An improved least squares Laplacian pyramid for image compression

Michael Unser

Biomedical Engineering and Instrumentation Program, Bldg 13, Room 3W13, National Center for Research Resources, National Institutes of Health, Bethesda, MD 20892, USA

Received 20 December 1990
Revised 17 June 1991

Abstract. This paper describes two ways of improving Burt and Adelson's Laplacian pyramid, a technique developed for image compression. The Laplacian pyramid is a multi-resolution image representation that captures the loss of information occurring through repeated reduction of the spatial resolution. The generation of this data structure involves the use of two complementary functions: EXPAND, which increases the size of an image by a factor of 2, and REDUCE, which performs the reverse operation. The first modification is the adjunction of a pre-filter to the initial EXPAND function in order to guarantee an image extrapolation that is an exact interpolation of the coarser resolution level. The second refinement is a REDUCE operation modified to minimize information loss. The corresponding least squares Laplacian pyramid (LSLP) is generated by adding a post-filter to the initial REDUCE function. These new functions have an efficient implementation using recursive algorithms. Preliminary experiments indicate improved performance: for a Gaussian-like kernel ($a=\frac{3}{8}$), the new EXPAND function exhibits a 2 to 2.5 dB attenuation of the first level of the Laplacian pyramid, while the complete scheme (LSLP) leads to a 4.7 to 8.5 dB improvement in the two images used to test the procedure. For comparable compression ratios, the subjective image quality for the LSLP appears to be significantly better. A theoretical relationship between the present approach and the family of quadrature mirror filter image pyramids is also derived.

Zusammenfassung. Diese Arbeit beschreibt zwei Methoden, die Laplacepyramide von Burt und Adelson zu verbessern, welche zur Bildkompression dient. Die Laplacepyramide ist eine Bilddarstellung durch Mehrfachauflösung, welche den Informationsverlust erfasst, der durch die wiederholte Reduktion der räumlichen Auflösung entsteht. Für die Erzeugung dieser Datenstruktur werden zwei komplementäre Funktionen gebraucht: EXPAND, welche das Bild um einen Faktor zwei vergrössert, und REDUCE, welche die inverse Operation durchführt. Die erste Modifikation besteht im Hinzufügen eines Vorfilters zur EXPAND Funktion, um eine Bildextrapolation zu erreichen, welche eine präzise Interpolation des gröberen Auflösungsniveaus ist. Die zweite Verbesserung betrifft die REDUCE Funktion, welche modifiziert wird, um einen minimalen Informationsverlust zu erreichen. Die entsprechende Laplacepyramide kleinster Quadrate (LSLP) wird durch Hinzufügen eines Nachfilters zur REDUCE Funktion erzeugt. Diese zwei neuen Funktionen erlauben eine effiziente Realisierung mithilfe rekursiver Algorithmen. Erste Versuche deuten auf eine deutliche Verbesserung hin: für einen Gauss-artigen Kern ($a=\frac{3}{8}$) erreicht die neue EXPAND Funktion eine Dämpfung von 2 bis 2.5 dB im ersten Niveau der Laplacepyramide, während die vollständige Methode (LSLP) eine Verbesserung von 4.7 bis 8.5 dB für zwei Testbilder erreicht. Für vergleichbare Bildkompressionsfaktoren ist die subjektive Bildqualität von LSLP wesentlich besser. Ferner wird ein theoretischer Zusammenhang zwischen dieser Methode und der Familie Bildpyramiden mit quadratischen Spiegelfiltern hergeleitet.

Résumé. Ce papier décrit deux façons d'améliorer la pyramide Laplacienne de Burt et Adelson, une technique développée pour la compression d'images. La pyramide Laplacienne est une représentation multi-résolution d'image qui code la perte d'information liée à une réduction répétée de la résolution spatiale. La génération de cette structure s'effectue à l'aide de deux opérations complémentaires: EXPAND, qui accroît la taille de l'image par un facteur deux, et REDUCE qui effectue l'opération inverse. La première modification est l'adjonction d'un préfiltre à la fonction EXPAND afin de garantir que l'agrandissement d'une image à partir d'une représentation plus grossière donne lieu à une interpolation exacte. La seconde amélioration est la redéfinition de la fonction REDUCE afin de minimiser la perte d'information. Ceci donne lieu à une représentation pyramidale aux moindres carrés (LSLP) qui diffère de la précédente par la simple adjonction d'un post-filtre. Ces nouvelles fonctions se prêtent à une mise en oeuvre très efficace par filtrage récursif. Des expériences préliminaires indiquent une amélioration des performances: pour un noyau quasi-gaussien ($a=\frac{3}{8}$), la nouvelle fonction EXPAND atténue le premier niveau de la pyramide

Laplacienne de 2 à 2.5 dB, tandis que l'algorithme complet (LSLP) donne lieu à une amélioration de 4.7 à 8.5 dB sur les deux images utilisées afin de tester la procédure. Pour des taux de compression comparables, les images codées avec LSLP sont de qualité subjective supérieure. Finalement, un lien théorique est établi entre l'approche présente et la famille des représentations pyramidales par filtres mirroirs en quadrature.

Keywords. Laplacian pyramid, multi-resolution representation, image compression, image coding, Gaussian pyramid, least squares approximation, recursive filter, quadrature mirror filters, wavelet transform.

1. Introduction

Multi-resolution data representations are becoming increasingly popular in image processing applications. Pyramid data structures, in particular, play an important role in coding, and are ideally suited for progressive image transmission [13, 15]. In these data structures, the image is represented hierarchically with each level corresponding to a reduced-resolution approximation. An example of such a coding scheme is the Laplacian pyramid proposed by Burt and Adelson in which the difference between successive levels of a Gaussian pyramid is transmitted [3]. This approach compares favorably with earlier techniques, such as transform or predictive image coding, especially when large compression ratios are desired [7]. Recent developments in pyramid image compression also include subband coding techniques [18, 20], orthogonal pyramid structures [1, 12] and wavelet transforms [9], which are all based on the concept of quadrature mirror filters (QMF) [4].

The Laplacian pyramid coding technique described by Burt and Adelson relies on the use of two complementary functions: REDUCE and EXPAND. REDUCE computes a lower resolution level of the Gaussian pyramid by decreasing the resolution by a factor of two. EXPAND performs the reverse operation by mapping the coarser level onto a finer sampling grid. These two functions, as defined initially, were sub-optimal in two respects. First, the basic EXPAND function induces some image blurring, tending to increase the energy of the residual image. Second, the initial REDUCE function fails to minimize the loss of information (in the least squares sense) from one level to the next one. It will be shown here that these limitations can be corrected through the appropriate insertion of additional post- and pre-filtering modules. These operators have an infinite impulse response (IIR) and yet can be implemented very efficiently using simple forward and backward recursions, as discussed in Appendix A.

The presentation is organized as follows. Following a series of definitions, a brief review of the Laplacian pyramid coding concept is given in Section 2. A modified EXPAND function that guarantees an exact image interpolation is described in Section 3. The least squares Laplacian pyramid is introduced in Section 4 and the corresponding REDUCE function is derived. The performance improvement of this new approach is illustrated both qualitatively and quantitatively with some experimental results in Section 5. Finally, the present approach is reinterpreted in terms of quadrature mirror filters in order to bring out the relationship with recent subband (or wavelet transform) coding techniques.

2. Burt's Laplacian pyramid

2.1. Notation and operators

The techniques described in this paper are intended for the processing of digital images. However, to simplify the presentation, we have chosen to concentrate on the pyramidal representation of a one-dimensional signal: $\{f(k)\}_{k\in\mathbb{Z}}$. All subsequent results carry over directly to higher dimensions if one makes use of separable filtering kernels. In practice, for digital images, this means that a pyramid representation can be obtained from the successive application of one-dimensional operators along the rows and columns.

There are two operations that are particularly useful for our purpose: the up-sampling of a signal by an integer multiple m (in particular, $m=2$), which is defined as

$$[f]_{\uparrow m}(k)=\begin{cases} f(k') & \text{for } k=mk', \\ 0 & \text{otherwise.}\end{cases} \tag{2.1}$$

and its dual, the decimation by an integer factor m,

$$[f]_{\downarrow m}(k)=f(km). \tag{2.2}$$

We will rely heavily on the z-transform representation of a signal, which, as a reminder, is defined as

$$f(k)\leftrightarrow F(z)=\sum_{k=-\infty}^{+\infty} f(k)z^{-k}. \tag{2.3}$$

In particular, Burt's generating kernel [3] is central to the construction of the Gaussian or Laplacian pyramid and is conveniently represented as

$$w_2(k)\leftrightarrow W_2(z)=(\tfrac{1}{2}-a)(z^{-2}+z^2) + \tfrac{1}{2}(z^{-1}+z)+2a. \tag{2.4}$$

This operator is symmetric and has a sum equal to two, independent of a. The decimated version of this kernel is

$$w_1(k)=[w_2]_{\downarrow 2}(k) \leftrightarrow W_1(z)=(\tfrac{1}{2}-a)(z^{-1}+z)+2a, \tag{2.5}$$

and has a sum equal to one.

2.2. Basic REDUCE and EXPAND functions

The Gaussian pyramid is a multi-resolution representation of a signal. It is characterized by a sequence of signals $f_0, f_1, \ldots, f_n$, with the number of samples reduced by a factor of two in each of the principal directions from one level to the next. The finer or zero level of the pyramid is given by the initial data sequence

$$f_0(k)=f(k), \tag{2.6}$$

and the successive coarser resolution levels are constructed iteratively using the REDUCE operator

$$f_{i+1}(k)=\text{REDUCE}[f_i](k). \tag{2.7}$$

This operation requires some form of lowpass filtering and decimation by a factor of 2. Two examples of pyramid representations are shown in Fig. 1. Burt and Adelson [3] use a 5-point quasi-Gaussian pre-filter and their REDUCE function can be described as

$$f_{i+1}(k)=\tfrac{1}{2}[w_2 * f_i]_{\downarrow 2}(k), \tag{2.8}$$

where the generating kernel w_2 is defined by (2.4).

The complement of REDUCE is the EXPAND function, which performs a signal extrapolation to a finer resolution level,

$$\hat{f}_{i,i+1}(k)=\text{EXPAND}[f_{i+1}](k). \tag{2.9}$$

This operation involves an up-sampling by a factor of two and some form of interpolation. Burt and Adelson use the following operator [3]:

$$\hat{f}_{i,i+1}(k)=w_2 * [f_{i+1}]_{\uparrow 2}(k). \tag{2.10}$$

These two procedures are summarized in Figs. 2(a) and 2(b).

2.3. The Laplacian coding concept

The Laplacian pyramid captures the loss of information resulting from an application of the REDUCE function and is the difference between two successive levels of the Gaussian pyramid:

$$\Delta f_i(k)=f_{i-1}(k)-\text{EXPAND}[f_i](k). \tag{2.11}$$

The key idea in the Laplacian coding scheme is to transmit the sequence of difference images $\Delta f_1, \ldots, \Delta f_n$ with sample values less extensively correlated than are the initial image pixels. The original image is then recovered by progressively expanding and summing the levels of the Laplacian pyramid, starting at the coarsest level. The main advantage of this approach is that the entropy of

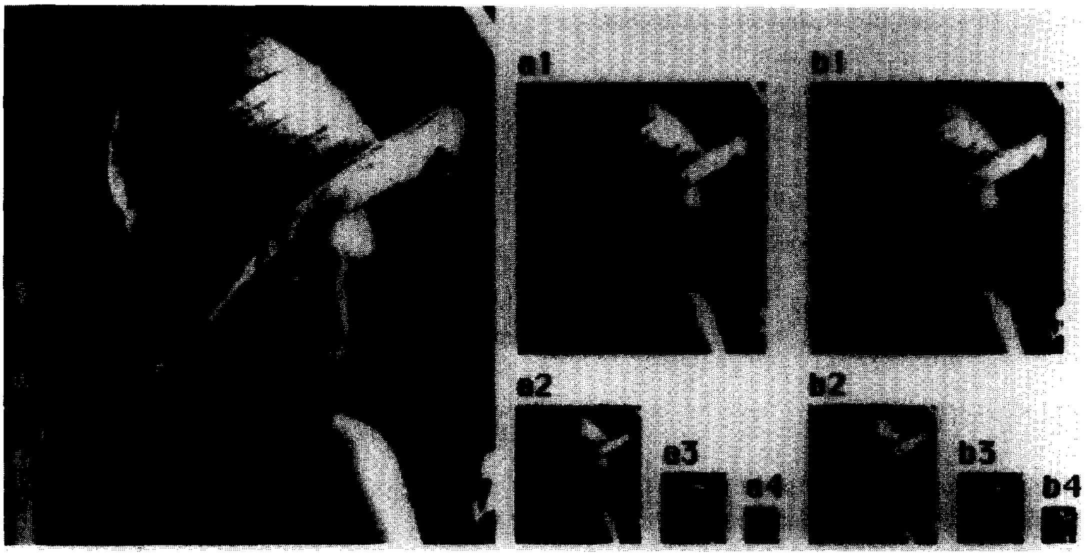

Fig. 1. Examples of image pyramids. (0): 208 × 222 'female' image (level 0), (a1–a4): levels 1 to 4 of the basic Gaussian pyramid, (b1–b4): levels 1 to 4 of the least squares Gaussian pyramid.

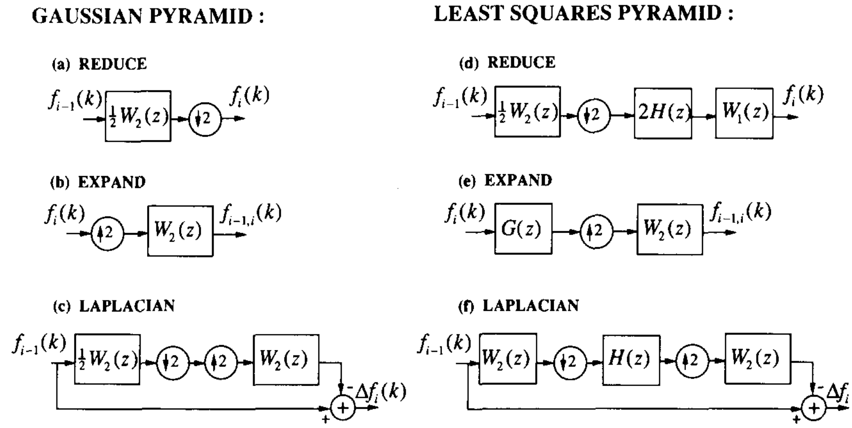

Fig. 2. Schematic representation of the REDUCE, EXPAND and LAPLACIAN functions for the standard and least squares Gaussian pyramids.

the difference images is usually smaller than that of the initial image. Thus, the amount of transmitted information can be reduced by source coding.

If one is willing to accept some image degradation, a substantial improvement of performance can be further achieved through quantization. Burt and Adelson have shown that the degradation can be made almost imperceptible through a proper choice of the number of quantization levels. The scheme they propose uses more quantization steps for coarser levels of the pyramid. The sample values at coarser spatial resolutions have to be coded more carefully because their contribution affects a larger number of pixels in the final reconstructed image.

Although this approach achieves excellent image coding performance, we have evidence that it can be further improved. The reason for this is that Burt's construction of the Laplacian pyramid is sub-optimal by several criteria:

(i) The EXPAND function as defined by (2.8) does not achieve a true image interpolation; that is, EXPAND$[f_i]_{\downarrow 2}$ as defined by Burt will generally differ from f_i.

(ii) The variances of the Laplacian images are not necessarily minimal.

(iii) The REDUCE and EXPAND functions as defined by (2.8) and (2.10) are not strictly complementary in that EXPAND so defined is not the generalized inverse of REDUCE:

$$\text{REDUCE}[f_i](k) \neq \text{REDUCE}[\text{EXPAND}[\text{REDUCE}[f_i]]](k).$$

The following sections will show how these criteria can be taken into account.

3. The Laplacian pyramid with interpolation

The major limitation of the method proposed by Burt et al. is that the EXPAND function defined by (2.10) does not produce a valid image interpolation in the sense that the pixel values at the nodes are not preserved when a coarser level is used to approximate the next finer level. In fact, $\hat{f}_{i,i+1}(k)$ is a smoothed extrapolation of $f_{i+1}(k)$ and the energy of the difference signal is therefore unnecessarily large. We have defined a modified EXPAND function that guarantees strict signal interpolation in the sense defined above. This constraint is formally expressed as

$$[\hat{f}_{i,i+1}]_{\downarrow 2}(k) = f_{i+1}(k). \tag{3.1}$$

This condition can be satisfied by applying the previous EXPAND operation to an auxiliary sequence $\{p_{i+1}(k)\}$:

$$\hat{f}_{i,i+1}(k) = [p_{i+1}]_{\uparrow 2} * w_2(k) = \sum_{l=-\infty}^{+\infty} p_{i+1}(l) w_2(k-2l), \tag{3.2}$$

chosen to satisfy the constraint

$$f_{i+1}(k) = p_{i+1} * w_1(k) = \sum_{l=-\infty}^{+\infty} p_{i+1}(l) w_1(k-l). \tag{3.3}$$

It is straightforward to verify that (3.2) and (3.3) are mutually consistent with (3.1). Equation (3.3) is an easily invertible constant coefficient difference equation. In fact, p_{i+1} can be determined by convolution,

$$p_{i+1}(k) = g * f_{i+1}(k), \tag{3.4}$$

where g is the inverse filter given by

$$g(k) \leftrightarrow G(z) = W_1(z)^{-1} = \frac{1}{(\frac{1}{2}-a)(z^{-1}+z)+2a}. \tag{3.5}$$

The poles of this filter are

$$z_{1,2} = \frac{-2a \pm \sqrt{4a-1}}{1-2a} \tag{3.6}$$

and form a reciprocal pair. For $a \geqslant \frac{1}{4}$, these poles are real and the system can be decomposed as a cascade or a sum of causal and anti-causal simple exponential filters. For $a = \frac{1}{2}$, $G(z) = 1$ and an exact interpolation can be achieved with no filtering at all. Otherwise, this operator can be implemented recursively with as few as two adds and three multiplies per sample point, as shown in Appendix A (see Table A). As the signals encountered in practice are of finite extent (e.g. $\{f(k) \mid k = 1, \ldots, K\}$), we have chosen to implement both finite and infinite impulse response filters using the following boundary conditions:

$$\begin{aligned} f(K+k) &= f(K-k), \\ f(1-k) &= f(1+k), \end{aligned} \quad k = 1, \ldots, K. \tag{3.7}$$

This type of signal extrapolation using mirror symmetry is commonly used in image processing applications and has the advantage of suppressing border artifacts.

Our modified EXPAND operator is represented schematically in Fig. 2(e) and is described formally as

$$\hat{f}_{i,i+1}(k) = w_2 * [g * f_{i+1}]_{\uparrow 2}(k). \tag{3.8}$$

It differs from (2.10) only by the adjunction of a pre-filter (g).

4. The least squares pyramid

A further refinement is to choose a compression scheme that minimizes the energy of the Laplacian. For this purpose, it is convenient to use the auxiliary coefficient sequence $\{p_i(k)\}$ defined earlier and to express the Laplacian as

$$\Delta f_i(k) = f_{i-1}(k) - w_2 * [p_i]_{\uparrow 2}(k). \qquad (4.1)$$

We now seek the series of coefficients $\{p_i(k)\}$ that minimizes the error criterion,

$$\varepsilon^2 = \sum_{k=-\infty}^{+\infty} (\Delta f_i(k))^2 = \sum_{k=-\infty}^{+\infty} (f_{i-1}(k) - w_2 * [p_i]_{\uparrow 2}(k))^2. \qquad (4.2)$$

As demonstrated in Appendix B, the optimal sequence of coefficients $p_i(k)$ satisfies the following equation:

$$[w_2 * w_2]_{\downarrow 2} * p_i(k) = [w_{22}]_{\downarrow 2} * p_i(k) = [w_2 * f_{i-1}]_{\downarrow 2}(k). \qquad (4.3)$$

The solution is determined by first convolving f_{i-1} with w_2, performing a decimation by a factor two, and finally filtering the resulting sequence with the operator h that implements the inverse of $[w_{22}]_{\downarrow 2}$:

$$p_i(k) = h * [w_2 * f_{i-1}]_{\downarrow 2}(k). \qquad (4.4)$$

By determining $[w_{22}]_{\downarrow 2}(k)$ explicitly, the corresponding IIR filter is characterized in the z-transform domain,

$$h(k) \leftrightarrow H(z) = [(\tfrac{1}{4} - a + a^2)(z^2 + z^{-2}) + (\tfrac{1}{4} + 2a - 4a^2)(z + z^{-1}) + 1 - 2a + 6a^2]^{-1}. \qquad (4.5)$$

The poles of this operator are simple and real for $\frac{1}{4} \leqslant a \leqslant \frac{1}{2}$. As shown in Appendix A, h can be implemented recursively with as few as five multiplications and four additions per sample point. The relevant filter parameters for different values of a are given in Table A. By substituting (4.4) in (4.1), we find that the least squares Laplacian is given by

$$\Delta f_i(k) = f_{i-1}(k) - w_2 * [h * [w_2 * f_{i-1}]_{\downarrow 2}]_{\uparrow 2}(k). \qquad (4.6)$$

Similarly, the corresponding REDUCE function is obtained by substitution of (4.4) in (3.3),

$$f_{i+1}(k) = w_1 * h * [w_2 * f_i]_{\downarrow 2}(k), \qquad (4.7)$$

and differs from (2.8) by the inclusion of two additional levels of post-filtering provided by w_1 and h. By recalling that $g * w_1(k) = \delta(k)$, we note that (4.6) is fully compatible with both (2.11) and the modified EXPAND function defined by (3.8). However, a direct evaluation of the least squares Laplacian through (4.6) is preferable for most practical purposes. It is more economical and also reduces the propagation of roundoff errors.

These results are summarized in Fig. 2 which provides a block diagram representation of the EXPAND, REDUCE and LAPLACIAN functions and a comparison of the conventional and least squares Laplacian pyramids.

5. Results

5.1. Experiments

The experiments were performed with $a = \frac{3}{8}$, unless indicated otherwise. In these comparisons, the three following procedures were considered: (i) the initial Laplacian pyramid (LP) based on (2.8) and (2.10), (ii) the Laplacian pyramid with interpolation (LPI) based on (2.8) and (3.8), and (iii) the least squares Laplacian pyramid (LSLP) based on (4.6), (4.7) and (3.8).

The Gaussian and least squares pyramidal representations for two test images are shown in Figs. 1 and 3, respectively. In both cases, the sharpness of the least squares pyramid is preserved at all resolution levels, while the corresponding images in the Gaussian pyramid seem increasingly blurred by comparison. The distinction between the two methods is even more striking if one looks at the Laplacian images displayed in Figs. 4 and 5. The same intensity scaling factors were applied to all images

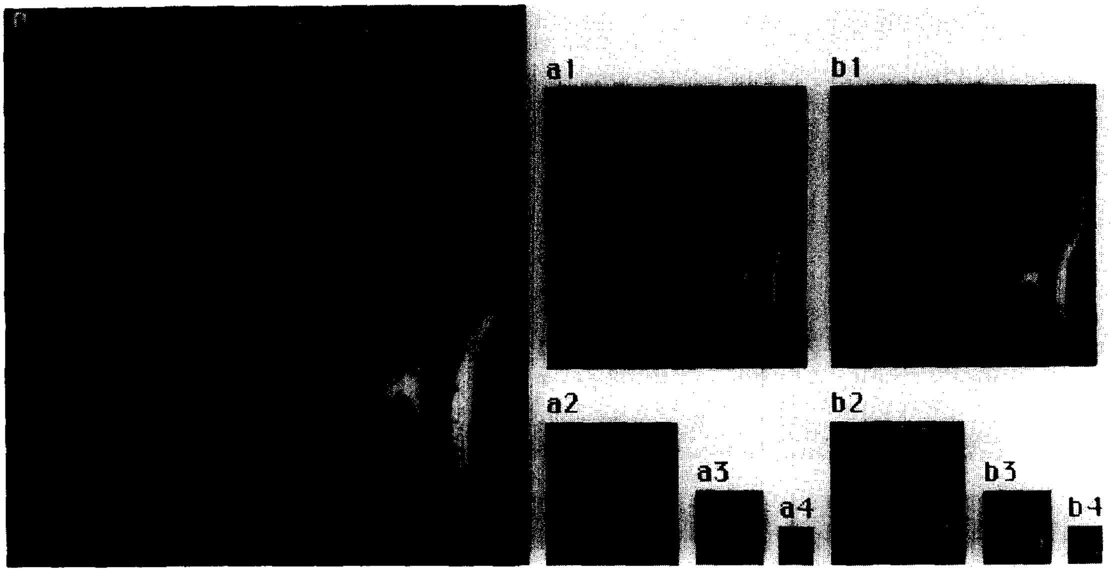

Fig. 3. Comparison of image pyramids. (0): 238 × 253 'MRI' image (level 0), (a1–a4): levels 1 to 4 of the basic Gaussian pyramid, (b1–b4): levels 1 to 4 of the least squares Gaussian pyramid.

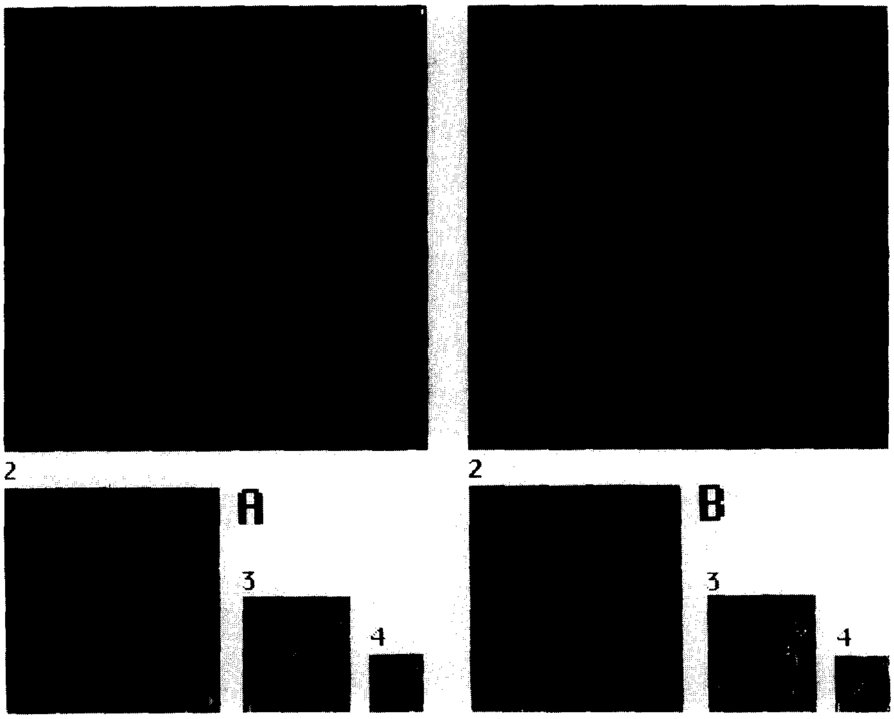

Fig. 4. Comparison of Laplacian pyramids for the 'female' image. (A1–4): four bottom levels of the basic Laplacian pyramid, (B1–4): four bottom levels of the least squares Laplacian pyramid.

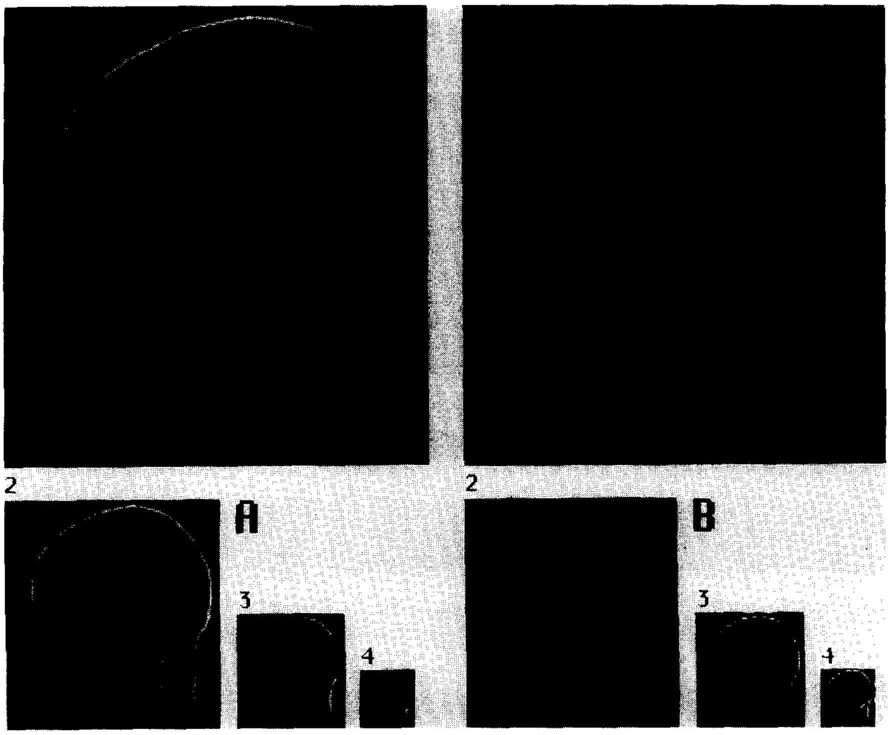

Fig. 5. Comparison of Laplacian pyramids for the 'MRI' image. (A1–4): four bottom levels of the basic Laplacian pyramid, (B1–4): four bottom levels of the least squares Laplacian pyramid.

to facilitate the comparison. For the initial LP, the amount of information at each level is quite significant and the initial subject is still recognizable. In the case of the LSLP, the energy of the Laplacian is reduced drastically and only very high frequency details are visible in this representation.

In a first stage, the performance of the decomposition can be assessed in terms of simple statistics such as the entropy, the standard deviation or root mean square (RMS) error, and the minimum and maximum values of the Laplacian images. Another useful indicator is the signal-to-noise-ratio associated with a full scale expansion of a lower resolution level f_i,

$$\mathrm{SNR}_i = 10\log_{10}\left(\frac{\sum_{k,l}[f_0(k,l)-\bar{f}]^2}{\sum_{k,l}[f_0(k,l)-\hat{f}_{0,i}(k,l)]^2}\right), \qquad (5.1)$$

where $\bar{f}$ denotes the image's average intensity value. These measures are given in Tables 1 and 2 for test images (a) and (b), respectively. For $a=\frac{3}{8}$, the LPI is superior to the basic LP in all respects (e.g.,

Table 1

Comparison of performance measures at successive pyramid levels for the 'female' image with $a=\frac{3}{8}$

Pyramid level	Range	RMS	Entropy	SNR (dB)
LP-1	(−80, 85)	11.75	5.10	23.70
LP-2	(−69, 64)	12.27	5.39	19.44
LP-3	(−60, 52)	14.98	5.85	16.48
LP-4	(−53, 56)	17.50	6.09	—
LPI-1	(−67, 82)	9.71	4.77	25.35
LPI-2	(−60, 53)	9.55	4.95	20.70
LPI-3	(−47, 44)	11.67	5.45	17.51
LPI-4	(−50, 45)	13.55	5.70	—
LSLP-1	(−60, 81)	6.81	4.36	28.43
LSLP-2	(−79, 76)	12.36	5.35	22.96
LSLP-3	(−79, 116)	16.91	5.96	19.48
LSLP-4	(−89, 83)	23.92	6.42	—

Table 2

Comparison of performance measures at successive pyramid levels for the 'MRI' image with $a=\frac{3}{8}$

Pyramid level	Range	RMS	Entropy	SNR (dB)
LP-1	(−76, 101)	11.99	4.83	26.55
LP-2	(−65, 99)	15.70	5.26	20.95
LP-3	(−61, 94)	17.47	5.57	18.01
LP-4	(−39, 82)	14.60	5.58	—
LPI-1	(−71, 91)	9.77	4.59	29.15
LPI-2	(−52, 79)	12.38	4.92	22.44
LPI-3	(−51, 73)	13.72	5.22	18.87
LPI-4	(−33, 63)	11.36	5.21	—
LSLP-1	(−55, 60)	4.50	3.63	35.05
LSLP-2	(−101, 111)	14.31	5.33	25.48
LSLP-3	(−101, 133)	22.50	6.11	20.78
LSLP-4	(−108, 127)	28.36	6.38	—

reduced range, smaller standard deviation and entropy, and better signal approximation). As expected, the LSLP provides an even better signal approximation. In fact, the SNR values obtained for an LSLP extrapolation at a given level i are comparable to those obtained for an LP extrapolation at level $i-1$ with four times more sample values. The improvement of the LSLP is particularly striking at the finer resolution level at which the residual RMS error is approximately reduced by a factor of 2. Note, however, that this effect is reversed for the coarser levels and that the LSLP has the tendency to pack the energy into the top of the pyramid. In terms of image coding, this means that while fewer bits are required for representing the finer levels of the LSLP, more bits will be necessary for coding the coarser levels, a result consistent with the bit allocation strategy used by Burt and Adelson.

For lossless image coding, the number of bits per pixel (bit-rate) necessary to transmit the top of the pyramid up to level i is approximately

$$B_i=\sum_{k=i}^{n} H_i/4^{i-1}, \tag{5.2}$$

where H_i denotes the entropy at the ith level of the Laplacian pyramid. A convenient way to assess the performance in a progressive data transmission scheme is to evaluate the associated mean square distortion in terms of percentage,

$$D_i=100\frac{\sum_{k,l}[f_0(k,l)-\hat{f}_{0,i}(k,l)]^2}{\sum_{k,l}[f_0(k,l)-\bar{f}]^2}. \tag{5.3}$$

The rate-distortion curves for our test images are given in Fig. 6. For both images, the LSLP achieves the best performance at all resolution levels. The LP is the worst and the LPI is in between.

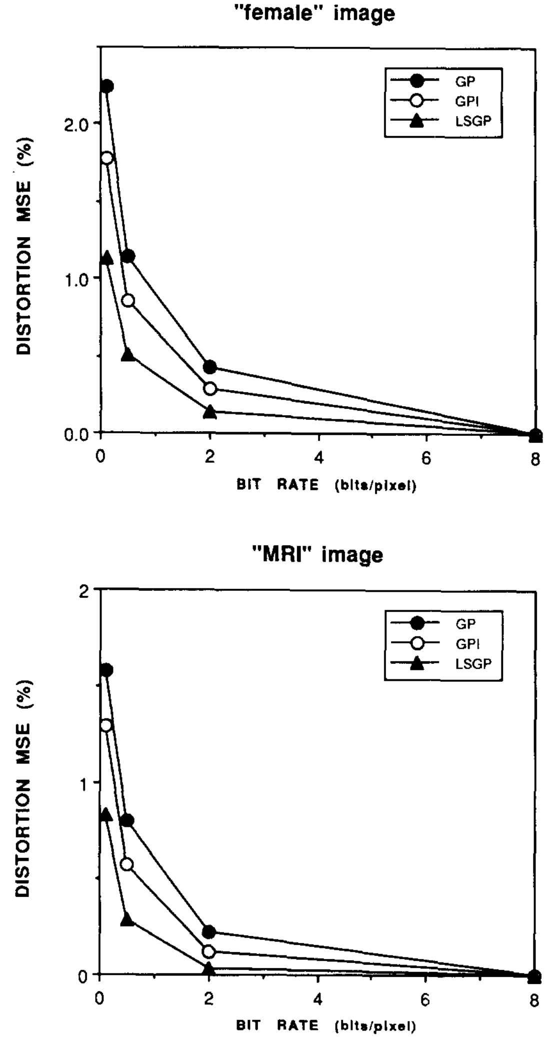

Fig. 6. Rate distortion curves for 'female' and 'MRI' images for the three image pyramids: Gaussian pyramid (GP), Gaussian pyramid with interpolation (GPI) and least squares Gaussian pyramid (LSGP).

An aspect that must also be taken into account in this comparison is that the performance of the pyramid decomposition depends on the value of the parameter a. In principle, our modified scheme should result in some improvement for any value of this parameter, although this effect may not always be as dramatic as in the examples discussed above. A case of special interest occurs when $a=0.5$ in which case the LP is equivalent to the LPI (i.e. $g(k)=$identity). The corresponding error statistics for the MRI image are given in Table 3. The performances of the LPI are slightly superior to those obtained with $a=\frac{3}{8}$. The LSLP performs best but the improvement is not as dramatic as in Table 2. For comparison, we have also included the results for the LP with $a\cong 0.6$, the parameter value that resulted in the greatest reduction in entropy and variance in the series of experiments reported by Burt and Adelson [3]. The improvement over the LP with $a=\frac{3}{8}$ is substantial, emphasizing the importance of the optimization of this parameter. Despite these excellent results, the optimized LP is still less performant than the LSLP which provides its best results for $a=\frac{3}{8}$.

The Laplacian pyramid coding scheme proposed by Burt and Adelson is especially suited for lossy image transmission [3]. The quantization scheme that they propose uses fewer bins for the higher resolution levels of the pyramid, which takes into account the fact that human contrast sensitivity decreases with high spatial frequencies. We have conducted some preliminary experiments to compare the efficiency of the different pyramid representations for this type of image coding. The experimental procedure is similar to the one used in [3] with some minor differences. The important features of the present compression algorithm are as follows:

(i) The coding and the decoding are performed in parallel starting at the coarsest level of the pyramid. In the present case, the pyramid has three levels and the coarsest (f_3) is coded precisely using all eight bits per node (256 gray level values). The corresponding contribution to the total bit-rate is only $\frac{8}{64}=0.125$ bits/pixel.

(ii) A Laplacian image is computed from the difference between a particular level of the Gaussian pyramid and the expanded version of the encoded image one level coarser. This technique takes into account quantization errors introduced at coarser resolution levels.

(iii) The number of levels for each Laplacian image is fixed and should be determined using psychophysical information. The values of these levels are determined using a discrete form of the Max minimum error quantization algorithm [11] applied to the histogram of the

Table 3

Comparison of performance measures at successive pyramid levels for the 'MRI' image with $a=0.5$ and $a=0.6$

Pyramid level	Range	RMS	Entropy	SNR (dB)
LP-1 ($a=0.6$)	(−71, 91)	7.09	4.06	31.11
LP-2 ($a=0.6$)	(−97, 124)	13.74	4.92	24.15
LP-3 ($a=0.6$)	(−81, 133)	18.69	5.49	20.07
LP-4 ($a=0.6$)	(−69, 120)	19.10	5.67	—
LP or LPI-1 ($a=0.5$)	(−73, 95)	8.54	4.31	29.49
LP or LPI-2 ($a=0.5$)	(−79, 111)	13.75	4.92	22.78
LP or LPI-3 ($a=0.5$)	(−60, 103)	16.82	5.32	19.15
LP or LPI-4 ($a=0.5$)	(−45, 93)	15.53	5.38	—
LSLP-1 ($a=0.5$)	(−65, 79)	5.93	3.89	32.65
LSLP-2 ($a=0.5$)	(−120, 145)	16.84	5.12	24.93
LSLP-3 ($a=0.5$)	(−125, 190)	27.84	5.71	20.50
LSLP-4 ($a=0.5$)	(−129, 196)	33.73	6.01	—

images. The corresponding quantization levels are selected to minimize the approximation error and are not necessarily equidistant as was the case in the approach chosen by Burt and Adelson. In this series of experiments, the Laplacian images 2 and 3 were represented by 5 and 15 levels, respectively. The finest level of the pyramid was either not transmitted at all to achieve bit-rates lower than 1 bit/pixel or represented by 3 levels.

(iv) The effective bit-rates are estimated from the entropies of the quantized images using (5.2). These estimates are somewhat optimistic as they ignore the transmission of the code book information. A practical approach to this problem is to summarize this information in terms of the coefficients of a parametric model of the Laplacian histogram (for example, the two parameters α and β of a generalized exponential model $p(\Delta f)=C_0\,\mathrm{e}^{-(|\Delta f|/\alpha)^\beta}$). These parameters can then be used to determine uniquely the optimal quantization levels in the Lloyd–Max scheme and their corresponding code words in a variable length Huffman code [6].

Some examples of image coding with bit-rates as low as 0.7 bits/pixel are shown in Fig. 7. The same number of quantization levels were used in all cases with the exception of Fig. 7(d). This latter image is an improvement of Fig. 7(c) obtained by adding a finer level of the LSLP quantized with three levels; it is visually indistinguishable from the original. The image obtained using LP (Fig. 7(a)) appears to be out of focus and is of lesser quality (both qualitatively and quantitatively) than the results obtained with the LPI and LSLP. The LSLP scheme is clearly superior and appears to preserve most of the image details. The same qualitative behavior has also been observed for different compression ratios and test images. For the test image in Fig. 7, we have also observed that the quality of the LP reconstruction is noticeably degraded for bit-rates lower than 1.5 bits/pixel, while for the LSLP greater compression ratios still produce acceptable results, as illustrated by Fig. 7(c). In these preliminary experiments, the performance of LSLP appears to be consistently superior.

5.2. Discussion

Our experimental results show that both the LPI and LSLP should be superior to the standard LP proposed by Burt et al. Two types of improvements have been considered and both seem to be equally helpful, at least for $a<\frac{1}{2}$. The first is the requirement that an image extrapolation be a true interpolation of a lower level approximation. A simple way to enforce this constraint is to add a pre-filter $(g(k))$ to the basic EXPAND operation. The effect of this operator is less significant when a is close to $\frac{1}{2}$, in which case the LP and LPI are essentially equivalent. The second is to minimize the amount of transmitted information. The only adjunction here is a post-filter following the basic REDUCE operation. The LSLP incorporates both of these mechanisms and has surprisingly good compression properties. This approach provides an attractive alternative to the standard LP and should allow greater efficiency in image coding. Since multi-resolution techniques are being used increasingly in image processing, there are many other potential applications including image segmentation [2, 16], edge detection [10], feature extraction and a variety of multi-grid algorithms for computer vision [14].

The experimental results presented in Section 5.1 indicate a performance improvement in a lossless progressive data transmission scheme (cf. Fig. 6). The reduction of the RMS error also suggests that the LSLP should result in some improvement for lossy image coding as confirmed by our preliminary experiments (cf. Fig. 7). These results, however, are still preliminary and require further investigation. For instance, it seems important to determine an optimal bit allocation strategy for a given compression ratio and to compare the coding results for a variety of test images using objective psychovisual criteria. A detailed evaluation of the dependence of the relative performance of the algorithms on the parameter a may also be appropriate.

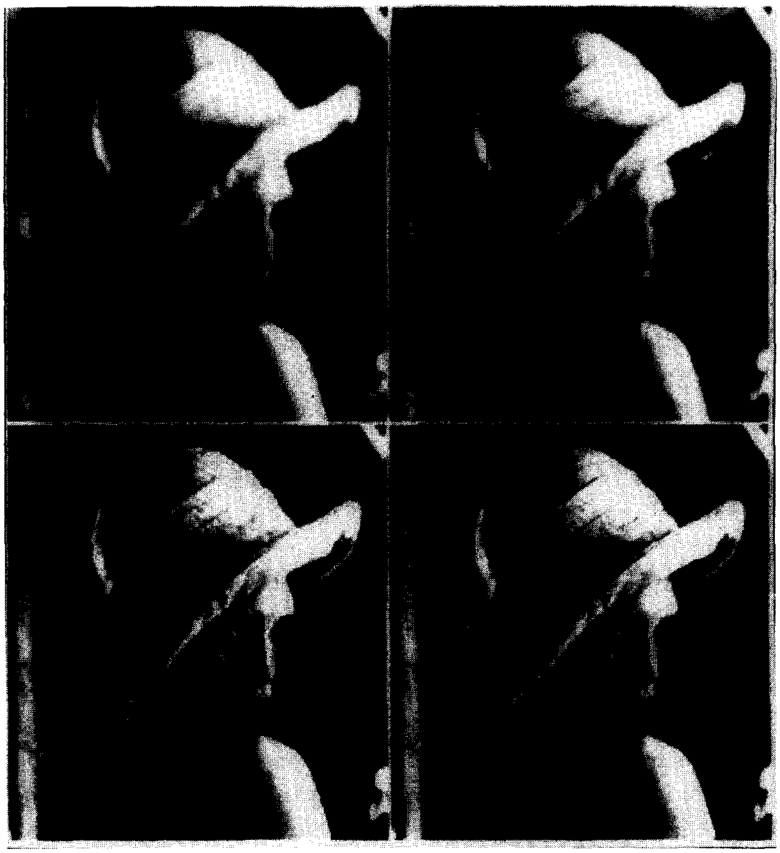

Fig. 7. Comparison of lossy image compression schemes. (a) Laplacian pyramid (LP): $B=0.746$ bits/pixel, SNR = 23.02 dB; (b) Laplacian pyramid with interpolation (LPI): $B=0.708$ bits/pixel, SNR = 24.51 dB; (c) least squares Laplacian pyramid (LSLP): $B=0.702$ bits/pixel, SNR = 27.15 dB; (d) LSLP: $B=1.667$ bits/pixel, SNR = 31.90 dB.

As described in Appendix A, the additional pre- and post-filters can be implemented very efficiently and the increase in computation is negligible. For instance, the CPU times (standard 16 MHz Apple Macintosh IIcx) required to compute the first level of the Laplacian of a 256 × 256 image using Burt's LP, the LSI and LSLP are 18 s, 25 s and 27 s, respectively. The complexity of the LSI and LSLP are comparable because the use of the interpolation pre-filter can be avoided in the second scheme (cf. Fig. 2(f)).

The value $a=\frac{3}{8}=0.375$ was used for most of our experiments. It is close to the value 0.36 recommended by Burt for the greatest reduction of the side lobes of the transfer function [2]. Note that $a=\frac{3}{8}$ corresponds to an implicit choice of a quadratic B-spline interpolator [17]. In terms of performance, this value of a seems to be preferable over others (cf. Tables 2 and 3), largely because of the smoothness and Gaussian-like shape of the corresponding interpolation kernel, which appears to be most appropriate for a large class of images. The unmodified LP, on the other hand, seems to perform best for $a=0.6$ [3]. An explanation for this observation is that the corresponding correction filters in our modified scheme have a very fast decay (i.e., $g(k)=O(z_1^{|k|})$ and $h(k)=O(\rho^{|k|})$, where $z_1=0.084$ and $\rho=0.074$), and can be relatively well approximated by an identity filter. Another value of interest is $a=\frac{1}{2}$. This value leads to a triangular interpolation function and corresponds to image reconstruction by piecewise linear (or bilinear) interpolation. This scheme is equivalent to a first-order spline interpolation.

Clearly, the theory presented here is not restricted to the particular form of interpolation function given by (2.4). It is straightforward to adapt these results to any given kernel $w(k) \leftrightarrow W(z)$. The only constraint is the stability of the approximation and interpolation filters, which, in the general case, are given by

$$H(z)=\frac{2}{W(z^{1/2})W(z^{-1/2})+W(-z^{1/2})W(-z^{-1/2})}, \tag{5.4}$$

$$G(z)=\frac{2}{W(z)+W(-z)}. \tag{5.5}$$

There is another advantage for the use of the least squares pyramid. In a standard complete pyramidal representation the number of nodes is increased by $\frac{1}{3}$ when compared to the initial number of pixels. In the LSLP, the total number of nodes can be reduced by $\frac{1}{4}$ (e.g., made equal to the initial number of pixels) because the residual error at each step is orthogonal to the reduced resolution signal approximation. In other words, the LS REDUCE function is a projection operator with the property that

$$\mathrm{REDUCE}_{\mathrm{LS}}[\Delta f_i(k)]=0. \tag{5.6}$$

For a bi-dimensional image with M grid points, (5.6) provides us with a set of $\frac{1}{4}M$ linear constraints. The true number of degrees of freedom of the LS Laplacian is therefore $\frac{3}{4}M$ and not M as may be thought initially. In fact, we will show in the last section that the quadrature mirror filter (QMF) concept offers a simple solution for dealing with this redundancy. We will thereby also establish the relationship between the present approach and recent work in orthogonal pyramid structures [1, 12], wavelet transforms [9] and subband coding techniques [18, 20].

5.3. Link with QMF pyramids

Quadrature mirror filters, introduced by Croisier et al. in 1976 [4], provide an attractive method for splitting a signal into critically sampled filtered components. Such filter banks can be applied iteratively to produce a subband decomposition of the spectrum into octave bandwidth pieces [18]. The two attractive features of this technique are (i) the reversibility of the process (error free reconstruction) and (ii) the fact that the resulting signal decomposition uses no more samples than the initial representation. Recently, several authors have applied this concept to pyramid image compression and have reported substantial improvements in performance [1, 12, 18, 20]. QMF banks also provide an efficient way of computing wavelet transforms, as has been shown recently by Mallat and Daubechies [5, 8, 9].

The block diagram of a QMF bank is represented in Fig. 8. In the basic QMF design [12, 19], the transfer functions of the filters are chosen such that

$$\begin{aligned} F_0(z)&=G_0(1/z)=F(z),\\ F_1(z)&=F_1(1/z)=zF(-z), \end{aligned} \tag{5.7}$$

where $F(z)$ is a lowpass filter prototype satisfying the perfect reconstruction property

$$F(z)F(1/z)+F(-z)F(-1/z)=2. \tag{5.8}$$

To establish its relationship to the present approach, we will construct a QMF bank such that its lower branch (lowpass) precisely computes the least squares signal estimates derived in Section 4. We derive this result by manipulating the block diagram in Fig. 9(a), which performs successively the REDUCE and EXPAND functions described in Sections 3 and 4. The first step is to note that

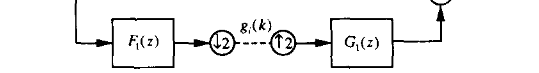

Fig. 8. Block diagram of a quadrature mirror filter bank for signal analysis and synthesis.

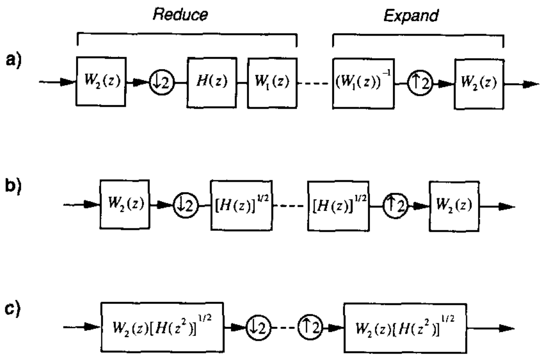

Fig. 9. QMF interpretation of the least squares approximation procedure: equivalent block diagrams.

the two central filters ($W_1(z)$ and $G(z)$) cancel each other. Second, the filter $H(z)$ is factored into a product of square-root components (Fig. 9(b)). Finally, the filters are moved on each side of the sampling modules by upsampling their impulse response by a factor of two (this is achieved by replacing z by z^2 in their z-transform) (Fig. 9(c)). At the end of this process, we have an equivalent system (i.e., same input and output) for which the pre-filters and post-filters are identical and given by

$$F(z)=F(1/z)=W_2(z)(H(z^2))^{1/2}. \tag{5.9}$$

Using (5.4), it is then easy to verify by substitution that this operator satisfies the perfect reconstruction property (5.8). Since the final output of the QMF bank is equal to its input, it follows that the corresponding highpass branch precisely codes for the residual signal displayed in the least squares Laplacian pyramid. This approach is easily extended to higher dimensions by iterating the subband decomposition along the rows and columns according to the procedure initially described by Vetterli [18]. The main advantage of such a QMF decomposition is that the residual signal is now represented without redundancy (i.e., the sum of the number of lowpass and highpass samples is equal to the initial number of samples).

In order to obtain a decomposition closer to our initial design, we choose an alternative, but globally equivalent, factorization with

$$\begin{aligned}
F_0(z)&=\tfrac{1}{2}W_2(z)H(z^2)W_1(z^2),\\
G_0(z)&=\frac{W_2(z)}{W_1(z^2)},\\
F_1(z)&=\tfrac{1}{2}zW_2(-z)H(z^2)W_1(z^2),\\
G_1(z)&=z^{-1}\frac{W_2(-z)}{W_1(z^2)},
\end{aligned} \tag{5.10}$$

for which it can be verified that the filters $F_0(z)$ and $G_0(z)$ are precisely those required for the REDUCE and EXPAND function described in Sections 3 and 4. The advantage of this latter decomposition is that it can be implemented recursively using the fast algorithms described in Sections 3 and 4 (see Figs. 2(d) and 2(e)) and Appendix A. The highpass components can be evaluated using the same procedure, provided that the FIR smoothing kernel ($W_2(z)$) (which is used as a pre- and post-filter) is replaced by its modulated and shifted counterparts: $zW_2(-z)$ and $W_2(-z)/z$, respectively. We note that this particular choice of filters corresponds to a linear algebraic transform that is non-orthogonal, in contrast to a standard QMF bank as defined by (5.7)–(5.8), which can be interpreted as an orthogonal transformation [12]. These results also suggest that a QMF implementation of the present least squares image pyramid could provide a further improvement by $\frac{1}{4}$ over the coding procedure used in the experimental part of this paper.

6. Conclusion

Two methods for improving the Laplacian pyramid proposed by Burt and Adelson for image coding have been described:

(i) The EXPAND function has been redefined to ensure that the expansion of a coarser level onto a finer grid is an exact interpolation.

(ii) An improved REDUCE function has been derived in order to minimize the loss of information occurring during resolution conversion.

It is easy to modify the initial scheme to incorporate these new functions. This is achieved by adding a pre-filter and a post-filter in the expansion and reduction modules, respectively. These filters can be coded very efficiently and the resulting increase of computations is moderate.

For lossless progressive data transmission, the performance improvement that can be achieved in this way is significant. The least squares scheme performs best according to the quantitative criteria used in this paper. Preliminary results suggest that this approach allows improved image coding according to the lossy scheme developed by Burt and Adelson. The least squares pyramid also stands as an interesting alternative to the widely used Gaussian pyramid and should be useful in a variety of multi-resolution image processing algorithms. It has also been shown that the present approach can be linked to the family of QMF image pyramids (e.g., orthogonal pyramids, wavelet transforms, subband coders).

Appendix A. Efficient recursive filter implementation

This appendix describes an efficient way to implement the recursive filters required in the generation of a least squares Laplacian pyramid. The transfer functions of these operators are special cases of a class of IIR symmetrical filters described by the following equation:

$$H_{2N}(z) = \frac{c_0}{[z^N+z^{-N}]+\left(\sum_{k=1}^{N-1} a_k[z^k+z^{-k}]\right)+a_0} = \frac{c_0 z^N}{P_{2N}(z)}, \quad (A.1)$$

where c_0 and $\{a_k | k=0, \ldots, N-1\}$ are given constant coefficients. Due to the reverse symmetry of its coefficients, the polynomial $P_{2N}(z)$ of degree $2N$ is such that $z^{-N}P_{2N}(z)=z^N P_{2N}(z^{-1})$. This implies that the zeros of $P_{2N}(z)$ must necessarily appear in reciprocal pairs (e.g., if $P_{2N}(z_i)=0$ then $P_{2N}(z_i^{-1})=0$). These roots, which are assumed to lie outside the unit circle, are denoted by

$$\{(z_i, z_i^{-1}) \mid |z_i|<1, i=1, \ldots, N\}.$$

Consequently, $H_{2N}(z)$ can be factored as

$$H_{2N}(z)=c_0 \prod_{i=1}^{N} H(z; z_i), \quad (A.2)$$

where $H(z; z_i)$ is the transfer of a simple symmetrical filter, decomposed as

$$H(z; z_i)=\left(\frac{-z_i}{(1-z_i z^{-1})(1-z_i z)}\right) = \frac{-z_i}{(1-z_i^2)}\left(\frac{1}{1-z_i z^{-1}}+\frac{1}{1-z_i z}-1\right). \quad (A.3)$$

It follows that (A.1) can be implemented by a cascade of simple operators of the form specified by (A.3), each of which has a symmetrical exponential impulse response:

$$h(k; z_i)=\frac{-z_i}{(1-z_i^2)} z_i^{|k|}. \quad (A.4)$$

The implementation of these elementary units is based on the decomposition of $H(z; z_i)$ into a sum of simple causal and anti-causal first order systems, as given by the right-hand side of (A.3). The corresponding recursive filter equations are

$$\begin{aligned} y^+(k)&=x(k)+z_i y^+(k-1), \quad k=1, \ldots, K, \\ y^-(k)&=x(k)+z_i y^-(k+1), \quad k=K, \ldots, 1, \\ y(k)&=c_i(y^+(k)+y^-(k)-x(k)), \end{aligned} \quad (A.5)$$

where $x(k)$ and $y(k)$ are the input and output signals, respectively, and where $c_i=-z_i/(1-z_i^2)$ is a scaling constant. For boundary conditions specified by (3.7), the recursion is initialized with the following values:

$$\begin{aligned} y^{+}(1) &= \sum_{i=1}^{k_0} z_i^{|k-1|} x(k), \\ y^{-}(K) &= y^{+}(K), \end{aligned} \tag{A.6}$$

where k_0 is such that $z_i^{|k_0|}$ is below some prescribed level of precision. A slightly more economical alternative is an implementation based on a product decomposition:

$$\begin{aligned} y^{+}(k) &= x(k) + z_i y^{+}(k-1), \quad k = 1, \ldots, K, \\ y(K) &= c_i(2y^{+}(K) - x(K)), \\ y(k) &= z_i(y(k+1) - y^{+}(k)), \quad k = K-1, \ldots, 1. \end{aligned} \tag{A.7}$$

We note that the second equation is borrowed from the sum decomposition and is required to initialize the backward recursion correctly.

All operations in (A.7) (respectively (A.5)) are real, and it is necessary to use one (respectively two) one-dimensional real array(s) for storing the filtered sequences with sufficient precision to avoid a recursive propagation of errors. It is relatively straightforward to write a general subroutine that implements (A.1) from a succession of simple convolutions of the form (A.5) or (A.7); no additional intermediate storage is necessary for this task. It is also more economical to combine the individual scaling factors in (A.1) and (A.5) or (A.7) into a single multiplication at the end of the process. The relevant filter parameters for implementing some of the operators described in Sections 3 and 4 using this strategy are given in Table A.

This approach is also applicable in higher dimensions through the successive use of the same one-dimensional filter along the various dimensions of the data. For digital images there is no need for floating point data storage other than the one-dimensional array(s) required by the basic one-dimensional filtering module.

Appendix B. Derivation of the least squares coefficients

The error criterion (4.2) is decomposed as

$$\begin{aligned} \varepsilon^2 = & \sum_{k=-\infty}^{+\infty} (f_{i-1}(k))^2 \\ & + \sum_{k=-\infty}^{+\infty} (w_2 * [p_i]_{\uparrow 2}(k))^2 \\ & -2 \sum_{k=-\infty}^{+\infty} f_{i-1}(k)(w_2 * [p_i]_{\uparrow 2}(k)). \end{aligned}$$

Table A

Transfer functions and characteristic parameters of interpolation and least squares approximations filters as a function of the parameter a

Filter	a	Transfer function	c_0	Poles: $\{\lvert z_i\rvert < 1 \mid i = 1, \ldots, N\}$
$g(k)$	$\geqslant \frac{1}{4}$	$\dfrac{2/(1-2a)}{z + 4a/(1-2a) + z^{-1}}$	$\dfrac{2}{1-2a}$	$z_1 = \dfrac{\sqrt{4a-1} - 2a}{1-2a}$
$h(k; a=\frac{1}{3})$	0.3333	$\dfrac{36}{z^2 + 17z + 36 + 17z^{-1} + z^{-2}}$	36	$z_1 = -0.574403$, $z_2 = -0.068417$
$h(k; a=\frac{3}{8})$	0.375	$\dfrac{64}{z^2 + 28z + 70 + 28z^{-1} + z^{-2}}$	64	$z_1 = -0.446463$, $z_2 = -0.0395661$
$h(k; a=\frac{2}{5})$	0.4	$\dfrac{100}{(z^{-1} + 3 + z)(z^{-1} + 38 + z)}$	100	$z_1 = -0.381966$, $z_2 = -0.026334$
$h(k; a=\frac{1}{2})$	0.5	$\dfrac{4}{z + 6 + z^{-1}}$	4	$z_1 = \sqrt{8} - 3$

Defining $w_{22}(k)=\sum_{l=-\infty}^{+\infty} w_2(l)w_2(k+l)$ and making the appropriate change of variables, this expression is rewritten as

$$\begin{aligned}\varepsilon^2=&\sum_{k=-\infty}^{+\infty}(f_{i-1}(k))^2\\&+\sum_{k=-\infty}^{+\infty}[p_i]_{\uparrow 2}(k)(w_{22}*[p_i]_{\uparrow 2}(k))\\&-2\sum_{k=-\infty}^{+\infty}(w_2*f_{i-1}(k))[p_i]_{\uparrow 2}(k).\end{aligned}$$

The restriction of the two rightmost terms to the non-zero values of $[p_i]_{\uparrow 2}(k)$ (e.g., a decimation by a factor of 2) yields

$$\begin{aligned}\varepsilon^2=&\sum_{k=-\infty}^{+\infty}(f_{i-1}(k))^2\\&+\sum_{k=-\infty}^{+\infty}p_i(k)([w_{22}]_{\downarrow 2}*p_i(k))\\&-2\sum_{k=-\infty}^{+\infty}p_i(k)([w_2*f_{i-1}]_{\downarrow 2}(k)).\end{aligned}\qquad\text{(B.1)}$$

The partial derivative of (B.1) with respect to $p_i(k)$ is given by

$$\begin{aligned}\frac{\partial\varepsilon^2}{\partial p_i(k)}=&\,2[w_{22}]_{\downarrow 2}*p_i(k)\\&-2[w_2*f_{i-1}]_{\downarrow 2}(k).\end{aligned}\qquad\text{(B.2)}$$

The optimal sequence of coefficients is obtained by setting this expression equal to zero, which results in (4.3). Q.E.D.

Acknowledgments

I would like to thank Dr Akram Aldroubi for his helpful suggestions and comments and Dr Murray Eden for his constant support.

References

[1] E.H. Adelson, E. Simoncelli and R. Hingorani, "Orthogonal pyramid transforms for image coding", *Proc. SPIE Conf. Visual Communication and Image Processing*, Cambridge, MA, October 1987, pp. 50–58.

[2] P.J. Burt, "Fast algorithms for estimating local image properties", *Comput. Graph. Image Process.*, Vol. 21, 1983, pp. 368–382.

[3] P.J. Burt and E.H. Adelson, "The Laplacian pyramid as a compact code", *IEEE Trans. Comm.*, Vol. COM-31, No. 4, 1983, pp. 337–345.

[4] A. Croisier, D. Esteban and C. Galand, "Perfect channel splitting by use of interpolation, decimation, tree decomposition techniques", *Proc. Internat. Conf. Information Sciences/Systems*, Patras, August 1976, pp. 443–446.

[5] I. Daubechies, "Orthogonal bases of compactly supported wavelets", *Comm. Pure Appl. Math.*, Vol. 41, 1988, pp. 909–996.

[6] A.K. Jain, *Fundamentals of Digital Image Processing*, Prentice Hall, Englewood Cliffs, NJ, 1989.

[7] M. Kunt, A. Ikonomopoulos and M. Kocher, "Second generation image-coding techniques", *Proc. IEEE*, Vol. 73, No. 4, 1985, pp. 549–574.

[8] S.G. Mallat, "Multiresolution approximations and wavelet orthogonal bases of $L^2(R)$", *Trans. Amer. Math. Soc.*, Vol. 315, No. 1, 1989, pp. 69–87.

[9] S.G. Mallat, "A theory of multiresolution signal decomposition: The wavelet representation", *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-11, No. 7, 1989, pp. 674–693.

[10] D. Marr and E. Hildreth, "Theory of edge detection", *Proc. Roy. Soc. London*, Vol. B207, 1980, pp. 187–217.

[11] J. Max, "Quantizing for minimum distortion", *IRE Trans. Inform. Theory*, Vol. IT-6, 1960, pp. 7–12.

[12] E.P. Simoncelli and E.H. Adelson, "Non-separable extensions of quadrature mirror filters to multiple dimensions", *Proc. IEEE*, Vol. 78, No. 4, 1990, pp. 652–664.

[13] S.L. Tanimoto, "Image transmission with gross information first", *Comput. Graph. Image Process.*, Vol. 9, 1979, pp. 72–76.

[14] D. Terzopoulos, "Multilevel computational processes for visual surface reconstruction", *Comput. Vision Graph. Image Process.*, Vol. 24, 1983, pp. 52–96.

[15] K.H. Tzou, "Progressive image transmission: a review and comparison of techniques", *Optical Engrg.*, Vol. 26, 1987, pp. 581–589.

[16] M. Unser and M. Eden, "Multi-resolution feature extraction and selection for texture segmentation", *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. PAMI-11, No. 7, 1989, pp. 717–728.

[17] M. Unser, A. Aldroubi and M. Eden, "Fast *B*-spline transforms for continuous image representation and interpolation", *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 13, No. 3, 1991, pp. 277–285.

[18] M. Vetterli, "Multi-dimensional sub-band coding: some theory and algorithms", *Signal Processing*, Vol. 6, No. 2, April 1984, pp. 97–112.

[19] M. Vetterli, "A theory of multirate filter banks", *IEEE Trans. Acoust. Speech Signal Process.*, Vol. ASSP-35, No. 3, 1987, pp. 356–372.

[20] J.W. Woods and S.D. O'Neil, "Sub-band coding of images", *IEEE Trans. Acoust. Speech Signal Process.*, Vol. ASSP-34, 1986, pp. 1278–1288.