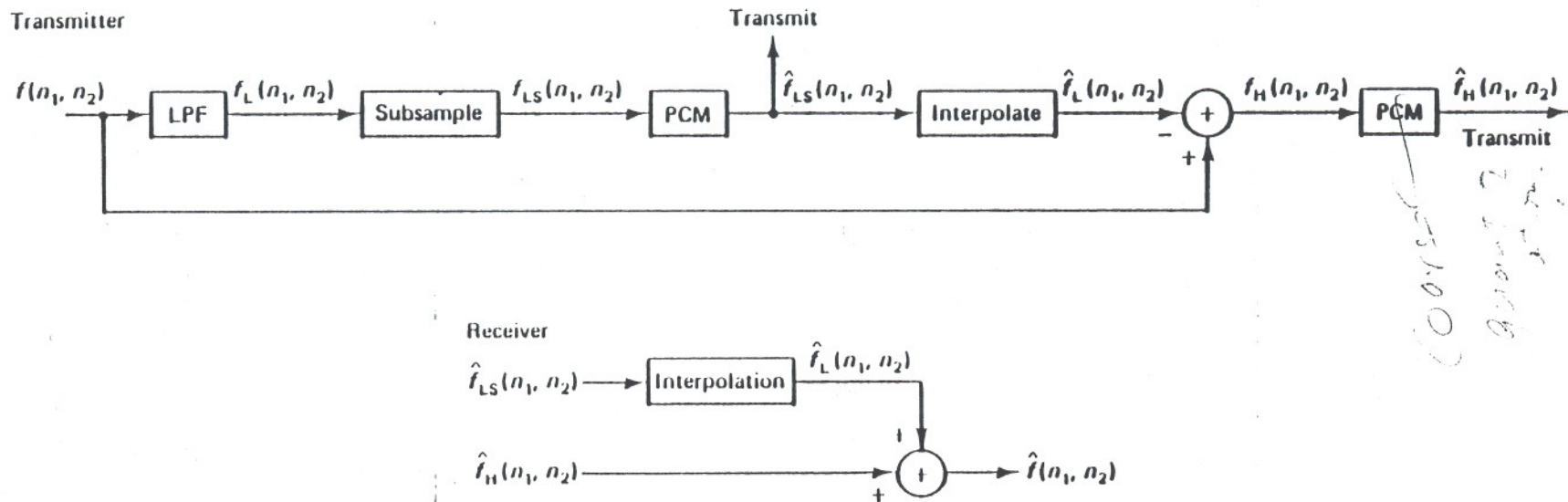


Two-Channel Image Coder



$f_L(n_1, n_2)$: Can be under-sampled (typically by 8×8), but requires above 5 bits/sample

$f_H(n_1, n_2)$: Cannot be under-sampled, but can be coarsely quantized (2-3 bits/pixel)

$$\text{Bit rate} \approx \frac{5}{64} + 2-3 \text{ bits/pixel} \approx 2-3 \text{ bits/pixel}$$

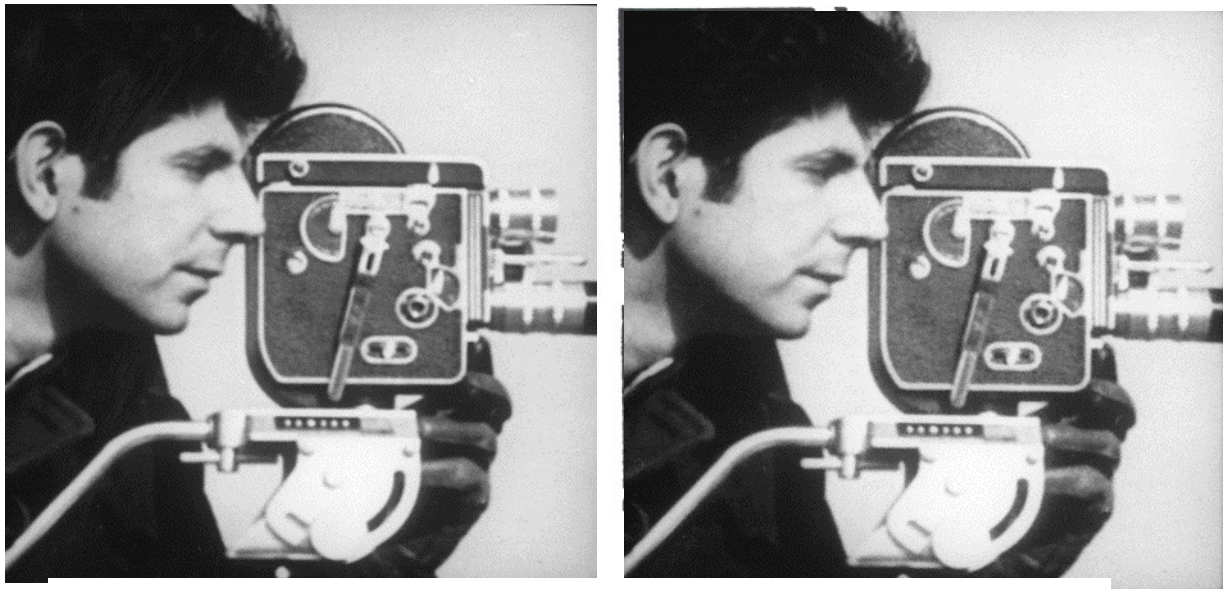


Figure 10.32 Example of image coding by a two-channel coder. (a) Original image of 512 x 512 pixels; (b) coded image at 3 1/2 bits/pixel. NMSE = 1.0%, SNR = 19.8 dB.

It is possible to develop many image representations [Rosenfeld] that can be viewed as pyramids. In this section, we discuss one particular representation developed by [Burt and Adelson]. This pyramid representation consists of an original image and successively lower resolution (blurred) images and can be used for image coding.

Let $f_0(n_1, n_2)$ denote an original image of $N \times N$ pixels where $N = 2^M + 1$, for example, 129×129 , 257×257 , 513×513 , and so forth. It is straightforward to generate an image of $(2^M + 1) \times (2^M + 1)$ pixels from an image of $2^M \times 2^M$ pixels, for example, by simply repeating the last column once and the last row once. We assume a square image for simplicity. We will refer to $f_0(n_1, n_2)$ as the base level image of the pyramid. The image at one level above the base is obtained by lowpass filtering $f_0(n_1, n_2)$ and then subsampling the result. Suppose we filter $f_0(n_1, n_2)$ with a lowpass filter $h_0(n_1, n_2)$ and denote the result by $f_0^t(n_1, n_2)$ so that

$$f_0^t(n_1, n_2) = L[f_0(n_1, n_2)] = f_0(n_1, n_2) * h_0(n_1, n_2) \quad (10.43)$$

where $L[\cdot]$ is the lowpass filtering operation. Since $f_0^t(n_1, n_2)$ has a lower spatial resolution than $f_0(n_1, n_2)$ due to lowpass filtering, we can subsample $f_0^t(n_1, n_2)$. We denote the result of the subsampling operation by $f_1(n_1, n_2)$. The image $f_1(n_1, n_2)$ is smaller in size than $f_0(n_1, n_2)$ due to subsampling and is the image at one level above the base of the pyramid. We will refer to $f_1(n_1, n_2)$ as the first-level image of the pyramid. The second-level image, $f_2(n_1, n_2)$, is obtained by lowpass filtering the first-level image $f_1(n_1, n_2)$ and then subsampling the result. This procedure can be repeated to generate higher level images $f_3(n_1, n_2)$, $f_4(n_1, n_2)$, and so forth.

Pyramid Coding and Subband Coding

- **Basic Idea:** Successive lowpass filtering and subsampling.

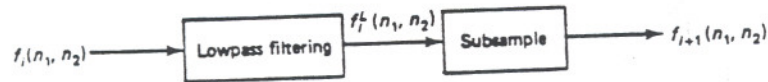


Figure 10.33 Process of generating the $i + 1$ th-level image $f_{i+1}(n_1, n_2)$ from the i th-level image $f_i(n_1, n_2)$ in Gaussian pyramid image representation.

- Filtering:

$$f_i^L(n_1, n_2) = f_i(n_1, n_2) * h(n_1, n_2)$$

- Subsampling.

$$f_{i+1}(n_1, n_2) = \begin{cases} f_i^L(2n_1, 2n_2) & 0 \leq n_1, n_2 \leq 2^{M-1} \\ 0 & \text{Otherwise} \end{cases}$$

- Type of filter determines the kind of pyramid.
- Gaussian pyramid: $h(n_1, n_2) = h(n_1)h(n_2)$

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

a is between .3 and .6

Pyramid Coding and Subband Coding

- Application to image coding:
 - Code successive images down the pyramid from the ones above it.
 - Use intrafram coding techniques to code the image at top of the pyramid.
 - Interpolate $f_{i+1}(n_1, n_2)$ to obtain a prediction for $f_i(n_1, n_2)$.

$$\hat{f}_i(n_1, n_2) = I[f_{i+1}(n_1, n_2)]$$

- Code the prediction error:

$$e_i(n_1, n_2) = f_i(n_1, n_2) - \hat{f}_i(n_1, n_2)$$

to construct $f_i(n_1, n_2)$.

- Repeat until the bottom level image, i.e. the original is reconstructed.
- The sequence $f_i(n_1, n_2)$ is a *Gaussian Pyramid*.
- The sequence $e_i(n_1, n_2)$ is a *Laplacian Pyramid*.
- Other examples of Pyramid coding:
 - Subband coding.
 - Wavelet coding.

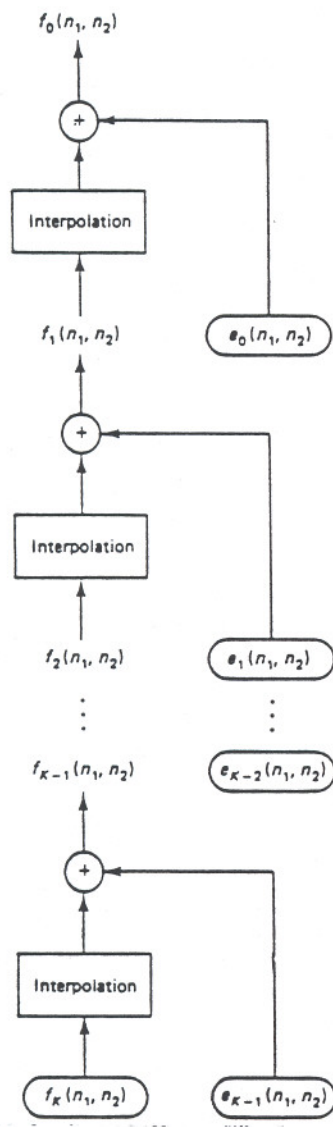


Figure 10.37 Laplacian pyramid generation. The base image $f_0(n_1, n_2)$ can be reconstructed from $e_i(n_1, n_2)$ for $0 \leq i \leq K - 1$ and $f_K(n_1, n_2)$.



Figure 10.36 Example of the Gaussian pyramid representation for image of 513 x 513 pixels with $K=4$.

The Gaussian pyramid representation can be used in developing an approach to image coding. To code the original image $f_0(n_1, n_2)$, we code $f_1(n_1, n_2)$ and the difference between $f_0(n_1, n_2)$ and a prediction of $f_0(n_1, n_2)$ from $f_1(n_1, n_2)$. Suppose we predict $f_0(n_1, n_2)$ by interpolating $f_1(n_1, n_2)$. Denoting the interpolated image by $f_1'(n_1, n_2)$, we find that the error signal $e_0(n_1, n_2)$ coded is

$$\begin{aligned} e_0(n_1, n_2) &= f_0(n_1, n_2) - I[f_1(n_1, n_2)] \\ &= f_0(n_1, n_2) - f_1'(n_1, n_2) \end{aligned} \quad (10.46)$$

where $I[\cdot]$ is the spatial interpolation operation. The interpolation process expands the support size of $f_1(n_1, n_2)$, and the support size of $f_1'(n_1, n_2)$ is the same as $f_0(n_1, n_2)$. One advantage of coding $f_1(n_1, n_2)$ and $e_0(n_1, n_2)$ rather than $f_0(n_1, n_2)$ is that the coder used can be adapted to the characteristics of $f_1(n_1, n_2)$ and $e_0(n_1, n_2)$. If we do not quantize $f_1(n_1, n_2)$ and $e_0(n_1, n_2)$, from (10.46) $f_0(n_1, n_2)$ can be recovered exactly by

$$f_0(n_1, n_2) = I[f_1(n_1, n_2)] + e_0(n_1, n_2). \quad (10.47)$$

In image coding, $f_1(n_1, n_2)$ and $e_0(n_1, n_2)$ are quantized and the reconstructed image $\hat{f}_0(n_1, n_2)$ is obtained from (10.47) by

$$\hat{f}_0(n_1, n_2) = I[\hat{f}_1(n_1, n_2)] + \hat{e}_0(n_1, n_2) \quad (10.48)$$

where $\hat{f}_0(n_1, n_2)$ and $\hat{e}_0(n_1, n_2)$ are quantized versions of $f_0(n_1, n_2)$ and $e_0(n_1, n_2)$. If we stop here, the structure of the coding method is identical to the two-channel coder we discussed in the previous section. The image $f_1(n_1, n_2)$ can be viewed as the subsampled lows component $f_{LS}(n_1, n_2)$ and $e_0(n_1, n_2)$ can be viewed as the highs component $f_H(n_1, n_2)$ in the system in Figure 10.31.

267



Figure 10.38 Example of the Laplacian pyramid image representation with $K = 4$. The original image used is the 513×513 -pixel image $f_0(n_1, n_2)$ in Figure 10.36. $e_i(n_1, n_2)$ for $0 \leq i \leq 3$ and $f_4(n_1, n_2)$.

the difference of the two Gaussian functions. The difference of two Gaussians can be modeled [Marr] approximately by the Laplacian of a Gaussian, hence the name "Laplacian pyramid."

From the above discussion, the pyramid coding method we discussed can be viewed as an example of subband image coding. As we have stated briefly, in subband image coding, an image is divided into different frequency bands and each band is coded with its own coder. In the pyramid coding method we discussed, the bandpass filtering operation is performed implicitly and the bandpass filters are obtained heuristically. In a typical subband image coder, the bandpass filters are designed more theoretically [Vetterli; Woods and O'Neil].

Figure 10.39 illustrates the performance of an image coding system in which $f_K(n_1, n_2)$ and $e_i(n_1, n_2)$ for $0 \leq i \leq K-1$ are coded with coders adapted to the signal characteristics. Qualitatively, higher-level images have more variance and more bits/pixel are assigned. Fortunately, however, they are smaller in size. Figure 10.39 shows an image coded at $\frac{1}{2}$ bit/pixel. The original image used is the 513×513 -pixel image $f_0(n_1, n_2)$ in Figure 10.36. The bit rate of less than 1 bit/pixel was possible in this example by entropy coding and by exploiting the observation that most pixels of the 513×513 -pixel image $e_0(n_1, n_2)$ are quantized to zero.

One major advantage of the pyramid-based coding method we discussed above is its suitability for progressive data transmission. By first sending the top-level image $f_K(n_1, n_2)$ and interpolating it at the receiver, we have a very blurred image. We then transmit $e_{K-1}(n_1, n_2)$ to reconstruct $f_{K-1}(n_1, n_2)$, which has a higher spatial resolution than $f_K(n_1, n_2)$. As we repeat the process, the reconstructed image at the receiver will have successively higher spatial resolution. In some applications, it may be possible to stop the transmission before we fully

270



Figure 10.39 Example of the Laplacian pyramid image coding with $K = 4$ at 1/2 bit/pixel. The original image used is the 513 x 513-pixel image $f_0(n_1, n_2)$ in Figure 10.36

recover the base level image $f_0(n_1, n_2)$. For example, we may be able to judge from a blurred image that the image is not what we want. Fortunately, the images are transmitted from the top to the base of the pyramid. The size of images increases by approximately a factor of four as we go down each level of the pyramid.

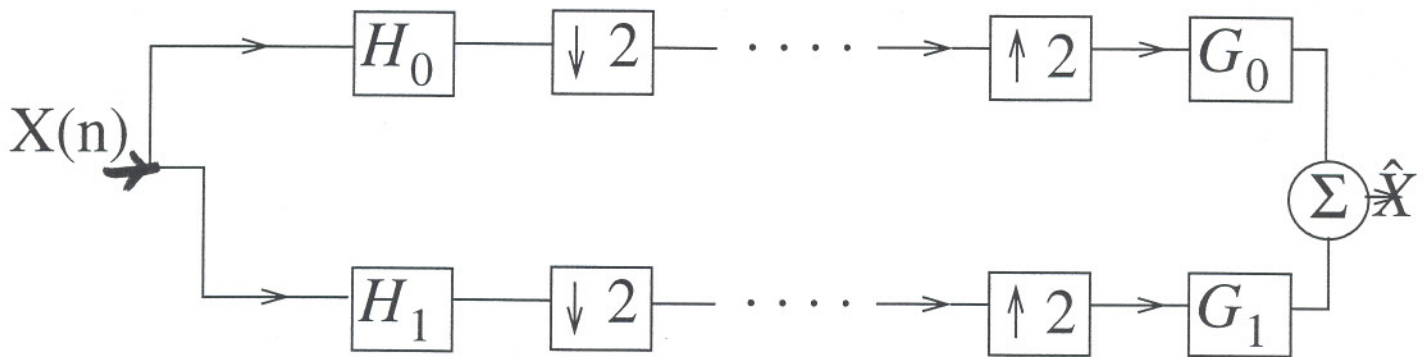
In addition to image coding, the Laplacian pyramid can also be used in other applications. For example, as we discussed above, the result of repetitive interpolation of $e_i(n_1, n_2)$ such that its size is the same as that of $f_0(n_1, n_2)$ can be viewed as approximately the result of filtering $f_0(n_1, n_2)$ with the Laplacian of a Gaussian. As we discussed in Section 8.3.3, zero-crossing points of the result of filtering $f_0(n_1, n_2)$ with the Laplacian of a Gaussian are the edge points in the edge detection method by Marr and Hildreth.

10.3.6 Adaptive Coding and Vector Quantization

The waveform coding techniques discussed in previous sections can be modified to adapt to changing local image characteristics. In a PCM system, the reconstruction levels can be chosen adaptively. In a DM system, the step size Δ can be chosen adaptively. In regions where the intensity varies slowly, for example, Δ can be chosen to be small to reduce granular noise. In regions where the intensity increases or decreases rapidly, Δ can be chosen to be large to reduce the slope overload distortion problem. In a DPCM system, the prediction coefficients and the reconstruction levels can be chosen adaptively. Reconstruction levels can also be chosen adaptively in a two-channel coder and a pyramid coder. The number of bits assigned to each pixel can also be chosen adaptively in all the waveform coders we discussed. In regions where the quantized signal varies very slowly, for example, we may want to assign a smaller number of bits/pixel. It is also possible to have a fixed number of bits/frame, while the bit rate varies at a pixel level.

In adaptive coding, the parameters in the coder are adapted based on some

Subband Coding



$$\hat{X}(\omega) = \frac{1}{2} [H_0(\omega)G_0(\omega) + H_1(\omega)G_1(\omega)]X(\omega) + \frac{1}{2} [H_0(\omega + \pi)G_0(\omega) + H_1(\omega + \pi)G_1(\omega)]X(\omega + \pi)$$

Consider QMF Filters:

$$\begin{cases} H_0(\omega) = G_0(-\omega) = F(\omega) \\ H_1(\omega) = G_1(-\omega) = e^{j\omega} F(-\omega + \pi) \end{cases}$$

$$\longrightarrow \hat{X}(\omega) = \frac{1}{2} [F(\omega)F(-\omega) + F(-\omega + \pi)F(\omega + \pi)]X(\omega)$$

$$\text{IMPOSE: } |F(\omega)|^2 + |F(\omega + \pi)|^2 = 2$$

$$\longrightarrow \hat{X}(\omega) = X(\omega) \longrightarrow \text{Perfect Reconstruction}$$

Filter Design:

- QMF filters:

$$h_1(n) = (-1)^n h_0(N - 1 - n)$$

N = # of taps

- Johnston's filter coefficients

$$h_0(N - 1 - n) = h_0(n)$$

—————> symmetric —————> NPR

8 tap Johnston filters:

$$h(0) = h(7) = 0.00938$$

$$h(1) = h(6) = 0.06942$$

$$h(2) = h(5) = -0.07065$$

$$h(3) = h(4) = 0.489980$$

Filter Design

- Cannot have linear phase FIR filters for QMF condition except for trivial 2 tap filter
→ amplitude distortion

- Well known filters

$$H_0(\omega) = A(\omega) \quad G_0(\omega) = B(\omega)$$

$$H_1(\omega) = e^{j\omega} B(\omega + \pi)$$

$$G_1(\omega) = e^{-j\omega} A(\omega + \pi)$$

$$a(n) = [1, 2, 1]$$

$$b(n) = [-1, 2, 6, 2, -1]$$

→ simple to implement
proposed by LeGall

Filter Design:

* Smith and Barnwell

$$h(0) = 0.03489$$

$$h(1) = -0.0109$$

$$h(2) = -0.0628$$

$$h(3) = 0.2239$$

$$h(4) = 0.55685$$

$$h(5) = 0.35797$$

$$h(6) = -0.0239$$

$$h(7) = -0.0759$$

Bit Allocation in Subband Coding:

R = Average # of bits per sample

R_K = Average # of bits per sample of subband K

M = # of subbands

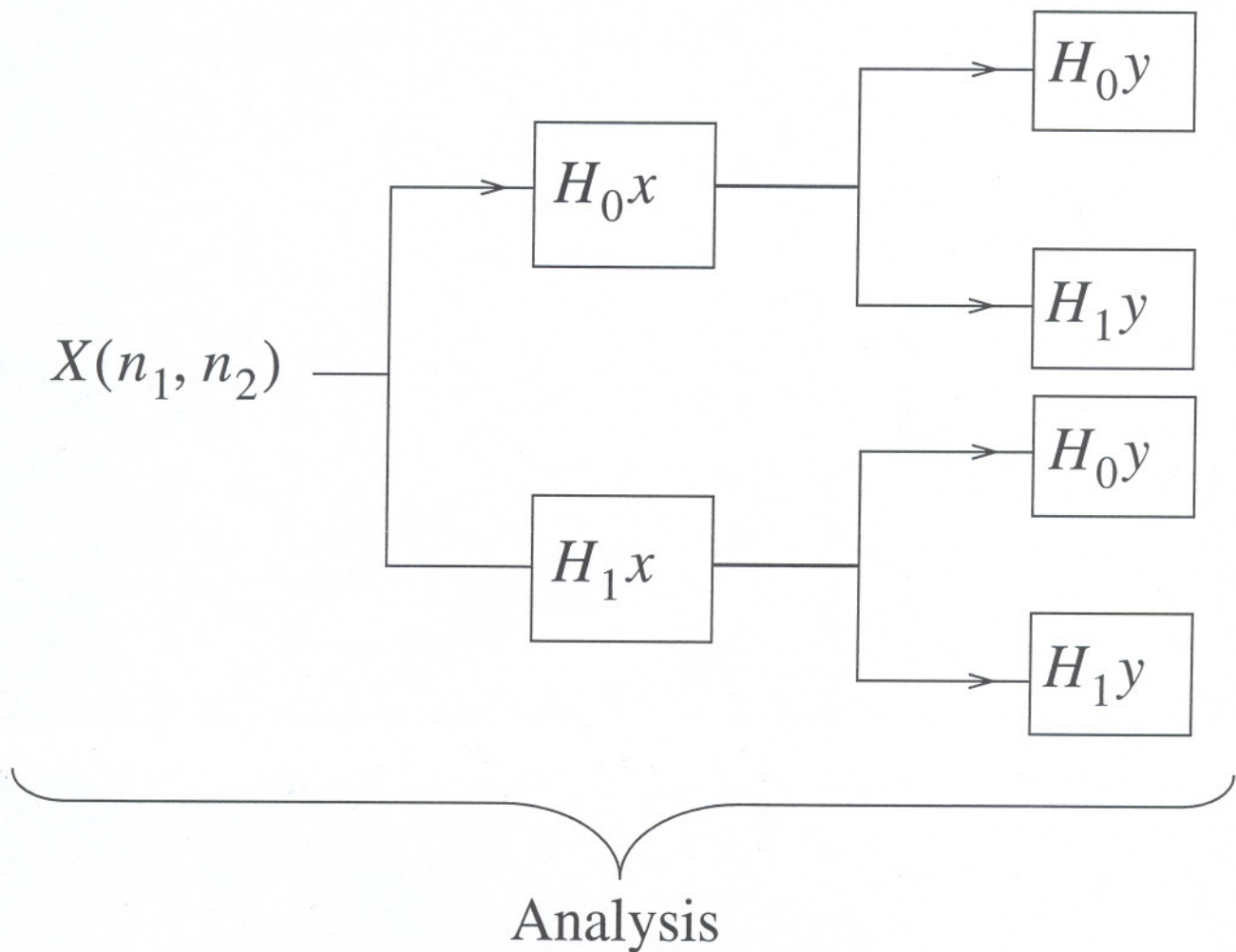
σ_K^2 = variance of coefficients in subband K :

$$R_K = R + \frac{1}{2} \log_2 \frac{\sigma_K^2}{\prod_{K=1}^M (\sigma_K^2)^{\frac{1}{M}}}$$

2D Subband Coding

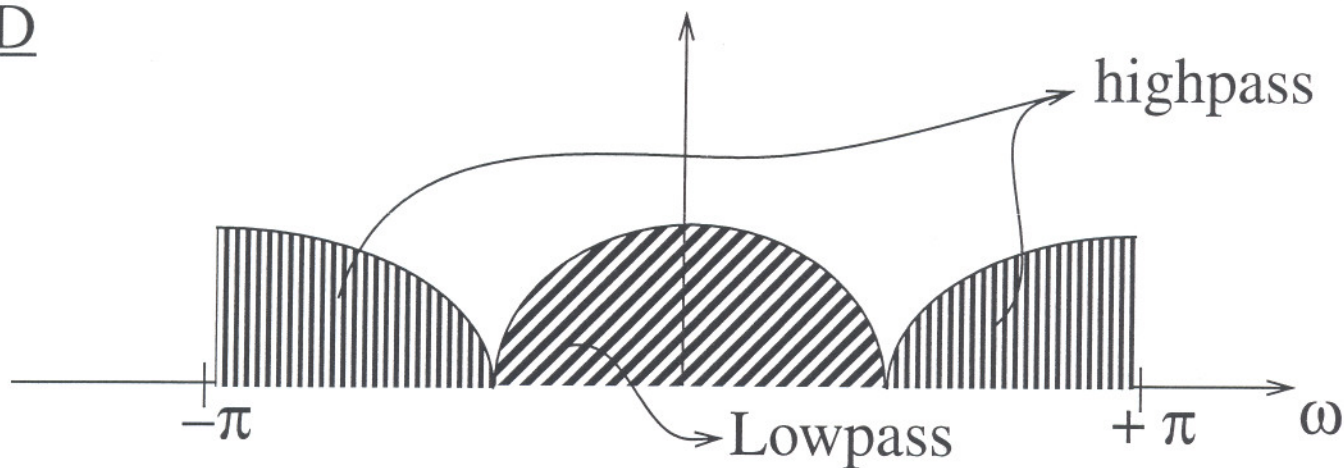
- Separable -----> Easy to implement
- Nonseparable

Separable subband Coding:

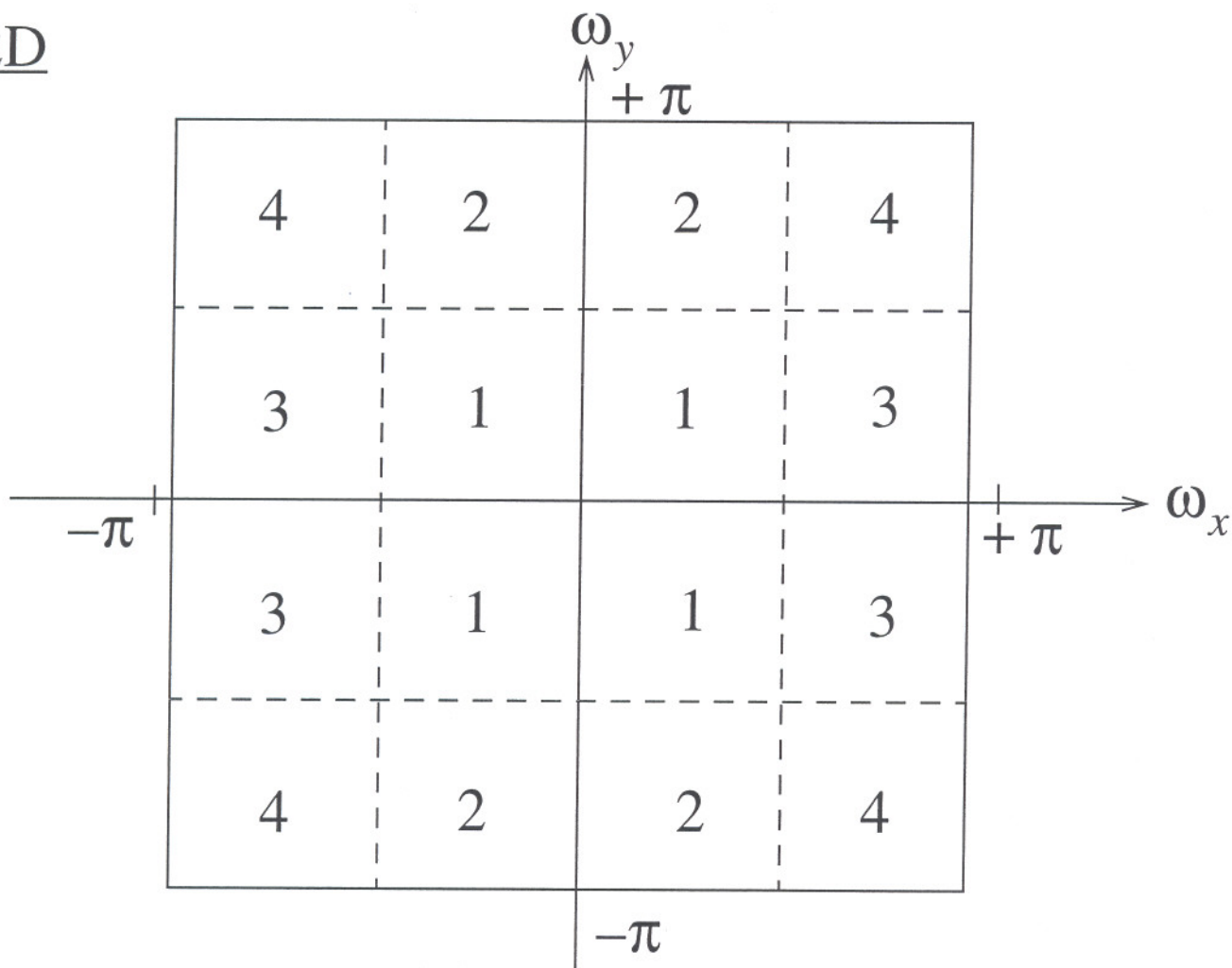


FREQUENCY DOMAIN

1D



2D

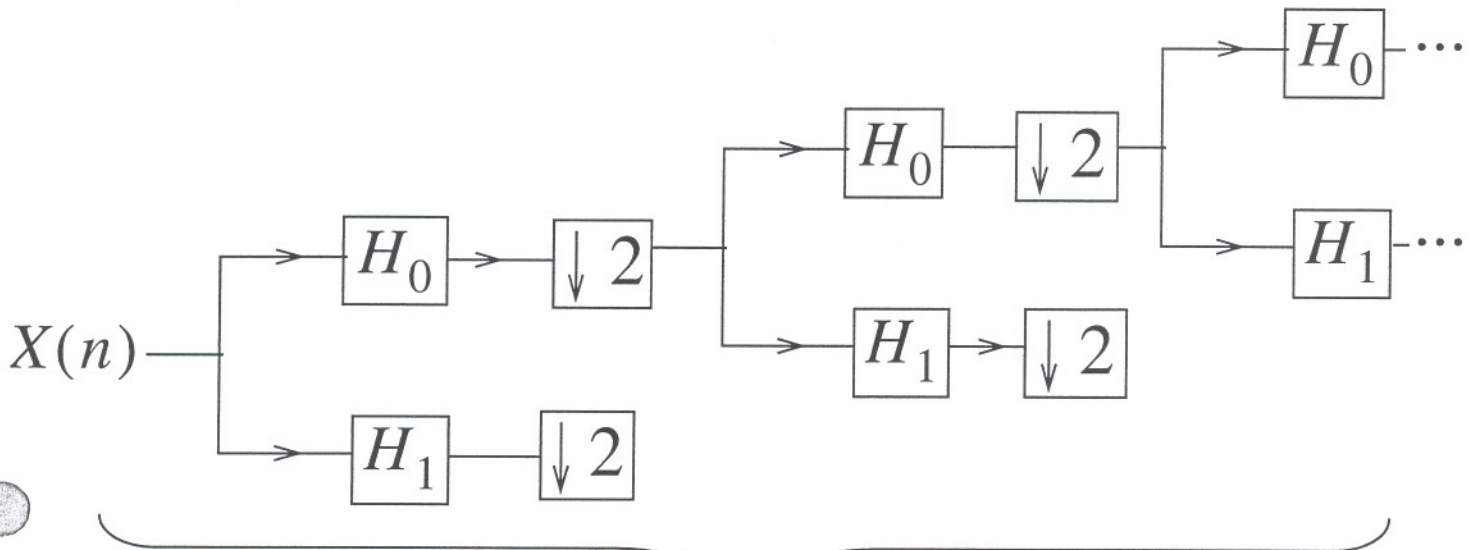


$$1 = L_x L_y$$
$$2 = L_x H_y$$

$$3 = H_x L_y$$
$$4 = H_x H_y$$

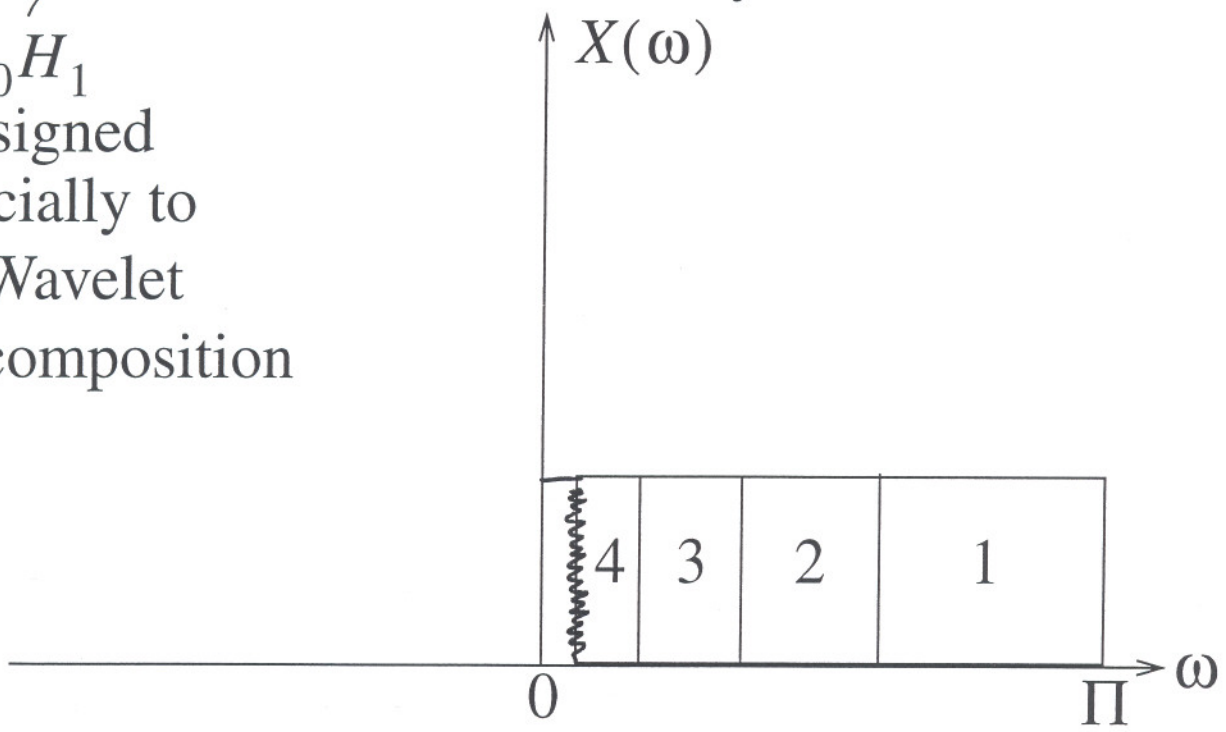
Wavelets

- A special kind of Subband Transform
- Historically developed independent of subband coding



$H_0 H_1$
Designed
specially to
be Wavelet
Decomposition

Analysis



Famous Wavelet Filters

- Daubechies
- Haar
- Coiflet

4 Tap Daubechies Low Pass

$$h(0) = 0.4829$$

$$h(1) = 0.8365$$

$$h(2) = 0.22414$$

$$h(3) = -0.1294$$