ blocks (as the 8×8 block in the figure) to replace traditional block transforms.

We can characterize the analysis-synthesis process as a pyramidal scheme with critical sampling of the interpolation error, as an association of filter banks, or as a hierarchical DPCM system where samples are predicted by interpolation rather than conventional extrapolation [11, 12]. Actually, if we encode $y_{S,00}(\mathbf{n})$ using a DPCM system we would have a hybrid interpolative-extrapolative prediction system [11].

2.3. Interpolation

The choice of the filters boils down to the choice of an interpolation method. In Fig. 3, samples in the grid marked by • are available to interpolate the sample marked with ⊗. Typical support regions are those indicated by the

solid or dashed lines in Fig. 3 (4 or 16 neighbors). Optimum linear interpolators can be easily computed (assuming the signal characteristics are known). Nevertheless, simple non-linear interpolation has shown to produce better results than much more complex linear filters [13]. Even with the recent theoretical advances in non-linear systems [14, 15, 16], non-linear filters still lack adequate design techniques. Instead of exploring a complex ad-hoc design for the filter, we decided to settle on one of the simplest filters we can think of: a 2×2 median filter. The objective is to show the high potential of non-linear systems. Although simple, we will show that such system can outperform much more complex linear systems. For 4 input samples a_{ij}, we define the median filter by the following rule:

- Given set $\{a_{11}, a_{12}, a_{21}, a_{22}\}$
- Discard $min\{a_{11}, a_{12}, a_{21}, a_{22}\}$
- Discard $max\{a_{11}, a_{12}, a_{21}, a_{22}\}$
- Output the average of the remaining two elements.

See [5] for a discussion on properties of this filtering operation, as well as on its fast implementation algorithm. Such algorithm can be carried using B-bit integer arithmetic for B-bit images and is multiplication free.

3. JPEG-BASED CODING

For evaluation and comparisons in a complete image coding system, we embedded the transform into JPEG. The idea is to replace the DCT coefficients by our pyramid samples. This has been done before by substituting the DCT by the DWT and using the same coder except by replacing the transform [17]. Here, we follow the same principle: using 3 stages ($S = 3$) and grouping the pyramid samples into blocks as shown in Fig. 2. We, therefore, refer to our coder as NLP-JPEG and refer to regular JPEG coder as DCT-JPEG.

S is selected as 3 and $2S+1 = 7$ step sizes are selected for uniform quantizers. The image is transformed using the non-linear pyramid with quantizer feedback. The low-pass samples are encoded using a 2D DPCM as:

$$\tilde{x}_{S,00}(\mathbf{n}) = \frac{1}{2}(\hat{x}_{S,00}(\mathbf{n} - \begin{bmatrix}1\\0\end{bmatrix}) + \hat{x}_{S,00}(\mathbf{n} - \begin{bmatrix}0\\1\end{bmatrix})) \quad (8)$$

$$y_{S,00}(\mathbf{n}) = x_{S,00}(\mathbf{n}) - \tilde{x}_{S,00}(\mathbf{n}) \quad (9)$$

$$\hat{y}_{S,00}(\mathbf{n}) = Q_1^{-1}\{Q_1\{y_{S,00}(\mathbf{n})\}\} \quad (10)$$

$$\hat{x}_{S,00}(\mathbf{n}) = \hat{y}_{S,00}(\mathbf{n}) + \tilde{x}_{S,00}(\mathbf{n}) \quad (11)$$

The transformed samples are grouped into blocks of $2^S \times 2^S = 8 \times 8$ samples as in Fig. 2. 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. 2. The quantized samples are encoded using standard JPEG entropy coding based on Huffman codes.

The DCT-JPEG has 64 quantizer steps (one for each DCT coefficient), while the proposed one has only 7 for 3 stages. A complete description of an algorithm to optimize the quantizer steps can be found in [5]. It is a simplified variation of the method given in [18]. Among the modifications, the quantizer steps were constrained to be non-decreasing because of the recursive nature of the proposed transform. Note that $\Delta_n = 1$ leads to lossless coding.

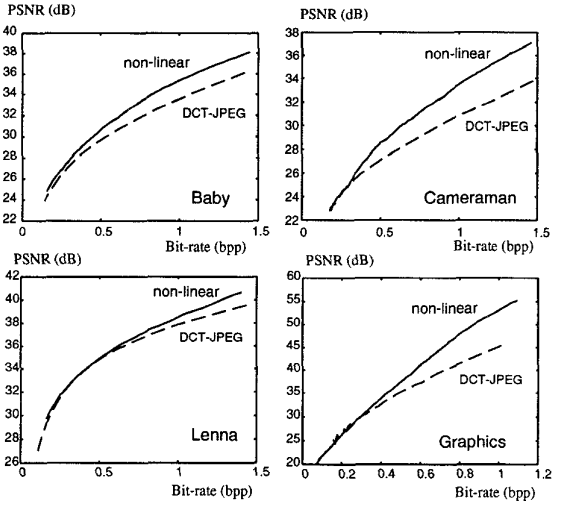

Figure 4: Plot of PSNR versus bit-rate for several images. In both cases optimized Huffman codes were used. Image "Cameraman" has 256×256 pixels while the others have 512×512 pixels.

Tests were carried to compare the performances of NLP- and DCT-JPEG. Fig. 4 shows peak signal-to-noise ratio (PSNR) values for typical images. In these plots we used optimized Huffman codes in JPEG for both the DCT and NLP based schemes. 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 is not accurate to encode texture regions. Images are presented for subjective comparison in Fig. 5.

4. CONCLUSIONS

We presented a PR critically decimated non-linear filter structure for compression applications. The structure used was a two-step filter bank, which is cascaded to produce a pyramid. Image coding tests were carried using JPEG and replacing the DCT by the proposed pyramidal scheme. The proposed scheme shows superior performance over DCT-JPEG both objectively and subjectively. The most appealing feature of the pyramid is its complexity, which is far less complex than most popular linear transforms and is suitable for hardware implementation.

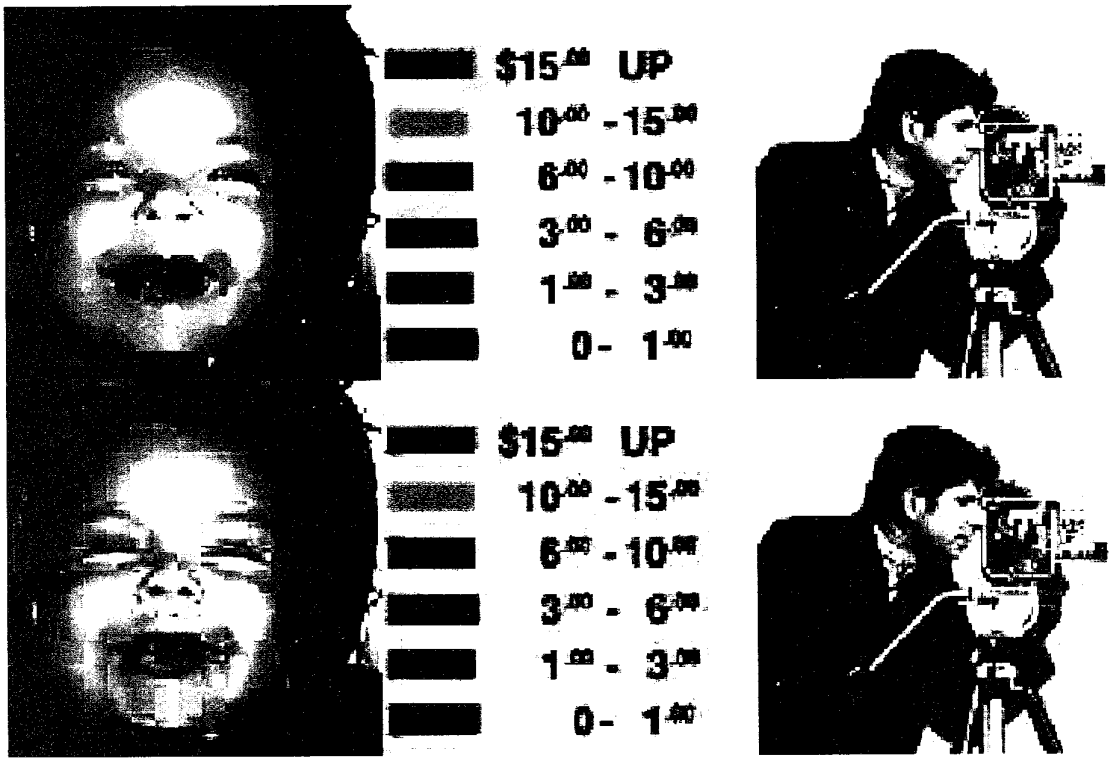

Figure 5: Zoom of reconstructed images. Top: NLP-JPEG. bottom: DCT-JPEG. Image "baby": NLP (30.04dB @ 0.45bpp) DCT (29.22dB @ 0.45bpp). Image "graphics": NLP (35.35dB @ 0.44bpp) DCT (32.70dB @ 0.44bpp). Image "cameraman": NLP (32.48dB @ 0.907bpp) DCT (30.26dB @ 0.914bpp).

5. REFERENCES

[1] P. Burt and J. Adelson, "The Laplacian pyramid as a compact Image Code," *IEEE Trans. Commun.*, vol. 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, NY: Van Nostrand Reinhold, 1993.

[4] D. F. Florêncio and R. W. Schafer, "A non-expansive pyramidal morphological image coder," in *Proc. ICIP*, Vol. 2, pp. 331-334, 1994.

[5] R. L. de Queiroz and D. A. F. Florêncio, "Non-expansive Pyramid For Image Coding Using a Non-Linear Filter Bank," to be submitted to *IEEE Trans. Image Processing*.

[6] Z.Zhou and A.N. Venetsanopoulos, "Morphological Methods in Image Coding," in *Proc. of ICASSP*, pp.481-484, vol. III, 1992.

[7] F.K. Sun and P. Maragos, "Experiments on Image Compression Using Morphological Pyramids," *SPIE VCIP.*, pp. 1303-1312, SPIE vol. 1199, 1989.

[8] A. Toet, "A morphological pyramidal image decomposition," *Pattern Recog. Lett.*, vol. 9, pp. 255-261, May 1989.

[9] O. Egger and Wei Li, "Very low bit rate image coding using morphological operators and adaptive decompositions," in *Proc. ICIP*, Vol. 2, pp. 326-330, 1994.

[10] D. E. Dugeon and R. M. Mersereau, *Multidimensional Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1984.

[11] R. L. de Queiroz and J. T. Yabu-Uti, "On a hybrid predictive-interpolative scheme for reducing processing speed in DPCM TV CODECs," *Proc. of EUSIPCO*, Vol. II, pp. 797-780, Sep. 1990.

[12] R. L. de Queiroz, *Multiresolution Systems for Redundancy Extraction and Progressive transmission in Image Coding*, in Portuguese. Master's Thesis, FEE UNICAMP, Brazil, Nov. 1990.

[13] D. F. Florêncio and R. W. Schafer, "Post-sampling aliasing control for images," in *Proc. of ICASSP*, pp. 893-896, vol. II, 1995.

[14] P. Maragos, "Slope Transforms: theory and application to nonlinear signal processing," *IEEE trans. on Signal Processing*, vol. 43 n. 4, pp. 864-877, Apr 1995.

[15] D. F. Florêncio and R. W. Schafer, "Critical morphological sampling, part 1: binary signals," preprint.

[16] D. F. Florêncio and R. W. Schafer, "Critical morphological sampling, part 2: gray-level signals," preprint.

[17] R. de Queiroz, C. Choi, Y. Huh, J. Hwang, and K. R. Rao, "Wavelet transforms in a JPEG-like image coder," *Proc. SPIE Conf. on VCIP*, SPIE Vol. 2308, pp. 1662–1673, 1994.

[18] S. Wu and A. Gersho, "Rate-constrained picture-adaptive quantization for JPEG baseline coders," *Proc. of ICASSP*, vol V, pp. 389–392, 1993.