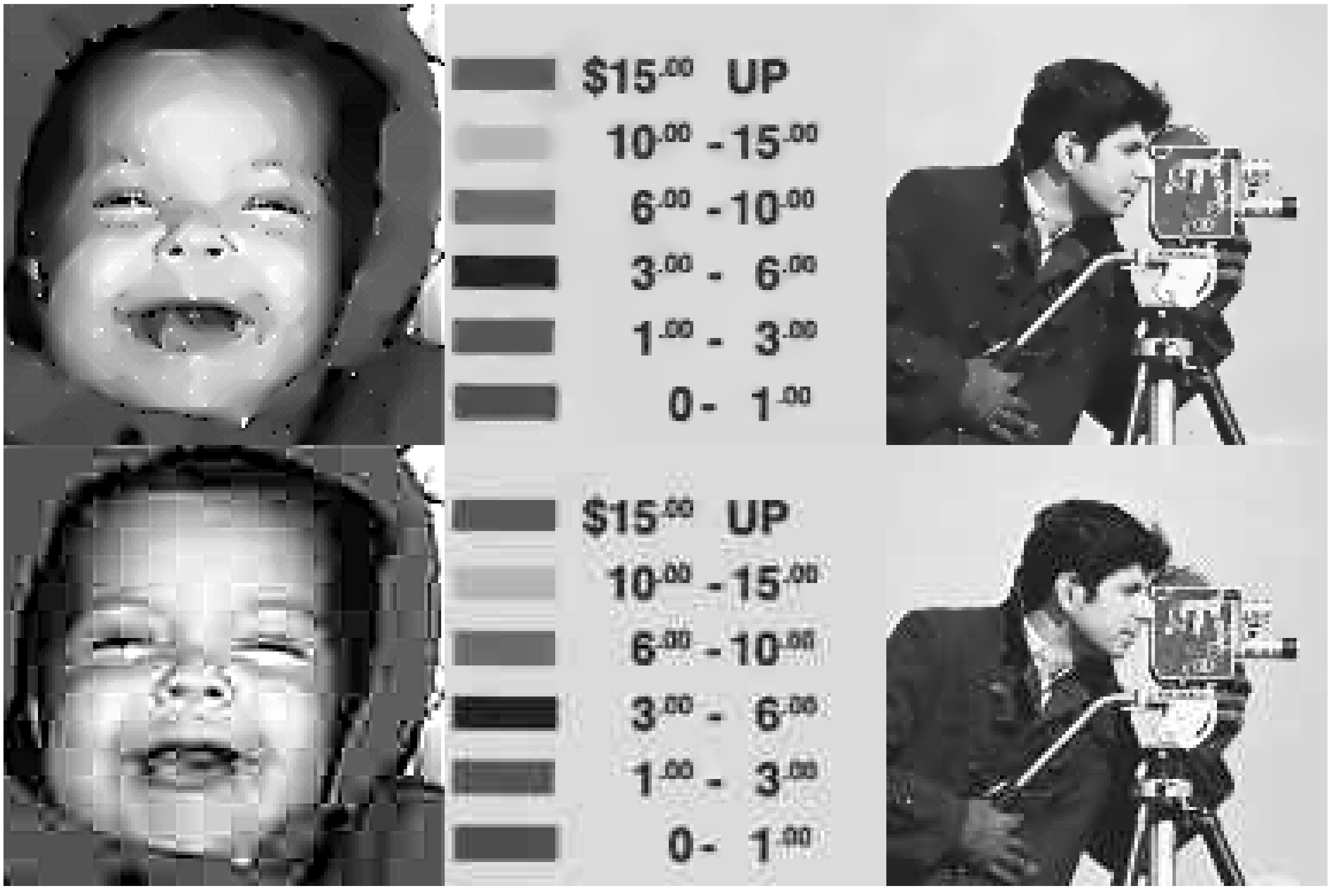


Figure 5: Zoom of reconstructed images. Top: NLP-JPEG, bottom: DCT-JPEG. Image "baby": NLP (30.04dB @ 0.45bpp) DCT (29.22dB @ 0.45bpp). Image "graphics": NLP (35.35dB @ 0.44bpp) DCT (32.70dB @ 0.44bpp). Image "cameraman": NLP (32.48dB @ 0.907bpp) DCT (30.26dB @ 0.914bpp).

[6] F.K. Sun and P. Maragos, "Experiments on image compression using morphological pyramids," *SPIE VCIP,* pp. 1303–1312, SPIE vol. 1199, 1989.

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

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

[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] R. L. de Queiroz, D. A. F. Florêncio, and R. W. Schafer, "Non-expansive pyramid for image coding using a non-linear filter bank," under review, *IEEE Trans. Image Processing.*

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

[12] 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, 1990.

[13] 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.

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

[15] 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.

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

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

[18] 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.

[19] A. Said and W. A. Pearlman, "Reversible image compression via multiresolution representation and predictive coding ," *Proc. SPIE Conf. on VCIP,* SPIE Vol. 2094, pp. 664–674, 1993.

Table 1: Performance (in bpp) of some lossless compressors.

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

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 [18]. Here, we follow the same principle: using three 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 nonlinear pyramid with quantizer feedback. The low-pass samples are encoded using a 2D DPCM as:

$$\tilde{x}_{S,00}(\mathbf{n}) = \frac{1}{2}\left(\hat{x}_{S,00}\left(\mathbf{n} - \left[\begin{smallmatrix}1\\0\end{smallmatrix}\right]\right) + \hat{x}_{S,00}\left(\mathbf{n} - \left[\begin{smallmatrix}0\\1\end{smallmatrix}\right]\right)\right) \quad (5)$$

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

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

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

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. 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 [10]. 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. We compared the performance of the NLP-JPEG for lossless compression against three dedicated lossless coders: (1) the non-DCT lossless JPEG coder; (2) Huffman based Said-Pearlman lossless coder [19]; (3) GnuZIP, which is a regular LZW compressor. Results are shown in Table 1.

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. Resolution is 256 × 256-pels for "Cameraman" 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. 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.

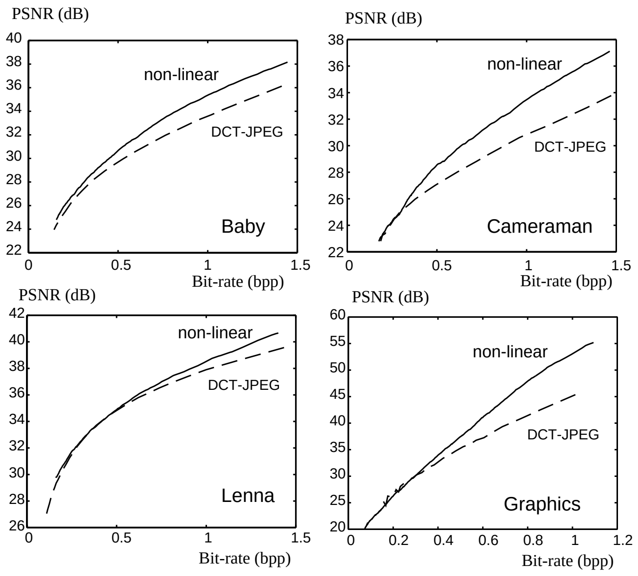


Figure 4: Plot of PSNR versus bit-rate for several images.

4. CONCLUSIONS

We presented a PR critically decimated nonlinear pyramidal structure for image compression based on the cascade of a two-step filter bank. 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. It also outperforms the alternative non-DCT based JPEG algorithm for lossless coding. 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.

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] Dinei A. F. Florêncio and Ronald W. Schafer, "Perfect reconstructing non-linear filter banks," in *Proc. of ICASSP*, 1996.

[5] Z. Zhou and A. N. Venetsanopoulos, "Morphological methods in image coding," in *Proc. of ICASSP*, pp. 481–484, vol. III, 1992.

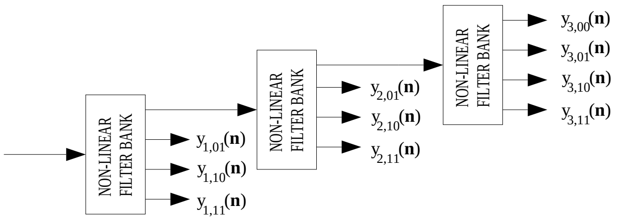


Figure 1: Generation of a multiresolution pyramid.

cess *synthesis*. We further extend the notation to define

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

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.

If we let $process(y_{k,ij}(\mathbf{n}), z(\mathbf{n}), Q_l)$ be

```
    $\tilde{x}_{k,ij}(\mathbf{n}) = \text{interpolate}(z(\mathbf{n}))$
    $y_{k,ij}(\mathbf{n}) = x_{k,ij}(\mathbf{n}) - \tilde{x}_{k,ij}(\mathbf{n})$
    encode $Q_l\{y_{k,ij}(\mathbf{n})\}$
    $\hat{y}_{k,ij}(\mathbf{n}) = Q_l^{-1}\{Q_l\{y_{k,ij}(\mathbf{n})\}\}$
    $\hat{x}_{k,ij}(\mathbf{n}) = \tilde{x}_{k,ij}(\mathbf{n}) + \hat{y}_{k,ij}(\mathbf{n})$
```

the description of the analysis process[1] is given by:

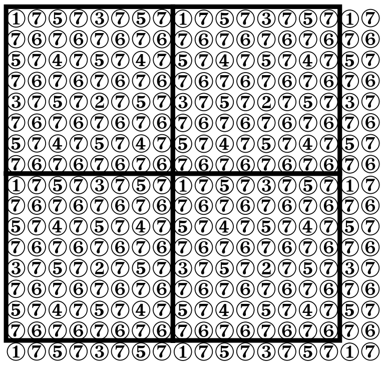
```
$l = 1$
$process(y_{S,00}(\mathbf{n}), , Q_1)$
for $k = S : -1 : 1$
    $process(y_{k,11}(\mathbf{n}), \hat{x}_{k,00}(\mathbf{n}), Q_{l+1})$
    $process(y_{k,01}(\mathbf{n}), \{\hat{x}_{k,00}(\mathbf{n}), \hat{x}_{k,11}(\mathbf{n})\}, Q_{l+2})$
    $process(y_{k,10}(\mathbf{n}), \{\hat{x}_{k,00}(\mathbf{n}), \hat{x}_{k,11}(\mathbf{n})\}, Q_{l+2})$
    $l = l + 2$
end
```

Note that at each iteration

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

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 [12, 13].



①⑦⑤⑦③⑦⑤⑦①⑦⑤⑦③⑦⑤⑦①⑦
⑦⑥⑦⑥⑦⑥⑦⑥⑦⑥⑦⑥⑦⑥⑦⑥⑦⑥
⑤⑦④⑦⑤⑦④⑦⑤⑦④⑦⑤⑦④⑦⑤⑦
⑦⑥⑦⑥⑦⑥⑦⑥⑦⑥⑦⑥⑦⑥⑦⑥⑦⑥
③⑦⑤⑦②⑦⑤⑦③⑦⑤⑦②⑦⑤⑦③⑦
⑦⑥⑦⑥⑦⑥⑦⑥⑦⑥⑦⑥⑦⑥⑦⑥⑦⑥
⑤⑦④⑦⑤⑦④⑦⑤⑦④⑦⑤⑦④⑦⑤⑦
⑦⑥⑦⑥⑦⑥⑦⑥⑦⑥⑦⑥⑦⑥⑦⑥⑦⑥
①⑦⑤⑦③⑦⑤⑦①⑦⑤⑦③⑦⑤⑦①⑦
⑦⑥⑦⑥⑦⑥⑦⑥⑦⑥⑦⑥⑦⑥⑦⑥⑦⑥
⑤⑦④⑦⑤⑦④⑦⑤⑦④⑦⑤⑦④⑦⑤⑦
⑦⑥⑦⑥⑦⑥⑦⑥⑦⑥⑦⑥⑦⑥⑦⑥⑦⑥
③⑦⑤⑦②⑦⑤⑦③⑦⑤⑦②⑦⑤⑦③⑦
⑦⑥⑦⑥⑦⑥⑦⑥⑦⑥⑦⑥⑦⑥⑦⑥⑦⑥
⑤⑦④⑦⑤⑦④⑦⑤⑦④⑦⑤⑦④⑦⑤⑦
⑦⑥⑦⑥⑦⑥⑦⑥⑦⑥⑦⑥⑦⑥⑦⑥⑦⑥
①⑦⑤⑦③⑦⑤⑦①⑦⑤⑦③⑦⑤⑦①⑦

Figure 2: Illustration of a 3-stage decomposition. Samples labeled "n+1" are transformed by computing the interpolation error using the 4 nearest samples labeled "1" through "n". We can also group the samples into blocks, as indicated.

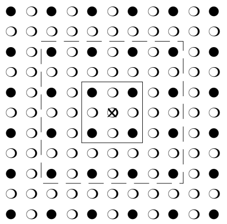


Figure 3: Typical support region of the interpolation filters.

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 use 4 or 16 neighbors. Optimum linear interpolators can be easily computed (assuming the signal characteristics are known). Nevertheless, simple nonlinear interpolation has shown to produce better results than much more complex linear filters [14]. Even with the recent theoretical advances in nonlinear systems [15, 16, 17], nonlinear 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 nonlinear systems. Although simple, we will show that such system can outperform much more complex linear systems. For four 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 [10] 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.

[1] more details can be found in [10].

A PYRAMIDAL CODER USING A NONLINEAR FILTER BANK

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

We propose a novel pyramidal coder, which resembles the general structure of a JPEG coder, but which uses a nonlinear transform to replace the DCT. The nonlinear transform is obtained by the hierarchical application of a median filter predictor at subsampled versions of the original signal. The transformed samples are grouped into square blocks and used to replace the DCT in the JPEG baseline 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. Objective comparisons show the superiority of the proposed coder against both baseline and lossless JPEG.

1. INTRODUCTION

Multiresolution techniques provide a convenient way of exploring the several levels of spatial redundancy existing on most images. The Laplacian pyramid coder [1] explores this idea, and became quite popular for image processing and coding despite the fact that it expands the number of samples. 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 as 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 filter bank, and does not require multiplications, nor 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 outperformed by the proposed nonlinear coder. Besides, since no dedicated lossless mode is required, the coder is also convenient to nearly lossless coding.

Perfect reconstruction (PR) in critically decimated systems is generally guaranteed by imposing conditions on the filter coefficients. When dealing with nonlinear filters, no such general conditions exist [4]. For this reason, nonlinear filter banks were restricted to non-critically decimated cases [5, 6, 7]. Recently, a new approach for critically decimated nonlinear filter banks has been introduced [4, 8, 9], where PR is obtained by imposing restrictions on the filter *structure* instead of on the filter *coefficients*. We use here a particularization of a more general framework [10].

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 [11], 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), and (1,1). The samples in the original signal map to the polyphase components according to the following grid pattern:

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

Applying the same notation to the transformed signal $y(\mathbf{n})$, the decomposition for one pyramid level can be described as:

$$\begin{aligned} y_{00}(\mathbf{n}) &= x_{00}(\mathbf{n}) & (1)\\ y_{11}(\mathbf{n}) &= x_{11}(\mathbf{n}) - F_0(x_{00}(\mathbf{n})) & (2)\\ y_{01,10}(\mathbf{n}) &= x_{01,10}(\mathbf{n}) - F_1(x_{00}(\mathbf{n}), x_{11}(\mathbf{n})) & (3) \end{aligned}$$

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. 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 [10].

2.2. The pyramid

As usual in the filter banks literature, we call the subband decomposition process *analysis* and the reconstruction pro-

In proceedings of ICASSP'96, Atlanta, GA