Subband Decomposition using Filter Banks

The analysis section consists of a set of filters (filter bank)
Subband Decomposition using Filter Banks

• Properties
  ➢ Progressive Hierarchical Transmission
  ➢ Redundancy reduction since image divided into almost mutually exclusive frequency bands (subband images which are inverse FT of the frequency bands) which have little mutual information in common (almost orthogonal)
    ⇒ Remaining redundancy mostly confined to each band
    ⇒ Redundancy reduction by orthogonalization
  ➢ Can exploit perceptual properties: HVS more sensitive to low frequency than to high frequency
    ⇒ coarser quantization in high-frequency bands
    ⇒ well suited for “perceptual coding”
Subband Decomposition using Filter Banks

• Filter banks (used in analysis/synthesis section)
  - Filter bank systems consist of an analysis section and a synthesis section:
    - Analysis section decomposes the input signal into subband signals
    - Synthesis section combines subband signals into the output reconstructed signal
  - If reconstructed signal is identical to input signal except for a possible delay, i.e., $x_r(n) = x(n-n_0) \Rightarrow$ perfectly reconstructing filter bank
  - If the total sampling rate in all the subbands (filter bank channels) is the same as the Nyquist sampling rate of the input signal ⇒ critically down-sampled filter bank ⇒ downsampling/upsampling factor equals number of channels
  - For some applications and for coding in particular, it is desirable to have critically downsampled perfectly reconstructing (or near perfectly reconstructing) filter banks
Subband Decomposition using Filter Banks

- **1D Description:** General N-Channel Filter bank

```
x(n)  \rightarrow \begin{array}{c}
      H_0(\omega) \\
      H_1(\omega) \\
      \vdots \\
      H_{N-1}(\omega) \\
    \end{array}
\rightarrow \begin{array}{c}
      \downarrow R \\
      \downarrow R \\
      \downarrow R \\
      \downarrow R \\
    \end{array}
\rightarrow \begin{array}{c}
      \text{Code} \\
      \text{Code} \\
      \text{Code} \\
      \text{Code} \\
    \end{array}
\rightarrow \begin{array}{c}
      \uparrow R \\
      \uparrow R \\
      \uparrow R \\
      \uparrow R \\
    \end{array}
\rightarrow \begin{array}{c}
      G_0(\omega) \\
      G_1(\omega) \\
      \vdots \\
      G_{N-1}(\omega) \\
    \end{array}
\rightarrow \sum \rightarrow x_r(n)
```

- BPF’s
- downsampler
- Code
- upsampler
- BPF’s

**Decimator**

**Interpolator**
Subband Decomposition using Filter Banks

- Critically downsampled $\Rightarrow R = N$
- Perfect reconstruction $\Rightarrow x_r(n) = x(n-n_0)$
  $\Rightarrow h_{FB}(n) = \delta(n-n_0)$
  $\Rightarrow H_{FB}(\omega) = e^{-j\omega n_0}$

✓ Subbands must form a “tiling” of the 1D or MD frequency domain without any gaps (and usually without overlap that cause redundancy)

✓ When using critical downsampling, the structure of each subband must be such that, when replicated, it forms a “tiling” of the 1D of MD frequency domain (without gaps and also usually without overlap except special types of overlap without redundancy)

Example: $N = 4$

![Diagram of subbands with frequencies H0, H1, H2, H3]
Subband Decomposition using Filter Banks

- 2D Description: Use separable filters, i.e.
  \[ H_{ij}(\omega_1,\omega_2) = H_i(\omega_1)H_j(\omega_2); \quad i=0,\ldots,N-1; \quad j=0,\ldots,N-1 \]
  \[ \Rightarrow \text{gives 16 bands (commonly used number)} \]
  - In practice we use Quadrature Mirror Filters (QMFs)
    - QMFs are two band filters

  \[ h_1(n) = (-1)^n h_0(n) \Rightarrow \text{need to design only } H_0 \]

\[ |H_0(\omega)|^2 + |H_1(\omega)|^2 = 1; \quad H_1(\omega) = H_0(\pi - \omega) \]
Subband Decomposition using Filter Banks

✓ Two-band systems are very simple and are often used

For perfect or near perfect reconstruction use QMFs. QMFs have the property that:

- Critical downsampling: QMFs decompose signal into 2 bands of equal width that can be downsampled by 2.

- Perfect Reconstruction: $x_r = x$ for ideal QMFs (perfect reconstruction)

Note: $x_r \approx x$ for designed QMFs (near perfect reconstruction)

Not able to achieve $|H_0(\omega)|^2 + |H_1(\omega)|^2 = 1$ in practice

$|H_0(\omega)|^2 + |H_1(\omega)|^2$ Distortion
Subband Decomposition using Filter Banks

- QMF filter banks widely used (Johnston, ICASSP 1980)
- Two band filter banks can be used to implement filter banks with more bands by using them successively in a tree structure. (We can get higher frequency resolution by using a tree structure, we thus can split each half-band into smaller bands)

Subbands are further subdivided by passing them through additional half-band filter banks after they have been downsampled.
Subband Decomposition using Filter Banks

- Subband bandwidth halved in each pass, making this a trivial approach for implementing uniform filter banks (constant Q filter banks) and octave band filter banks (non-uniform) using an incomplete binary tree

\[ x(n) \rightarrow H_0(\omega) \rightarrow 2\downarrow \rightarrow H_0(\omega) \rightarrow 2\downarrow \rightarrow H_0(\omega) \rightarrow 2\downarrow \rightarrow H_1(\omega) \rightarrow 2\downarrow \rightarrow H_1(\omega) \rightarrow 2\downarrow \rightarrow X(\omega) \]
Subband Decomposition using Filter Banks

- 2D Filter banks for subband coding
  - Problems
    - Difficult design: geometries constraints, PR
    - Difficult implementation
  - Separable Filter banks used
    - 2D Filter banks reduced to cascade of 1D Filter banks
    - For subband image coding, split rows (i.e. use horizontal filtering) then the columns (or vice versa)
Subband Decomposition using Filter Banks

Horizontal processing

Vertical processing

\[ x(n) \]

\[ H_0(\omega_1) \]

\[ H_1(\omega_1) \]

\[ \omega_2 \]

\[ \pi \]

\[ \frac{\pi}{2} \]

\[ 0 \]

\[ \omega_1 \]

\[ H_0(\omega_2) \]

\[ H_1(\omega_2) \]

\[ H_0(\omega_2) \]

\[ H_1(\omega_2) \]

\[ 2\downarrow \]

\[ LL \]

\[ LH \]

\[ HL \]

\[ HH \]
Subband Decomposition using Filter Banks

Example: Consider a 2-band split (rows, then columns)

1. \( X_{\text{DFT}} = \)
   \[
   \begin{array}{c|c}
   \hline
   \text{LP} & \text{HP} \\
   \hline
   \text{LP} & \text{HP} \\
   \hline
   \end{array}
   \]
   Take each row and put into 2-band split

2. Filter columns of result, \( Y \):

   \[
   \begin{array}{c|c}
   \hline
   \text{LP} & \text{HP} \\
   \hline
   \text{LP} & \text{HP} \\
   \hline
   \end{array}
   \]

   \[
   \begin{array}{c|c}
   \hline
   \text{LL} & \text{LH} \\
   \hline
   \text{HL} & \text{HH} \\
   \hline
   \end{array}
   \]

   By continuing successively ⇒

   Low in row \( \uparrow \) \hspace{1cm} High in row \( \uparrow \) \hspace{1cm} High in column
Image Filtering

- Note on filtering images

\[ x(n_1, n_2) \rightarrow H(\omega_1, \omega_2) \rightarrow y(n_1, n_2) \]

 Observation: We have an increase in total number of pixels as a result of filtering, which is not desirable. We need method to avoid this.
Image Filtering

- **Method 1: Use circular convolution** $x \ast_{\text{circular}} h$
  - Result same as extending image periodically and doing a linear convolution then keeping the samples in the main period of size $N \times N$
  - Usually in here we have discontinuity which introduces more error and visual artifacts near image border (leftmost)
  - Also in subband coding, we must code a discontinuity which results in more error (harder to do since less correlation)
Image Filtering

- **Method 2: Use symmetric extension**

  ✓ **Product with period** $2N$
  ✓ If filter is symmetric (commonly used), then the result is symmetric ⇒ only $N$ samples are required for representation
  ✓ Filtering may be performed using symmetric convolution based on the DCT ⇒ desirable since DCT is standard and available on chip.
  - **Complexity is the same for both methods but symmetric extension performs better** (in a subband image coder in particular) since it forces edges to be correlated (smooth transition) ⇒ easier to code
  - **Constraint to remember**: filter must be linear phase ⇒ $h(n_1,n_2)$
    conjugate symmetric
Subband Decomposition using Filter Banks

- Note on subband coding configurations:
  - Uniform band split (constant Q Filter banks)
    Example: 16 bands

  - Non-uniform band split (Octave-band Filter bank)
Multirate Signal Processing Basics - Review

- Highlights of Multirate Signal Processing
  - **Downsampler**
    - In **time domain**: \( y(n) = x(Rn) \)
    - Form \( y(n) \) by taking blocks of \( R \) samples; keep first sample of each block and discard the remaining \( R-1 \) samples (always start block at sample \( n = 0 \))
    - In **frequency domain**: 
      \[
      Y(\omega) = \frac{1}{R} \sum_{r=0}^{R-1} X\left(\frac{\omega}{R} + \frac{2\pi r}{R}\right)
      \]
      - attenuation
      - scaling
      - aliasing
Multirate Signal Processing Basics - Review

- **Decimation**

\[ y(n) = r[Rn] = \sum_{m} h(m)x(Rn - m) \]

\[ Y(\omega) = \frac{1}{R} \sum_{r=0}^{R-1} H\left(\frac{\omega}{M} + \frac{2\pi r}{M}\right)X\left(\frac{\omega}{M} + \frac{2\pi r}{M}\right) \]

- **Upsampling**

\[ \text{Meaning: insert } M-1 \text{ zeros between each 2 input samples (Always start at sample } n = 0) \]
Multirate Signal Processing Basics - Review

✓ **Time domain:**

\[
y(n) = \begin{cases} 
  x\left(\frac{n}{R}\right), & n = 0, \pm R, \pm 2R, \ldots \\
  0, & \text{otherwise}
\end{cases}
\]

\[
y(n) = \sum_{l=-\infty}^{\infty} x(l) \delta(n - lM)
\]

✓ **Frequency domain:**

\[
Y(\omega) = X(R\omega)
\]

No attenuation here, only scaling of the Fourier Transform.
Multirate Signal Processing Basics - Review

✓ Interpolation

\[ y(n) = \sum_{m=0,\pm R,\pm 2R,\ldots} x\left(\frac{m}{R}\right)h(n-m) \]

or

\[ y(n) = \sum_{l=-\infty}^{\infty} x(l)h(n-lM) \]

✓ Frequency domain

\[ Y(\omega) = X(R\omega)H(\omega) \]

imaging \hspace{1cm} \text{removes unwanted images}