Image Filtering Using Multiresolution Representations

Surendra Ranganath

Abstract— **This paper shows how multiresolution representations can be used for filter design and implementation. These representations provide a coarse frequency decomposition of the image, which forms the basis for two filtering techniques. The first method, based on image pyramids, is used for approximating the convolution of an image with a given mask. In this technique, a filter is designed using a least-squares (ls) procedure based on filters synthesized from the basic pyramid equivalent filters. This approximates the mask frequency characteristic. Next, implementation of the filter involves linearly combining scaled and filtered pyramid levels, using weights obtained from the ls procedure. By this method, filtering and pyramid image coding can be combined, efficiently integrating the filter into the reconstruction procedure for the coded image. The second method is an adaptive noise reduction algorithm. An optimally filtered image is synthesized from the multiresolution levels, which in this case, are maintained at the original sampling density. Individual pixels of the image representation are linearly combined under a minimum mean square error criterion. This uses a local signal to noise ratio estimate to provide the best compromise between noise removal and resolution loss.**

Index Terms—**Adaptive image restoration, image enhancement, least-squares filter approximation, multiresolution image representations, pyramid coding.**

I. Introduction

IMAGE filtering is often implemented in the spatial domain as the convolution of the image and a two dimensional (2-D) mask. When the mask size is large, the requirements on convolver hardware can be quite demanding in terms of execution speed and precision. To ease the computational burden, an approximate convolution can be realized. Specifically, an approximation procedure can be used to decompose the large mask into a set of smaller masks. Then, sequential convolution of the image with this set of masks yields a good approximation to the desired result.

Various approximation techniques and filter realizations have been considered in the literature. Pratt [1] used the singular value decomposition to approximate a mask by a sum of separable masks. Abramatic and Faugeras [2] presented least squares and Chebychev design algorithms for mask approximation. Later, this idea, and its means of realization was patented [3]. In [4], O'Leary considered various forms of approximation by 3 × 3 kernels, and their numerical behavior. In other work, Wells [5], used cascaded uniform filters for efficient realization of Gaussian filters and pyramids.

In this paper, we consider image filtering using multiresolution representations. These representations consist of a set of filtered copies of the image, obtained by iterative filtering with a generating kernel. Two types of representations can be generated. One involves successively prefiltering and decimating the image, leading to a tapering, pyramid structure. The other involves filtering, but the original sampling density is maintained in all the filtered copies. By linearly combining the multiresolution levels (scaled, if necessary) with some weights, the frequency content of the original image can be altered, yielding a filtered image. To choose the weights in order to obtain an image filtered by some desired characteristic, an approximation technique can be used.

Another useful application that we consider is noise removal. The global filtering techniques outlined above can be used to design useful smoothing filters. However, since images are often statistically stationary only over local regions, global filters are unlikely to produce uniformly good results over the whole image. The multiresolution representations provide a natural means for local adaptation, which can be used to synthesize and apply locally optimum filters.

There is an important difference between the filter approximation work in [1]–[4] and ours. In previous work, the values of the mask coefficients obtained from the approximation procedure were unrestricted. The individual masks therefore exhibited no special structure or properties, except that when they were *sequentially* applied on the given image, the approximate filtered result was obtained. The individual intermediate images in the sequence, although computed, are by themselves not useful. In our approach, the approximation procedure is explicitly based on the image representation, and the filter description is obtained in terms of the representation. Thus the representation supplies the intermediate images to obtain the filtered image, in addition to being useful for achieving other processing objectives. An important example in this context is combining image coding and filtering. The usefulness of pyramids for image coding has been established [6]. Since the pyramid structure is used in our filter approximation, and also in the coding scheme, the same generic operations are useful for both objectives. Thus, coded images can be directly filtered during the reconstruction process, without having to first reconstruct the original image, and *then* start filtering. Besides this, progressive transmission and retrieval is possible, a desirable capability when dealing with large images. Also the

Manuscript received December 21, 1988; revised December 7, 1990. Recommended for acceptance by O. D. Faugeras.

The author is with Philips Laboratories, 345 Scarborough Road, Briarcliff Manor, NY 10510.

IEEE Log Number 9042519.

0162–8828/91/0500–0426$01.00 © 1991 IEEE

scheme naturally accommodates images of varying sizes.

An application area where the above requirements are important is medical imaging. In digital radiology, images of sizes ranging from 2000 × 2000 to 100 × 100 must be transmitted and stored, and enhanced for facilitating the detection of various pathologies. The applicable filters will vary depending on the modality used, and the pathology suspected. We have used the techniques developed here for enhancing medical images. Informal opinions of radiologists regarding the quality of the enhancements has been encouraging. Based on this preliminary opinion, the methods will undergo rigorous clinical testing to formally verify clinical applicability.

Several candidate pyramid representations are available in the literature for use in filter approximation. There are Burt's pyramids [7], [8], Crowley's difference of low pass transform [9], Watson's cortex transform [10], and the class of pyramid structures used in subband image coding [11]–[13]. We based our work on Burt's structures, for their simplicity, and the number of attractive properties they possess. By proper choice of a generating kernel for the structure, he showed that good approximations to Gaussian filtered images evolve at higher pyramid levels. Also, by subtracting appropriately scaled, adjacent levels of the Gaussian pyramid, good approximations to the Laplacian of a Gaussian (LoG) filtered image were obtained. Such filters have found application in recent computer vision and image processing algorithms [14], [15]. However, in the Gaussian pyramid, the range of scales is coarsely divided into octave intervals. This is adequate in its analogy to early human vision where there is neurophysiological evidence for spatially tuned, octave bandpass channels for isolating intensity changes [15], but it is not sufficient for realizing more general filtering operations. For example, in deriving scale space representations [16], it may be desirable to have more finely spaced scales, or images filtered to arbitrary scales. These can be obtained from the Burt representations using the filtering methods developed here.

In Section II of this paper we review multiresolution representations, and provide a least squares design technique for filter approximation based on the Burt pyramid(s). In Section III, we derive an adaptive algorithm in the minimum mean square error (mmse) framework for restoration of noisy images. In Section IV, we consider numerical complexity of these algorithms by assuming the architecture of a commercially available machine. In Section V, we present some experimental results and in Section VI our conclusions.

II. Multiresolution Filter Design

A. The Multiresolution Representations

Multiresolution representations are obtained by iterative filtering of the original image, thereby obtaining copies of the original with progressively decreasing bandwidth or increasing scale. Burt [7], [8] considered two types of representations. In one, the sampling density of the original image is maintained through all the copies or levels. In the other, called the pyramid representation, the sampling density is reduced in proportion to the decrease in bandwidth through the levels. A convenient factor for successive bandwidth reduction is one octave. This simplifies numerical computations in pyramid generation and is also in accordance with experimental evidence for such filters involved in early human vision.

We first consider the octave pyramid representation. The generating kernel is a 1-D low pass filter with impulse response $h(n)$ and discrete Fourier transform (DFT) $H(\omega)$, chosen to satisfy some desirable properties, listed below. The filter is applied separably to perform 2-D filtering. First, the original image is filtered, and if $H(\omega)$ is a half bandwidth filter, the filtered copy can be decimated by a factor of two in each dimension. This is the first level of the pyramid with the original image forming the zeroth level, or the base of the pyramid. Iteratively applying filtering and decimation operations generates successive levels of the pyramid representation. For example, if the original image is of size $2^k \times 2^k, (k+1)$ pyramid levels, ordered from 0 through k can be obtained, with the highest, or kth level, containing only one pixel. In practice, pyramid generation can be terminated at any useful intermediate level, $K \leq k$. The pyramid thus obtained represents low pass (LP) filtered copies of increasing scale. A bandpass (BP) pyramid can be derived by subtracting adjacent levels of the LP pyramid. Before subtraction, the higher level must be 2-D interpolated by a factor of two to the size of the lower level. If $\{g_i\}_0^K$ represents the LP pyramid levels, and $\{l_i\}_0^K$ the BP pyramid levels, then

$$l_i = \begin{cases} g_i - g_{i+1,1}, & i = 0, 1, \cdots, K-1 \\ g_i, & i = K \end{cases}$$

where $g_{i+1,1}$ represents g_{i+1} 2-D interpolated by a factor of 2.

Considering the operations involved, certain desirable properties for $h(n)$ may be stipulated [7], [8]:

1) $h(n)$ is of odd length, so that half sample phase shifts are conveniently avoided.
2) $h(n)$ has linear phase with even symmetry about its center coefficient.
3) All the coefficients are positive and sum to unity. This allows for preservation of the average level of the image, and the possibility of Gaussian approximations in LP pyramid generation.
4) The sum of all even coefficients of $h(n)$ is equal to the sum of all odd coefficients to ensure that spurious signal content at $\omega = \pi$ is not introduced during image interpolation.

Burt chose a five coefficient filter $h(n)$, which by imposing the above requirements, is given as

$$h(n) = \begin{cases} a, & n = 0 \\ 1/4, & n = \pm 1 \\ 1/4 - a/2, & n = \pm 2 \end{cases} \tag{1}$$

with one free parameter a. With $h(n)$ and the decimation/interpolation factor specified, an equivalent filter can be associated with each pyramid level. This can be derived from the sequence of operations used to arrive at that level starting from the original image. Burt showed that when $0.375 \leq a \leq 0.4$, the equivalent filters are approximately Gaussian in shape. In this case, the LP pyramid is called the Gaussian pyramid. Since difference of Gaussians approximate the LoG well, the BP pyramid is called the Laplacian pyramid.

In the nonreduced representations, rather than changing the sampling density, the filter kernel is modified by inserting zeros between the original coefficients. For example, the first LP level is obtained by filtering the original image with $h(n)$. The next level is obtained from the first by using a filter of characteristic $H(2\omega)$. This is synthesized from $h(n)$ by inserting a zero between every sample of $h(n)$. Numerically of course, zero multiplies can be avoided. The next level is obtained by filtering the previous level with $H(4\omega)$ and so on. This method of generating these representations was chosen by Burt to correspond to the pyramid. The pyramid can be obtained from this representation by decimating the levels by appropriate factors. In practice, if this correspondence with pyramids is not required, then these representations can be obtained by any sequence of low pass filtering operations to suit a given application. For example, strictly sequential convolution of the original image with an arbitrary $h(n)$ can be used.

Although Gaussian and Gaussian derived filters have found popular application in the computer vision literature, the pyramids described above can in general be generated by using any LP generating kernel. For example, they can be generated by using approximations to ideal LP filters. This is useful when a maximum amount of information must be maintained in the reduced copies while keeping aliasing to a minimum [17].

Having seen the basic ideas used in generating the representations, we next derive the filters associated with the representations.

B. The Basis Functions

During the process of pyramid generation, the 2-D frequency plane is decomposed into octave regions, the ideal decompositions being as shown in Fig. 1. For illustration, the bands are shown as rectangular rather than circularly symmetric. It is of interest now to calculate the filters or basis functions which perform this decomposition. For this, consider the generic operations involved in pyramid generation as shown in Fig. 2(a). Fig. 2(b) illustrates the spectra of various signals that arise during the pyramid generation process. We show this for 1-D only, since we make the separable extension to 2-D. With reference to Fig. 2(a), $\tilde{g}_1$ is obtained by filtering the original sampled signal, $g_0 \leftrightarrow G_0(\omega)$ with $h(n)$. Its DFT is given by $G_0(\omega)H(\omega)$, and the signal spectra are shown in Fig. 2(b) for the baseband, $\omega \in [-\pi, \pi]$ (the spectra repeat periodically outside this interval). Decimating $\tilde{g}_1$ by a factor of 2 gives g_1, the first Gaussian level, and its DFT can be derived as [23, Appendix A1]

$$G_1(\omega) = \frac{1}{2}[H(\omega/2)G_0(\omega/2) + H(\omega/2 - \pi)G_0(\omega/2 - \pi)].$$

The second term in the above equation represents aliasing due to the repeat spectra sliding into the baseband, and is also shown in Fig. 2(b). The severity of aliasing depends on the energy in the filtered signal $\tilde{g}_1$ in $|\omega| \in [\pi/2, \pi]$. The more $H(\omega)$ attenuates the signal in this band, the lesser the aliasing error, and this can be made arbitrarily small by choosing good

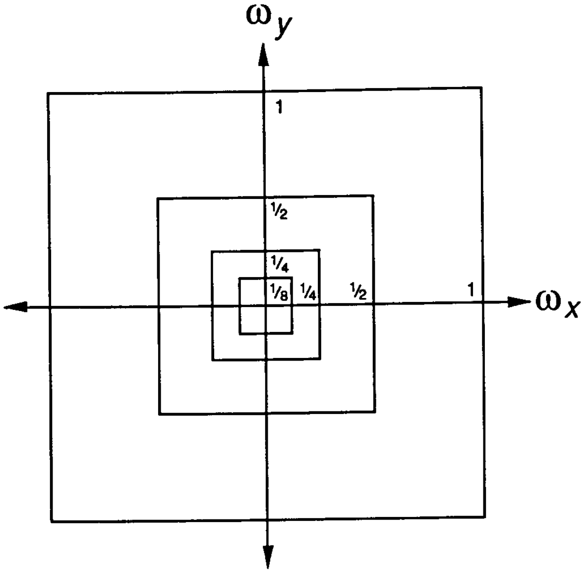


Fig. 1. Ideal decomposition of the frequency plane into octave bands. The axes are shown normalized by π.

approximations to a half bandwidth low pass filter. In the following, for ease of analysis we assume that this is the case, and neglect aliasing. Thus,

$$G_1(\omega) \approx \frac{1}{2}H(\omega/2)G_0(\omega/2)$$

In order to obtain the Laplacian level l_0, g_1 must first be interpolated to the size of g_0. Conceptually, this can be broken down into two steps, first inserting a zero between every sample of g_1 to obtain $\hat{g}_1$, and then filtering this with $h(n)$. The DFT of $\hat{g}_1$ is given by [23, Appendix A1]

$$\hat{G}_1(\omega) = G_1(2\omega) = \frac{1}{2}H(\omega)G_0(\omega).$$

Thus insertion of zeros between the samples of g_1 generates the high frequencies in $|\omega| \in [\pi/2, \pi]$ as shown in Fig. 2(b). The second step for interpolation removes these high frequencies. This can be done by low pass filtering $\hat{g}_1$ with $h(n)$, and using a gain of 2 to get $g_{1,1}$. The high frequencies that remain in $|\omega| \in [\pi/2, \pi]$ are the reconstruction or interpolation errors. Again, as in the case of decimation, we assume that this error can be made arbitrarily small, neglect it, and write

$$G_{1,1}(\omega) \approx H^2(\omega)G_0(\omega).$$

From this, the equivalent filters for the first expanded Gaussian level and the zeroth Laplacian level are given by $H^2(\omega)$, and $1 - H^2(\omega)$, respectively.

Following the above development, the equivalent filters associated with various pyramid levels can be derived. Let $F_{n,0}(\omega)$ and $L_{n,0}(\omega)$ be the filters associated with the nth Gaussian and Laplacian levels, respectively. Also let $F_{n,j}(\omega)$ and $L_{n,j}(\omega)$ be the filters associated with interpolating the nth Gaussian and Laplacian levels j times. Then the equivalent

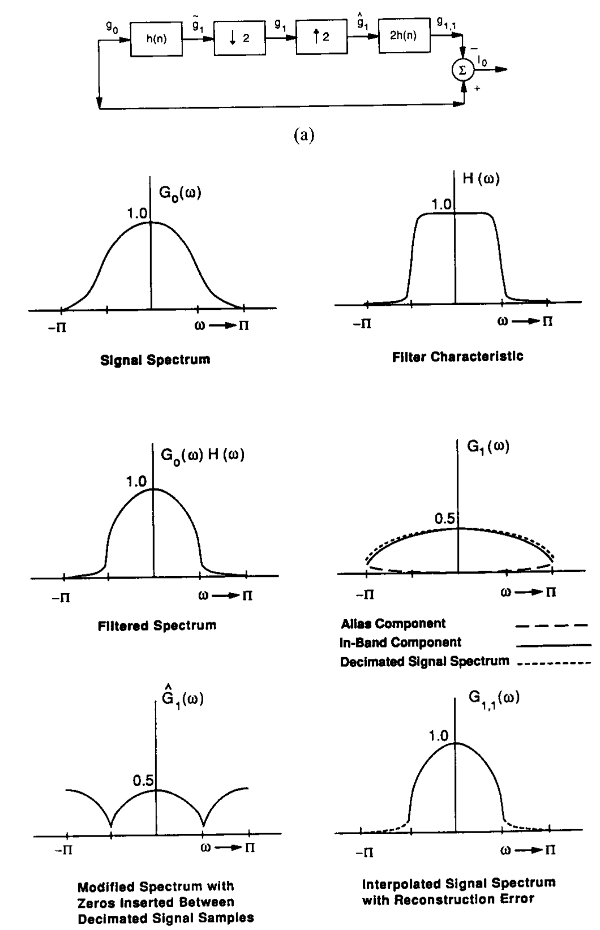


Fig. 2. (a) One-dimensional description of the generic operations in pyramid generation. (b) Signal spectra in the decimation/interpolation process.

filters of interest are given as

$$F_{n,0}(\omega) = \begin{cases} 1, & n = 0 \\ H(\omega/2)H(\omega/4)\cdots H(\omega/2^n), & n > 0 \end{cases} \tag{2a}$$

$$F_{n,1}(\omega) = \begin{cases} 1, & n = 0 \\ H^2(\omega)H(\omega/2)\cdots H(\omega/2^{n-1}), & n > 0 \end{cases} \tag{2b}$$

$$L_{n,0}(\omega) = \begin{cases} F_{n,0}(\omega) - F_{n+1,1}(\omega), & 0 \le n < K \\ F_{n,0}(\omega), & n = K \end{cases} \tag{2c}$$

$$F_{n,n}(\omega) = \begin{cases} 1, & n = 0 \\ H^2(\omega)\cdots H^2(2^{n-1}\omega), & n > 0 \end{cases} \tag{2d}$$

$$L_{n,n}(\omega) = \begin{cases} F_{n,n}(\omega) - F_{n+1,n+1}(\omega), & 0 \le n < K \\ F_{n,n}(\omega), & n = K. \end{cases} \tag{2e}$$

Each set of filters $\{F_{n,n}(\omega)\}$ and $\{L_{n,n}(\omega)\}$ is linearly independent (but not orthogonal) and moreover, the set of filters $\{L_{n,n}(\omega)\}$ satisfies

$$\sum_n L_{n,n}(\omega) = 1. \tag{3}$$

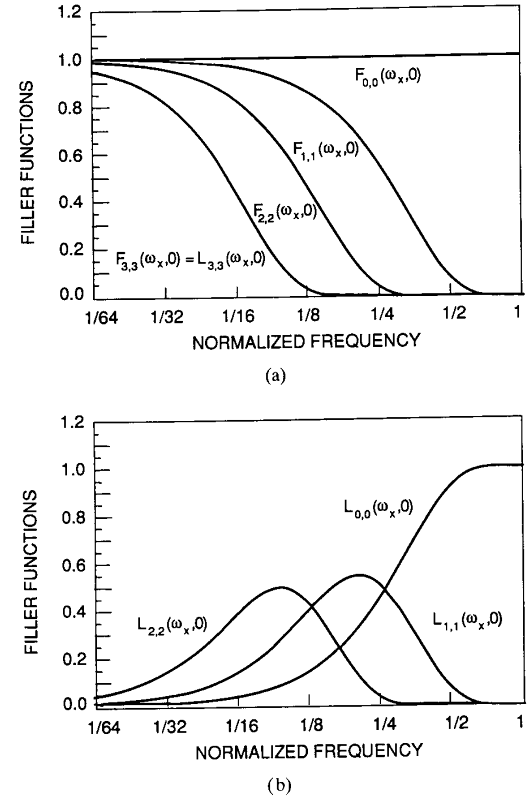


Fig. 3. (a) The equivalent filters of the Gaussian basis set shown in the $\omega_y = 0$ plane. (b) The equivalent filters of the Laplacian basis set shown in the $\omega_y = 0$ plane.

This allows exact reconstruction of the original image from the Laplacian pyramid representation. Equations (2) and (3) extend to the 2-D separable case by simply replacing $H(\omega)$ by $H(\omega_x)V(\omega_y)$ where $H(\omega_x)$ and $V(\omega_y)$ correspond to horizontal and vertical filters, respectively. Plots of these basis filters are shown for a Burt type generating kernel with

$$h(n) = \begin{cases} 3/8, & n = 0 \\ 1/4, & n = \pm 1 \\ 1/16, & n = \pm 2. \end{cases} \tag{4}$$

The plots in Fig. 3(a) and 3(b) show the filters $\{F_{n,n}(\omega_x, \omega_y)\}$ and $\{L_{n,n}(\omega_x, \omega_y)\}$, respectively, for $n = 0, 1, 2, 3$ through the $\omega_y = 0$ plane.

Since $\{L_{n,n}(\omega_x, \omega_y)\}$ is a linearly independent set of filters, we can use linear combinations of these filters to synthesize other filters, $X(\omega_x, \omega_y)$, as,

$$\sum_n a_n L_{n,n}(\omega_x, \omega_y) \approx X(\omega_x, \omega_y).$$

In the above approximation, $L_{0,0}(\omega_x, \omega_y)$ the filter corresponding to the base of the Laplacian pyramid is tuned to pass the highest octave of frequencies. Each higher level of the pyramid is associated with a filter tuned to successively lower octaves in the frequency plane. An idealized example of this frequency separation is shown in Fig. 1, when the highest pyramid level is 3. As the number of pyramid levels increase, more filters with pass bands at the lower end of the frequency spectrum are generated, but they will have decreasing influence on the approximation at higher frequencies. Thus they will only play a marginal role in shaping the approximation at higher frequencies. One may for practical purposes think that only one filter shapes each of the octave bands in the linear combination. Indeed, this is exactly the case when the generating kernel is an ideal half bandwidth low pass filter (this particular filter set is also orthogonal). Although these octave filters can be used to synthesize some useful filters, more general filters must be approximated from a less restrictive set. We show how to do this starting from the basic Laplacian filter set. The idea is to generate additional filters that will sample the frequency plane into finer regions.

Before doing so, we introduce some notation. Let $\boldsymbol{H}$ represent the operation of horizontal low pass filtering, with characteristic $H(\omega_x)$. Similarly, let $\boldsymbol{V}$ represent vertical low pass filtering with characteristic $V(\omega_y)$. Let $\boldsymbol{H}_n$ and $\boldsymbol{V}_n$ represent filtering with $H(n\omega_x)$ and $V(n\omega_y)$, respectively, and define $\overline{\boldsymbol{H}}_n = 1 - \boldsymbol{H}_n$, $\overline{\boldsymbol{V}}_n = 1 - \boldsymbol{V}_n$. The 2-D frequency plane may be subdivided into four equal regions of low, horizontal, vertical, and diagonal frequencies by operating on it with the four linearly independent filters corresponding to $\boldsymbol{HV}, \overline{\boldsymbol{H}}\boldsymbol{V}, \boldsymbol{H}\overline{\boldsymbol{V}}$, and $\overline{\boldsymbol{HV}}$, respectively. Their idealized pass bands are shown in Fig. 4. We can use these filters to subdivide any of the Laplacian bands and synthesize four new filters from $L_{n,0}(\omega_x, \omega_y)$ as

$$\{L'_{n,0}(\omega_x, \omega_y)\} = A_1(\omega_x, \omega_y)\{L_{n,0}(\omega_x, \omega_y)\}$$

where

$$A_1(\omega_x, \omega_y) = \{\boldsymbol{HV}, \overline{\boldsymbol{H}}\boldsymbol{V}, \boldsymbol{H}\overline{\boldsymbol{V}}, \overline{\boldsymbol{HV}}\}$$

The new set of augmented filters also preserves the property of exact image reconstruction since

$$\big(\boldsymbol{HV} + \overline{\boldsymbol{H}}\boldsymbol{V} + \boldsymbol{H}\overline{\boldsymbol{V}} + \overline{\boldsymbol{HV}}\big)L_{n,0}(\omega_x, \omega_y) = L_{n,0}(\omega_x, \omega_y).$$

Each of the four bands in $L_{n,0}(\omega_x, \omega_y)$ can be further decomposed into four smaller self similar bands by operating on them with filters from the set $A_2(\omega_x, \omega_y) = \{\boldsymbol{H}_2\boldsymbol{V}_2, \overline{\boldsymbol{H}}_2\boldsymbol{V}_2, \boldsymbol{H}_2\overline{\boldsymbol{V}}_2, \overline{\boldsymbol{H}}_2\overline{\boldsymbol{V}}_2\}$. This decomposition can be verified by deriving the filter characteristics corresponding to $\boldsymbol{H}_2$ and $\boldsymbol{V}_2$, starting from a low pass characteristic for $H(\omega)$, and using the appropriate operator definitions. These 16 new filters can be written as

$$\{L''_{n,0}(\omega_x, \omega_y)\} = \{A_2(\omega_x, \omega_y)\}\,\{L'_{n,0}(\omega_x, \omega_y)\}.$$

This process can be iterated to obtain finer subdivisions. By generating these augmented filters, more coefficients are made available within each octave to shape the response of the filter approximation.

In the above, the augmented filters are written in the frequency space of each Laplacian pyramid level. For linearly combining these filters, they can be written in the frequency space corresponding to the base of the pyramid. This is easily done [see (2)] as

$$\{L'_{n,n}(\omega_x, \omega_y)\} = \{A_1(2^n\omega_x, 2^n\omega_y)\}L_{n,n}(\omega_x, \omega_y)$$

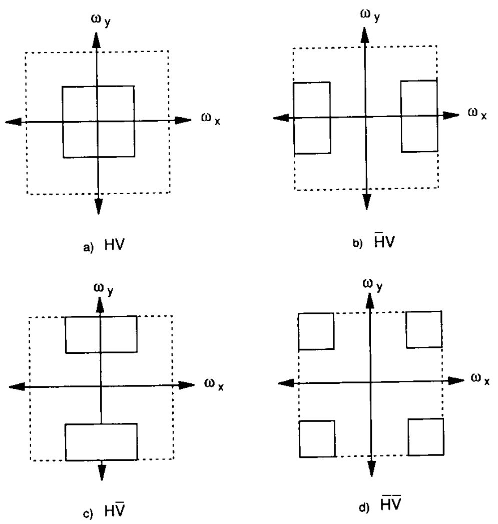


Fig. 4. Idealized pass bands of the filters, (a) $\boldsymbol{HV}$, (b) $\overline{\boldsymbol{H}}\boldsymbol{V}$, (c) $\boldsymbol{H}\overline{\boldsymbol{V}}$, and (d) $\overline{\boldsymbol{HV}}$ in the first stage of decomposition. The pass bands are enclosed by the solid lines.

and

$$\{L''_{n,n}(\omega_x, \omega_y)\} = \{A_2(2^n\omega_x, 2^n\omega_y)\}\{L'_{n,n}(\omega_x, \omega_y)\}$$

etc.

This yields an augmented set of basis filters which can be used in the linear combination, and we collectively call it the set $\{B_n(\omega_x, \omega_y)\}$.

The synthesis of the filters as obtained here reflects a desire to perfectly reconstruct the original image, and to obtain a reversible image representation with a finite number of filters. This is ensured in this case by using a low pass to high pass transformation, where the high pass filter is obtained by subtracting a low pass filter from unity. Another technique of effecting a low pass to high pass transformation is to modulate the low pass filter to π, i.e., $HPF(\omega) = LPF(\omega - \pi)$. Originally, the latter transformation was used in the subband coding of speech [18]. Subsequently, the decomposition method was extended to 2-D by Vetterli [11] and Woods and O'Neill [12]. Also in [12], a coding scheme was developed, based on the subband representation. Adelson *et al.* [13] have further explored the properties of the generating kernels involved, and have developed pyramid representations with orthogonal basis filters. Mallat [19] has provided an alternative mathematical framework based on wavelet theory for these representations. In the above class of pyramids, the generating kernels must satisfy the quadrature mirror filter (QMF) constraint for exact image reconstruction. Else, reconstructed images contain aliasing artifacts. Constrained optimization techniques [20] are used to design QMF approximations, and it has been observed that rather large even length kernels may be required for fidelity. However, in [13], smaller filters, with an odd number of taps are reported to perform satisfactorily with some modification to the pyramid synthesis and reconstruction processes.

We have chosen the subtractive form of basis filter synthesis primarily for two reasons:

1) Our main application area is medical imaging, where at present, exact image reconstruction from coded images is deemed necessary, although it is likely that this restriction will be relaxed in the future. Thus the representations on which we base our filtering techniques must allow exact reconstruction, and must be robust enough to provide good compression ratios if irreversible coding schemes are used.
2) There are less severe constraints on the choice of generating kernel, and computation and performance can be easily traded in any given application. Though we have preferred to do the processing in the spatial domain, we can also extend our filtering techniques to Fourier domain implementations such as Watson's cortex transform [10].

Having developed the basis filters of the representation above, we use these next to approximate filters.

C. Least Squares Filter Design

Given a real filter characteristic $X(\omega_x, \omega_y)$ with which the original image is to be filtered, the weighting coefficients a_i must be found such that

$$\sum_{i=0}^{m} a_i B_i(\omega_x, \omega_y) \approx X(\omega_x, \omega_y)$$

where $B_i(\omega_x, \omega_y)$ are the real, linearly independent, filters developed above. These filters have four quadrant symmetry in the plane, i.e., $B_i(\omega_x, \omega_y) = B_i(-\omega_x, \omega_y) = B_i(\omega_x, -\omega_y) = B_i(-\omega_x, -\omega_y)$. This follows directly from the separable 2-D extensions of (2), the construction of the augmented filters, and the fact that the generating kernel *h*(*n*) satisfies $H(\omega) = H(-\omega)$. Thus all filters realized by the linear combination above will retain the symmetry property, but will not necessarily be separable. A weighted least squares criterion for filter design finds the coefficients a_i by minimizing

$$e = \int_{-\pi}^{\pi}\int_{-\pi}^{\pi} W(\omega_x, \omega_y)\left[X(\omega_x, \omega_y) - \sum_{i=0}^{m} a_i B_i(\omega_x, \omega_y)\right]^2 d\omega_x \, d\omega_y$$

where $W(\omega_x, \omega_y)$ is a real weighting function chosen to distribute the error in a specified manner. If this is chosen to have four quadrant symmetry, then the expression for the least squares error reduces to

$$e = 4\int_{0}^{\pi}\int_{0}^{\pi} W(\omega_x, \omega_y)\,[X(\omega_x, \omega_y) - \sum_{i=0}^{m} a_i B_i(\omega_x, \omega_y)]^2 \, d\omega_x \, d\omega_y.$$

By uniformly sampling the $[0, \pi] \times [0, \pi]$ quadrant at $(\pi k/N, \pi l/N)$, $0 \le k, l \le N$, the discrete approximation to the least squares error becomes

$$e = 4C \sum_{k=0}^{N} \sum_{l=0}^{N} W(k, l)\left[X(k, l) - \sum_{i=0}^{m} a_i B_i(k, l)\right]^2$$

where C is a constant. The above equation can be written conveniently in matrix notation by lexicographic ordering of the sampled functions. This ordering defines an index $n = 0, 1, 2 \cdots$ as

$$n = (N+1)k + l, \quad 0 \le k, l \le N.$$

With respect to this indexing we define the following:

$\boldsymbol{x} = \{x(n) = X(k, l)\}$; an $(N+1)^2$ length vector.
$\boldsymbol{b} = \{b_i(n) = B_i(k, l)\}$; an $(N+1)^2$ length vector.
$\boldsymbol{B} = [\boldsymbol{b}_0 \boldsymbol{b}_1 \cdots \boldsymbol{b}_m]$; an $(N+1)^2 \times (m+1)$ matrix.
$\boldsymbol{D}$ = an $(N+1)^2$ diagonal matrix of entries $\{w(n) = W(k, l)\}$.
$\boldsymbol{a} = \{a_i; 0 \le i \le m\}$; the $(m+1)$ vector of coefficients.

With these definitions, we can write

$$e = 4C(\boldsymbol{x} - \boldsymbol{Ba})^T \boldsymbol{D}(\boldsymbol{x} - \boldsymbol{Ba}).$$

The least squares solution for the coefficients is then given by

$$\boldsymbol{a} = (\boldsymbol{B}^T \boldsymbol{D} \boldsymbol{B})^{-1} \boldsymbol{B}^T \boldsymbol{D} \boldsymbol{x}$$

and involves the inversion of an $(m+1) \times (m+1)$ matrix.

In our work we have sampled the frequency plane with N = 32. The dimension of the matrix to be inverted is the same as the number of basis filters used in the approximation. As a typical example, if the highest pyramid level is 3, and the l_0, l_1, l_2, and l_3 levels undergo 2, 1, 1, and 0 stages of subdivision respectively, then the total number of basis filters is 25 (16, 4, 4, and 1).

There are two mechanisms available, which can be jointly used to control the accuracy of the least square filter approximation. One is through the weighting function, $W(\omega_x, \omega_y)$, and the other is through choice of the pyramid parameters such as the highest level used, and the number of subdivisions of each level. For example, a larger number of subdivisions can be used in the frequency region where greater accuracy is desired. If the basis set used were orthogonal, selecting an appropriate combination of pyramid parameters for an acceptable result would be simple. Since the basis filters that we use are only linearly independent and not orthogonal, there is some interaction between filters that sample different frequency regions. Therefore, in practice, some experimentation may be required for choosing parameters that lead to an acceptable approximation.

A related consideration is choosing between two sets of parameters that yield acceptable filter approximations. From a computational point of view, the set that needs fewer computations during implementation is desirable. However, since the effects of aliasing have been neglected in the filter design, the accuracy of the filtered image may be different for the two sets of filters, and the more accurate set is desirable. These two requirements need not necessarily conflict. For most natural images of interest, we expected, and found in our experiments that aliasing artifacts were quite small.

Thus far we have derived basis filters associated with the pyramid representation, and used them for approximating filters. These filters are implemented globally on the image. In some situations, adaptive filtering may be desirable, where different regions of the image must be filtered differently according to content. One example of this is noise reduction, which we consider in the next section. Rather than pyramids, we use the nonreduced representations.

III. Restoration of Noisy Images

Smoothing of noisy images is often the first important preprocessing step in information extraction. Consequently, a great deal of attention has been focused on this problem, and many algorithms are available in the literature [21]. The problem can be formulated in a stochastic or deterministic framework, and a variety of solutions can be derived under different constraints and assumptions. The solutions found can be recursive, transform based, block processed, computed using local neighborhood operations, etc. Of these the last variety is computationally simple, lends itself quite naturally to parallel processing, and often provides satisfactory performance. Since the nonreduced multiresolution representation is constructed by iterative use of small masks, we consider it to derive a new adaptive algorithm for image smoothing.

These representations provide increasingly smoothed copies of the noisy image, but the individual levels do not provide the best tradeoff between image resolution loss and noise removal. Each equivalent filter that is generated is best suited to a given scale of detail in the image. Since natural images contain details over a range of scales, locally optimum filters must be applied. We show that these can be synthesized at every pixel from the filters involved in generating the representation, by using the linear mmse criterion.

Let the noisy image be given by

$$y_0(k,l) = s(k,l) + n(k,l)$$

where $s(k,l)$ represents the original image and $n(k,l)$, the noise field. The image $s(k,l)$ is assumed to be a wide sense stationary random field, with zero mean and covariance function

$$r'_s(k,l) = \sigma_s^2 r_s(k,l) \tag{5}$$

where σ_s^2 is the image variance, and $r_s(k,l)$ is the normalized, unit covariance function. Some of these restrictive assumptions about the signal will be relaxed a little further into this development to accommodate natural images. The noise field is assumed to be wide sense stationary, white, with zero mean and uncorrelated with the image field. The noise covariance function is given by

$$r_n(k,l) = \sigma_n^2 \delta(k,l)$$

with

$$\delta(k,l) = \begin{cases} 1, & k=0, l=0 \\ 0, & o.w. \end{cases}$$

Multiresolution representations are obtained iteratively, using only one generating kernel, but the method of generation uniquely defines an equivalent mask at each level. A 1-D generating kernel which is even symmetric, and has unit dc gain, can be chosen and used separably in the 2-D case. If the representations are generated as described by Burt, or by repeated convolutions with the same kernel, the above properties of the 1-D kernel carry over to the equivalent masks. Let the equivalent mask used for the ith level be $h_i(k,l)$ for $i > 0$, with support $W_i = \{-M_i \le k, l \le M_i\}$. At the bottom level, we have the noisy image with filter $h_0(k,l) = \delta(k,l)$. Then the ith level can be written as

$$y_i(k,l) = \sum_{(p,q)\in W_i} h_i(p,q) y_0(k-p, l-q). \tag{6}$$

At every pixel location, a vector of information is now available, aggregated over progressively larger neighborhoods. A linear combination of its elements is chosen as best representing the smoothed pixel in the mmse sense,

$$\hat{s}(k,l) = \sum_{i=0}^{K} a_i y_i(k,l) \tag{7}$$

where K is the highest level used. Denoting the expectation operator as $\boldsymbol{E}(\cdot)$, and minimizing the mse, $\boldsymbol{E}\left\{[s(k,l) - \hat{s}(k,l)]^2\right\}$, gives the orthogonality condition,

$$\boldsymbol{E}\{[s(k,l) - \hat{s}(k,l)] y_i(k,l)\} = 0, \quad 0 \le i \le K.$$

Substituting for $y_i(k,l)$ and $\hat{s}(k,l)$ from (6) and (7) yields

$$\sum_{m=0}^{K} \left[P(i,m) + \frac{1}{SNR} Q(i,m) \right] a_m = b_i, \quad 0 \le i \le K \tag{8}$$

where

$$P(i,m) = \sum_{(x,y)\in W_i} h_i(x,y) \sum_{(p,q)\in W_m} h_m(p,q) r_s(x-p, y-q) \tag{9a}$$

$$Q(i,m) = \sum_{(x,y)\in W_i} h_i(x,y) \sum_{(p,q)\in W_m} h_m(p,q) \delta(x-p, y-q) \tag{9b}$$

$$b_i = \sum_{(x,y)\in W_i} h_i(x,y) r_s(x,y) \tag{9c}$$

and

$$SNR = \frac{\sigma_s^2}{\sigma_n^2}. \tag{9d}$$

In matrix form, (8) can be written as

$$\left[\boldsymbol{P} + \frac{1}{SNR} \boldsymbol{Q} \right] \boldsymbol{a} = \boldsymbol{b} \tag{10}$$

Some properties of (8)–(10) are given by the following.

Theorem: Given a positive definite covariance sequence $r'_s(k,l)$ (5), the coefficient matrix $\left[\boldsymbol{P} + \frac{1}{SNR}\boldsymbol{Q}\right]$ is positive definite. Moreover, when $SNR = \infty, a_0 = 1$ and $a_m = 0, 1 \le m \le K$. When $SNR = 0, a_m = 0, 0 \le m \le K$.

The proof is given in Appendix A2. The above theorem shows that a solution always exists for the weighting coefficients under mild assumptions. Moreover, the coefficients exhibit appropriate behavior at extremes of *SNR*. For high *SNR*'s, the input image receives large weighting in synthesizing the output image, whereas at low *SNR*'s, the output is driven to zero. Due to the positive definiteness of the coefficient matrix in (10), the $K+1$ dimensional system of equations can be solved for *a*(*m*) using the Cholesky decomposition in about $(K+1)^3/6$ operations. Typically, K is 3 or 4, requiring about 11 or 21 operations to calculate the coefficients.

In the above, we assumed that $s(k,l)$ was zero mean. If, however, $s(k,l)$ had mean μ, then we require that $E[\hat{s}(k,l)] = \mu$. By introducing an additional variable a_{K+1} in the estimate $\hat{s}(k,l)$, i.e.,

$$\hat{s}(k,l) = \sum_{i=0}^{K} a_i y_i(k,l) + a_{K+1}$$

we obtain,

$$a_{K+1} = \mu\left(1 - \sum_{i=0}^{K} a_i\right). \tag{11}$$

For natural images, the mean may not be globally stationary. In this case we assume that the mean is locally stationary over small neighborhoods, and estimate it for every pixel as

$$\hat{\mu}(k,l) = \frac{1}{K+1}\sum_{i=0}^{K} y_i(k,l). \tag{12}$$

This is an unbiased estimate of the local mean, following from (6) and

$$\sum_{(k,\,l)\in W_i} h_i(k,l) = 1, \qquad 0 \le i \le K.$$

Using (11) and (12), the modified estimate is given as

$$\hat{s}(k,l) = \sum_{i=0}^{K} a_i' y_i(k,l)$$

with new coefficients

$$a_i' = a_i + \frac{1}{K+1}\left(1 - \sum_{m=0}^{K} a_m\right). \tag{13}$$

From the properties of the coefficients in the zero mean case, the behavior of $\{a_i'\}$ can be shown to be similar. The only difference is that when $SNR = 0, a_i' = \frac{1}{K+1}$, so that the output is an unbiased estimate of the input mean.

In order to calculate the weighting coefficients a_m, the image covariance function in (5) and the noise variance σ_n^2 must be known or estimated. The latter can be assumed known from consideration of the image formation process, or it can be estimated from flat regions of the noisy image. The form of the normalized covariance function can be estimated for a class of images, or a parametric form assumed. This leaves only the image variance to be specified, to calculate the *SNR*, and thus the weighting coefficients. This is again estimated for every pixel from a local neighborhood as

$$\hat{\sigma}_s^2(k,l) = \frac{1}{(M+1)^2}\sum_{(i,j)=-M}^{M} y_0^2(k-i,l-j) - \tilde{\mu}^2(k,l) \tag{14a}$$

where

$$\tilde{\mu}(k,l) = \frac{1}{(M+1)^2}\sum_{(i,j)=-M}^{M} y_0(k-i,l-j). \tag{14b}$$

The *SNR* is then estimated as

$$SNR(k,l) = \begin{cases} \frac{\hat{\sigma}_s^2(k,l)-\sigma_n^2}{\sigma_n^2}, & \hat{\sigma}_s^2(k,l) > \sigma_n^2 \\ C, & o.w. \end{cases} \tag{14c}$$

where C is a small positive number.

Though in principle, the weighting coefficients can be determined at every pixel, for visual smoothing (and computational efficiency) it is sufficient to precalculate a set of weights at discrete *SNR* values in a given range. To summarize, the multiresolution algorithm for noise reduction is given as the following.

Precomputation (for Weights):

1) Estimate, or assume known σ_n^2 and $r_s(k,l)$.
2) Calculate a set of weighting coefficients for a discrete set of *SNR* values.

Actual Computations (for Noise Reduction):

3) Compute the multiresolution levels.
4) At each pixel of the noisy image estimate the *SNR* as in (14), and pick the coefficients that correspond to the closest precalculated *SNR* value.
5) Form the linear combination of the multiresolution values at each pixel.

IV. Implementation and Numerical Complexity

For estimating the complexity of operations in pyramid filtering and nonreduced image restoration, we assume that the generating kernel $h(n)$ satisfies the conditions in Section II-A, and is of length p. The original image is assumed to be of size $N \times N$. To illustrate the implementation by example, a specific architecture is assumed: a pipelined SIMD machine, with four processors and four planes of configurable image memory with scratchpad arrays for computation. These could correspond to three color channels and a transparency channel. A commercially available machine with such an architecture is the PIXAR [22]. By designing special multiprocessor hardware to exploit further the inherent parallelism in the algorithm, computation times can be further reduced.

A. Gaussian Pyramid Generation

The Gaussian pyramid is generated by prefiltering and decimating by factors of 2 in each dimension. First, the rows are filtered. Since decimation by a factor of 2 follows, filtered output need be obtained only for every other sample in a row. Similarly, for the column filtering and decimation which follow. Thus the total operation count for generating the first level is

$$N \cdot \frac{1}{2}Np(\text{row ops}) + \frac{1}{2}N \cdot \frac{1}{2}Np(\text{column ops}) = \frac{3}{4}N^2 p.$$

The number of operations for generating all the Gaussian pyramid levels is then bounded by

$$\frac{3}{4}N^2 p \sum_{i=0}^{\infty}\left(\frac{1}{4}\right)^i = N^2 p.$$

The time on the assumed architecture would be proportional to $\frac{1}{4}N^2p$. The total storage for the Gaussian pyramid is bounded by $N^2\sum_{i=0}^{\infty}\left(\frac{1}{4}\right)^i = 4/3N^2$.

B. Laplacian Pyramid Generation

The Laplacian pyramid is derived from the Gaussian pyramid by factors of two interpolation and adjacent level subtraction. Interpolating a sequence of length $\frac{1}{2}N$ to N involves inserting one zero between every existing sample, and filtering the resulting sequence with $h(n)$. Next, it can be seen that to obtain interpolated samples at the location of a zero sample and an existing sample requires $\frac{1}{2}(p-1)$ and $\frac{1}{2}(p+1)$ operations, respectively. Therefore, the total operations for $\frac{1}{2}N$ to N length interpolation is $\frac{1}{2}Np$. This implies that an $\frac{1}{2}N \times \frac{1}{2}N$ image (first Gaussian level) can be interpolated up to $N \times N$ in $\frac{3}{4}N^2p$ operations. Again, the interpolation operation count for generating the Laplacian pyramid is bounded by N^2p. Also, an additional $4/3N^2$ operations are required for subtracting adjacent, scaled levels. Reconstructing the original image from the Laplacian pyramid representation also requires $N^2p + 4/3N^2$ operations. On the PIXAR, all of the operations can be implemented in time proportional to a quarter of the operation count. The storage is the same as for the Gaussian pyramid.

C. Filtering

The filtered output can be obtained by embedding the filtering and linear combinations into the reconstruction process from the Laplacian pyramid. There are various cases that can be considered with regard to frequency subdivision of the Laplacian levels.

One Stage of Subdivision: In these types of computations, some, or all of the Laplacian levels are filtered and linearly combined with various filtered copies. In the first stage of subdivision of any Laplacian level, four filtered copies are obtained, representing horizontal, vertical, diagonal, and low pass frequencies. A linear combination of these yields a filtered Laplacian level, given by

$$\hat{L}_{n,0} = \left(b_0\boldsymbol{HV} + b_1\overline{\boldsymbol{H}}\boldsymbol{V} + b_2\boldsymbol{H}\overline{\boldsymbol{V}} + b_3\overline{\boldsymbol{HV}}\right)L_{n,0},\quad n = 0, 1, 2, \cdots, K.$$

Using the fact that $\overline{\boldsymbol{H}} = 1 - \boldsymbol{H}$ and $\overline{\boldsymbol{V}} = 1 - \boldsymbol{V}$, the above equation simplifies in the number of convolutions involved to

$$\hat{L}_{n,0} = (a_0 + a_1\boldsymbol{H} + a_2\boldsymbol{V} + a_3\boldsymbol{HV})L_{n,0},\quad n = 0, 1, 2, \cdots, K$$

where the a_i are linear combinations of b_i. The operation count for this is $N^2/4^n(4p+8)$. The last number in this accounts for the four multiplications and additions for the linear combination. For the PIXAR implementation, the image memories are ordered from 0 to 3, and the above computations are realized as follows:

Stage 1:

1) Load L_n into all 4 memories.
2) Filter memories 1 and 3 with $\boldsymbol{H}$.
3) Filter memories 2 and 3 with $\boldsymbol{V}$.
4) Linearly combine the four memories into an accumulator.

All four processors can be used to implement each convolution by using them to simultaneously filter four rows or columns at a time. Therefore the net processing time for module Stage 1 on the PIXAR is proportional to time $N^2/4^n(p+2)$.

Two Stages of Subdivision: Here, each of the bands obtained in the first stage is further subdivided into 4 bands. Simplifying the expressions to reduce the number of convolutions as done above, the filtered output can be written as

$$\begin{aligned}\hat{L}_{n,0} = {} & (a_0 + a_1\boldsymbol{H} + a_2\boldsymbol{V} + a_3\boldsymbol{HV})L_{n,0}\\ & + (a_4 + a_5\boldsymbol{H} + a_6\boldsymbol{V} + a_7\boldsymbol{HV})\boldsymbol{H}_2L_{n,0}\\ & + (a_8 + a_9\boldsymbol{H} + a_{10}\boldsymbol{V} + a_{11}\boldsymbol{HV})\boldsymbol{V}_2L_{n,0}\\ & + (a_{12} + a_{13}\boldsymbol{H} + a_{14}\boldsymbol{V} + a_{15}\boldsymbol{HV})\boldsymbol{H}_2\boldsymbol{V}_2L_{n,0}\end{aligned}$$

Stage 2:

1) Perform Stage 1 computations.
2) Filter the 4 image memories a line at a time with $\boldsymbol{H}_2$, and linearly combine the result into the accumulator. The operations in this step do not overwrite image memory contents from step 1.
3) Filter the 4 image memories with $\boldsymbol{V}_2$, overwriting the image memories with the filtered results and linearly combine the result into the accumulator.
4) Filter the 4 image memories with $\boldsymbol{H}_2$, and linearly combine the result into the accumulator.

This takes four times longer to compute than Stage 1, i.e., time proportional to $4N^2/4^n(p+2)$. Along similar lines, it can be shown that Stage 3 would require 20 times the computations as Stage 1. The largest computational burden is demanded by the lowest Laplacian level, and computational cost decreases exponentially with higher levels. With dedicated hardware, these operations could be made more efficient. Finally, as an important special case we note that when there is no subdivision of any of the Laplacian levels, filtering simply involves introducing weighting factors for the levels during the reconstruction process. This involves negligible overhead when a pyramid representation is available, as for example, in image coding.

D. Image Restoration

The restoration algorithm adaptively combines the original image and K filtered copies on a pixel by pixel basis. Since four image memories are assumed, K is at most three in the examples we consider. A set of weights themselves have been assumed to be pre-calculated. In order to apply appropriate weights at a pixel, the *SNR* must be estimated. This is efficiently done by using a sliding window technique. If M is the side of a square window over which the variance of a pixel is estimated, then each pixel needs approximately $6M$ operations for this calculation. Calculating the multiresolution levels requires $2pK$ operations per pixel. The linear combination of the levels with the optimum weights need $K+1$ operations per pixel, giving a total operation count

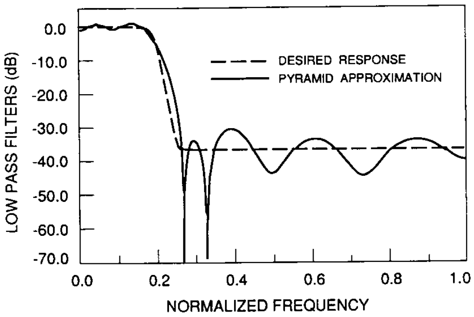


Fig. 5. The $\omega_y = 0$ slice of the frequency plane showing the desired response and the pyramid $K = 3(2, 1, 1, 1)$ approximation.

of $N^2(6M + 2pK + K + 1)$. To use realistic numbers, if $K = 3, M = 7$, and $p = 3$, this amounts to $64N^2$ operations. On the PIXAR, the time would be proportional to roughly a quarter of this operation count. Storage for $K + 1$ images would be required.

V. Experimental Results and Discussion

The images used in the experiments were all 1024 × 1024, 8 bit, monochrome images, and three examples are shown.

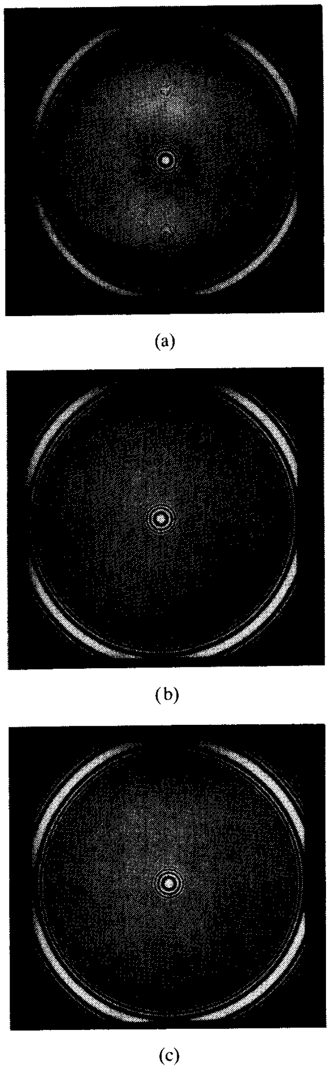
Example 1: Low Pass Filter Design

Here the least squares procedure was used to approximate a circularly symmetric low pass filter, given parametrically by [10],

$$X(\omega_x, \omega_y) = (1 - A)(\gamma/\omega_c)^2 \exp\left[-\pi(r\gamma/\omega_c)^2\right] \otimes \prod(r/2\omega_c) + A \tag{15}$$

where A is the stop band attenuation, γ is a sharpness factor that controls the width of the transition region, ω_c is the corner frequency of the filter, $r = \left(\omega_x^2 + \omega_y^2\right)^{1/2}, \otimes$ is the convolution operator, and $\prod$ is a rectangular pulse of unit width and height. The $\omega_y = 0$ slice of the filter response obtained with $\gamma = 4, \omega_c = 0.2\pi$, and $A = 0.015$ (36 dB stop band attenuation) is shown in Fig. 5. This filter was approximated by the pyramid realization with $K = 3$ the highest level, L_0 band Stage 2 decomposed, L_1, L_2, and L_3 bands each Stage 1 decomposed. For brevity, we hereafter denote such pyramid realizations as $K = 3(2, 1, 1, 1)$. The resulting filter characteristic is also plotted in Fig. 5.

Results of implementation were compared on a circular zone plate image shown in Fig. 6(a). The frequency increases linearly from the center upto a maximum, and then decreases again. The power at all frequencies is kept constant. It should be noted that Fig. 6(a), does not purely represent the intensities of the digital samples, but also incorporates the responses of the display system, and the photographic process. Fig. 6(b) shows the result of convolving a 25 × 25 mask representing the required filter with the zone plate. The mask was obtained by truncating the inverse 2-D discrete Fourier transform of (15) to the above support, and normalizing for unit dc gain. The result of the pyramid realization is shown in Fig. 6(c). Considering that aliasing and reconstruction errors were neglected in the design process, and the simple generating kernel of (4) was used, the result is a good approximation.



(a)

(b)

(c)

Fig. 6. Experiments in low pass filter approximation. (a) Original zone plate image. (b) Filtered image obtained by convolving with a 25 × 25 mask representing the desired impulse response. (c) Filtered image from the $K = 3(2, 1, 1, 1)$ pyramid realization.

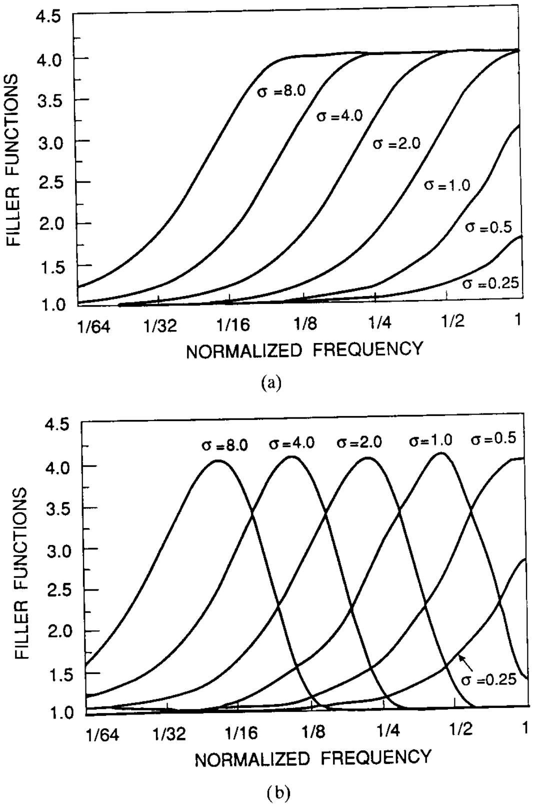
Example 2: Enhancement Filters

In this example, two families of high pass and band pass enhancement filters given by

$$X_1(\omega_x, \omega_y) = a - (a - 1)\exp\left(-r^2\sigma^2/2\right)$$

and

$$X_2(\omega_x, \omega_y) = 1 + \frac{1}{2}b\sigma^2 r^2 \exp\left(-r^2\sigma^2/2\right)$$



(a)

(b)

Fig. 7. An $\omega_y = 0$ slice through a family of high pass enhancement filters designed using a $K = 3(2,1,1,0)$ pyramid realization. (b) An $\omega_y = 0$ slice through a family of bandpass enhancement filters designed using a $K = 3(2,1,1,0)$ pyramid realization, except the $\sigma = 8$ case which was realized by $K = 3(1,2,1,1)$.

respectively, were designed. In the above equations, $r = \left(\omega_x^2 + \omega_y^2\right)^{1/2}$. For the filter designs we chose $a = 4, b = 3$, and increased σ in octave steps from 0.25 to 8.0. All of the above filters were realized by pyramid approximation. With one exception, all of them were realized with $K = 3(2,1,1,0)$. The bandpass characteristic with $\sigma = 8$ was realized using $K = 3(1,2,1,1)$. The family of filter characteristics from this approximation are shown in Figs. 7(a) and 7(b), for the highpass and bandpass types, respectively.

Enhancement results are shown for the digital chest radiograph of Fig. 8(a). The results obtained using the $\sigma = 4$ characteristic from the two families is shown in Fig. 8(b)–(e). The high pass enhancement result is shown in Fig. 8(b). To assess the errors introduced from using the pyramid realization, the result should be compared to that obtained by convolving the original image with the impulse response of the desired characteristic. The latter image was computed using a 25 × 25 dc normalized mask, obtained from the inverse DFT of the desired filter characteristic. Rather than showing this image, where differences with the pyramid realization are not discernible, we show an error image in Fig. 8(c). This shows the absolute difference between the two images, multiplied by a factor of 100. The mse between the "true" and approximate realizations was 1.53. The corresponding results for the bandpass enhancement is shown in Fig. 8(d) (pyramid realization) and (e) (absolute error multiplied by 100). The mse in this case was 1.96. The band pass enhancement is predominantly in the low–mid frequencies. While this type of filter characteristic may not be suitable for global enhancement, it could possibly be used as a customized filter for enhancing local pathologies with suitable adaptation. Enhancements of the type in Fig. 8(b) have been informally judged by a few radiologists to be quite good globally, and much better than the traditionally used unsharp mask. However, a rigorous testing of these filters for enhancing pathology is required.

Example 3: Noise Reduction

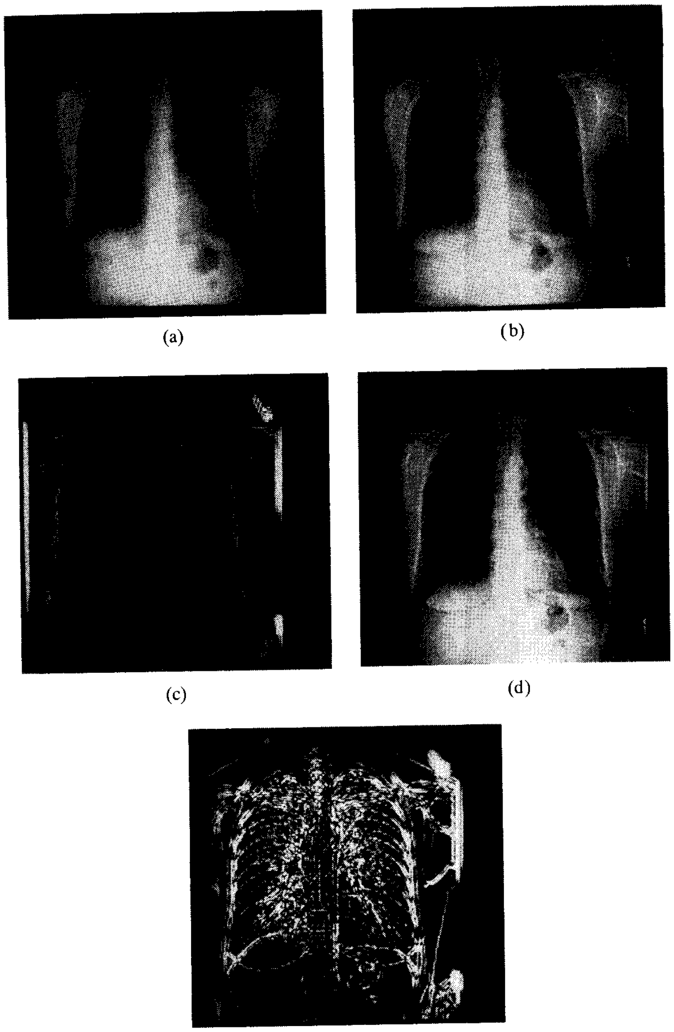
Here we show results of the noise reduction algorithm using the nonreduced representations. Fig. 9(a) shows the noisy image with a global *SNR* of 5 dB $\left(\sigma_n^2 = 564\right)$. Fig. 9(b)–(d) are the progressively filtered copies in the representation. The generating kernel applied separably was

$$h(n) = \begin{cases} \frac{1}{2}, & n = 0 \\ \frac{1}{4}, & n = \pm 1. \end{cases}$$

The equivalent filters progressing from the noisy image to the lowest resolution image correspond to the characteristics 1, $H(\omega)$, $H(\omega)H(2\omega)$, and $H(\omega)H(2\omega)H(4\omega)$. Calculation of the optimum coefficients requires specifying the covariance function and the *SNR*. We chose the form of the unit covariance function to be

$$r_s(k,l) = 0.9^{\left(k^2+l^2\right)^{1/2}}$$

although this could be estimated for a class of images. The noise variance was assumed to be known. The *SNR* was calculated at every pixel using a 7 × 7 sliding window, and using (14). Rather than solving for the optimal coefficients at every pixel, a set of precalculated weights were used. About 600 sets, obtained at uniform intervals in the range $0.01 \leq SNR \leq 12.5$ were precalculated. The filtered image resulting from a weighted sum of the corresponding pixels in the different levels is shown in Fig. 9(e). As expected, this is better than any of the individual levels in the representation, since the net filter kernel is adapted to match the signal and noise content. The mean square errors in each image was also calculated, using the original noiseless image, and these were 101.4 (Fig. 9(b), first level), 78.6 (Fig. 9(c), second level), 120.9 (Fig. 9(d), third level) and 55.8 (Fig. 9(e), adaptively smoothed result). We compared this result to the case where the optimal coefficients were computed at every pixel, and found no visual difference. Window sizes of 5 × 5, 13 × 13, and 21 × 21, were also tried for computing the *SNR* estimates, before settling on the 7 × 7 window. Smaller windows are sensitive to high frequency signal variations, but not in regions of slowly varying signal. In the latter situation, the *SNR* is underestimated, and leads to oversmoothing. On the other hand, larger windows, though more accurate estimators of *SNR*


(a) (b)

(c) (d)

(e)

Fig. 8. Experiments in enhancement. (a) A digital chest radiograph. (b) High pass enhancement using the $\sigma = 4$ realization. (c) High pass enhancement error image multiplied by a factor of 100. (d) Bandpass enhancement using the $\sigma = 4$ realization. (e) Bandpass enhancement error image multiplied by a factor of 100.

in stationary signal regions, are prone to errors when they encompass different locally stationary regions.

VI. Conclusions

In this paper, we have shown the use of multiresolution structures for filter design and implementation. The frequency decomposition provided by the pyramid structure, augmented if necessary by finer decompositions, was used in a least squares technique for filter approximation. The weights obtained from this procedure were used to linearly combine the image decompositions to provide the filtered result. The implementation scheme was shown to have a parallel structure and to integrate quite naturally into the reconstruction process of pyramid coded images. This saves the necessity of having to first reconstruct the coded image, and then implement the original convolution. The work here was motivated by a desire to combine various image processing tasks using an appropriate image representation. Burt's pyramid representation served this purpose and as we have shown can integrate tasks of filtering and coding. Adaptive filtering using this representation to implement feature dependent filtering remains to be explored. In contrast to our work, previous work concerned with approximating convolution masks with smaller masks, were not based on any image representation. Thus the mask decomposition that was obtained previously served only to realize the required convolution, and served no other useful purpose.

We have used the nonreduced multiresolution representation to formulate an adaptive algorithm for noise reduction. The optimal coefficients which linearly combine the multiresolution levels on a pixel by pixel basis provide a good compromise between noise removal and signal blurring. Though in our implementation we explicitly computed all the multiresolution levels, this is not necessary, and other implementations are

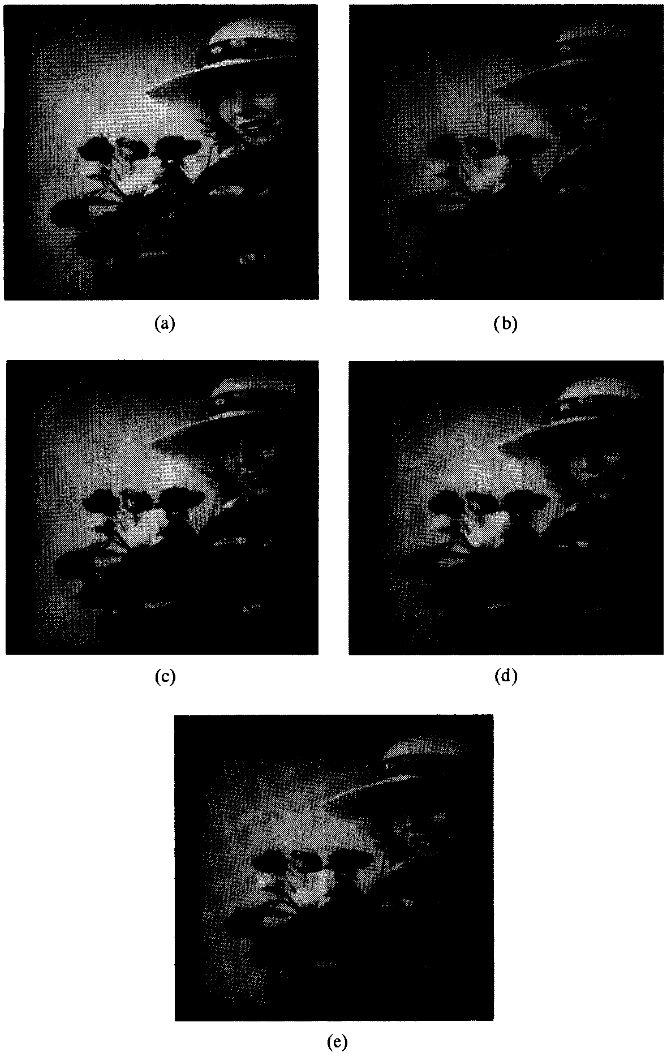


Fig. 9. Experiments in noise reduction. (a) Noisy image with *SNR* = 5 dB, forming the base of the multiresolution representation. (b) The first level. (c) The second level. (d) The third level. (e) The adaptively smoothed result.

possible depending on the hardware available.

Appendix A1

Decimation

Let the sequence $x(n)$ be decimated by a factor of 2 to give $y(n)$. Then, $y(n) = x(2n)$. The Fourier transform of $y(n)$ is

$$Y(\omega) = \sum_n y(n)e^{-j\omega n} = \sum_n x(2n)e^{-j\omega n}$$
$$= \sum_{k \text{ even}} x(k)e^{-j\omega k/2}.$$

Equivalently, the last summation can be written over all k as

$$Y(\omega) = \sum_k \frac{1}{2}\left[x(k) + (-1)^k x(k)\right]e^{-j\omega k/2}$$

or

$$Y(\omega) = \frac{1}{2}\sum_k x(k)e^{-j\omega k/2}$$
$$+ \frac{1}{2}\sum_k x(k)e^{-j(\omega/2-\pi)k}.$$

So,

$$Y(\omega) = \frac{1}{2}[X(\omega/2) + X(\omega/2 - \pi)].$$

Interpolation

Let $y(n)$ be a sequence obtained by inserting a zero between every sample of a sequence $x(n)$. Then

$$y(n) = \begin{cases} 0, & n \text{ odd} \\ x(n/2), & n \text{ even.} \end{cases}$$

Then the Fourier transform of $y(n)$ is

$$Y(\omega) = \sum_n y(n)e^{-j\omega n} = \sum_{n \text{ even}} x(n/2)e^{-j\omega n}.$$

Substituting $k = n/2$ in the last summation above,

$$Y(\omega) = \sum_k x(k)e^{-j2\omega k} = X(2\omega).$$

Appendix A2

Theorem: Given a positive definite covariance sequence $r'_s(k,l)$, the coefficient matrix $[\boldsymbol{P}+\frac{1}{SNR}\boldsymbol{Q}]$ is positive definite. Moreover, when $SNR = \infty$, $a_0 = 1$ and $a_m = 0$, $1 \le m \le K$. When $SNR = 0$, $a_m = 0$, $0 \le m \le K$.

Proof: The element $P(i,m)$ of the matrix $\boldsymbol{P}$ from (9a) is

$$P(i,m) = \sum_{(x,y)\in W_i} h_i(x,y) \sum_{(p,q)\in W_m} h_m(p,q) r_s(x-p, y-q). \tag{A1}$$

Using the inverse Fourier transform relation,

$$r_s(x-p,y-q) = \frac{1}{(2\pi)^2}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi} S_s(\omega_x,\omega_y)e^{j[\omega_x(x-p)+\omega_y(y-q)]} \cdot d\omega_x\, d\omega_y \tag{A2}$$

where $S_s(\omega_x,\omega_y)$ is the unit spectral density function of the image random field. From the positive definiteness of the covariance sequence, it follows that $S_s(\omega_x,\omega_y) > 0$. Using (A2) and the Fourier transform relation,

$$H_i(\omega_x,\omega_y) = \sum_{(x,y)\in W_i} h_i(x,y)e^{-j(\omega_x x+\omega_y y)} \tag{A3}$$

in (A1), we get

$$P(i,m) = \frac{1}{(2\pi)^2}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi} S_s(\omega_x,\omega_y)H_i^*(\omega_x,\omega_y) \cdot H_m(\omega_x,\omega_y)\, d\omega_x\, d\omega_y. \tag{A4}$$

In the following, for brevity, we write $S_s(\omega_x,\omega_y)$ and $H_i(\omega_x,\omega_y)$ simply as S_s and H_i. The expression for $Q(i,m)$ follows similarly. Then the elements of the matrix $\boldsymbol{A} = \boldsymbol{P} + \frac{1}{SNR}\boldsymbol{Q}$ can be written as

$$A(i,m) = \frac{1}{(2\pi)^2}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi}\left[S_s + \frac{1}{SNR}\right] H_i^* H_m\, d\omega_x\, d\omega_y. \tag{A5}$$

Now, the matrix $\boldsymbol{A}$ is positive definite if $\sum_{i=0}^{K}\sum_{m=0}^{K} a_i^* A(i,m) a_m > 0$. Substituting for $A(i,m)$ from (A5) into the quadratic form, we get

$$\frac{1}{(2\pi)^2}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi}\left[S_s + \frac{1}{SNR}\right]\left|\sum_{i=0}^{K} a_i H_i\right|^2 d\omega_x\, d\omega_y.$$

Since $S_s > 0, \boldsymbol{A}$ is positive definite.

When $SNR = \infty, \boldsymbol{A} = \boldsymbol{P}$. Hence $\boldsymbol{P}$ is also positive definite. Using equations (A2)–(A4) in (8) and (9c), for the $SNR = \infty$ case, (8) becomes

$$\sum_{m=0}^{K} a_m \int_{-\pi}^{\pi}\int_{-\pi}^{\pi} S_s H_i^* H_m\, d\omega_x\, d\omega_y = \int_{-\pi}^{\pi}\int_{-\pi}^{\pi} S_s H_i\, d\omega_x\, d\omega_y, \quad 0 \le i \le K. \tag{A6}$$

Since $h_i(x,y) = h_i(x)h_i(y)$ and $h_i(x) = h_i(-x), H_i^* = H_i$ and (A6) can be rewritten as

$$(a_0 - 1)\int_{-\pi}^{\pi}\int_{-\pi}^{\pi} S_s H_i^*\, d\omega_x\, d\omega_y + \sum_{m=0}^{K} a_m \int_{-\pi}^{\pi}\int_{-\pi}^{\pi} S_s H_i^* H_m\, d\omega_x\, d\omega_y = 0, \qquad 0 \le i \le K \tag{A7}$$

where we have used the fact that $H_0 = 1$. Defining a new set of coefficients, $a'_0 = a_0 - 1, a'_m = a_m, 1 \le m \le K$, (A7) in matrix form becomes $\boldsymbol{P}\boldsymbol{a}' = 0$. Since $\boldsymbol{P}$ is positive definite, $\boldsymbol{a}' = 0$. Hence, $a_0 = 1$ and $a_m = 0, 1 \le m \le K$.

When $SNR = 0$, i.e., when the signal variance is zero, and the noise variance is not, (10) reduces to $\boldsymbol{Q}\boldsymbol{a} = 0$. It is easy to show that $\boldsymbol{Q}$ is positive definite. Hence $\boldsymbol{a} = 0$.

Acknowledgment

I thank M. Shneier for many valuable discussions and suggestions during the course of this work. H. Blume of Philips Medical Systems provided digital radiographs, suggestions on enhancement parameters for these, and arranged for radiologists to view them. R. Prodan for an efficient convolution routine and for help in setting up the display system from which the results were photographed.

References

[1] W. K. Pratt, "An intelligent image processing display terminal," in *Proc. SPIE Tech. Symp.*, vol. 27, San Diego, CA, Aug. 1979.

[2] J. F. Abramatic and O. D. Faugeras, "Sequential convolution techniques for image filtering," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-30, no. 1, pp. 1–10, Feb. 1982.

[3] W. K. Pratt *et al.*, "Method and apparatus for improved digital image processing," US Patent 4 330 833, 1982.

[4] D. P. O'Leary, "Some algorithms for approximating convolutions," *Comput. Vision, Graphics, Image. Processing*, vol. 41, no. 3, pp. 333–345, Mar. 1988.

[5] W. M. Wells, "Efficient synthesis of gaussian filters by cascaded uniform filters," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, no. 2, pp. 234–239, Mar. 1986.

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

[7] P. J. Burt, "Fast filter transforms for image processing," *Comput. Graphics Image Processing*, no. 16, pp. 20–51, 1981.

[8] ——, "Fast algorithms for estimating local image properties," *Comput. Graphics Image Processing*, no. 21, pp. 368–382, 1983.

[9] J. L. Crowley and R. M. Stern, "Fast computation of the difference of low pass transform," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, pp. 212–222, 1984.

[10] A. B. Watson, "The cortex transform: Rapid computation of simulated neural images," *Comput. Vision, Graphics, Image Processing*, vol. 39, pp. 311–327, 1987.

[11] M. Vetterli, "Multidimensional sub-band coding: Some theory and algorithms," *Signal Processing*, vol. 6, pp. 97–112, Apr. 1984.

[12] J.W. Woods and S.D. O'Neill, "Subband coding of images," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-34, no. 5, pp. 1278–1288, Oct. 1986.

[13] E. Adelson *et al.,* "Orthogonal pyramid transforms for image coding," in *Proc. SPIE Conf. Visual Communications and Image Processing II,* vol. SPIE-845, 1987, pp. 50–58.

[14] A. Rosenfeld, Ed., *Multiresolution Image Processing and Analysis.* New York: Springer-Verlag, 1984.

[15] D. Marr, *Vision.* New York: Freeman, 1982.

[16] A.P. Witkin, "Scale space filtering," in *Proc. 7th Int. Joint Conf. Artificial Intell.,* pp. 1019–1021, Palo Alto, CA, 1983, pp. 1019–1021.

[17] P. Meer *et al.,* "Frequency domain analysis and synthesis of image pyramid generating kernels," *IEEE Trans. Pattern Anal. Machine Intell.,* vol. PAMI-9, no. 4, pp. 512–522, July 1987.

[18] R.E. Crochiere *et al.,* "Digital coding of speech in subbands," *Bell Syst. Tech. J.,* vol. 55, pp. 1069–1085, Oct. 1976.

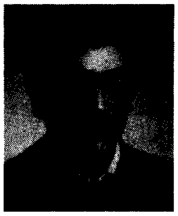
[19] S.G. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," Dep. Comput. Inform. Sci., Univ. Pennsylvania, GRASP Lab. Tech. Memo MS-CIS-87-22, 1987.

[20] J.D. Johnston, "A filter family designed for use in quadrature mirror filter bank," in *Proc. Int. Conf. Audio, Acoustics, Speech, Signal. Processing,* Apr. 1980, pp. 291–294.

[21] R. Chellappa and A.A. Sawchuk, Eds., *Digital Image Processing and Analysis, Volume I: Digital Image Processing.* Washington, DC: IEEE Computer Society Press, 1985.

[22] Pixar, Inc., San Rafael, CA.

[23] R.W. Schafer and L.R. Rabiner, "A digital signal processing approach to interpolation," *Proc. IEEE,* vol. 61, no. 6, pp. 692–702, June 1973.



Surendra Ranganath was born in India on March 23, 1954. He received the B. Tech. degree in electrical engineering from the Indian Institute of Technology, Kanpur, in 1975, the M.E. degree in electrical communication engineering from the Indian Institute of Science, Bangalore, in 1977, and the Ph.D. degree from the University of California at Davis in 1983.

From 1982 to 1985 he was with the Applied Research Group at Tektronix, Inc. where he was working in the area of digital video processing for enhanced and high definition TV. Since 1986, he has been with Philips Laboratories, Briarcliff Manor, NY, in the medical imaging group. His research interests are in digital signal and image processing, computer vision, and neural networks.