Wavelets and Multiresolution Processing

Wavelets

- Fourier transform has it basis functions in sinusoids
- Wavelets based on small waves of varying frequency and limited duration
- In addition to frequency, wavelets capture temporal information
  - Bound in both frequency and time domains
  - Localized wave and decays to zero instead of oscillating forever
- Comparison with Fourier transform
  - Fourier transform used to analyze signals by converting signals into a continuous series of sine and cosine functions, each with a constant frequency and amplitude, and of infinite duration
  - Real world signals (images) have a finite duration and exhibit abrupt changes in frequency
  - Wavelet transform converts a signal into a series of wavelets
  - In theory, signals processed by wavelets can be stored more efficiently compared to Fourier transform
  - Wavelets can be constructed with rough edges, to better approximate real-world signals
  - Wavelets do not remove information but move it around, separating out the noise and averaging the signal
  - Noise (or detail) and average are expressed as sum and difference of signal, sampled at different points
    * In a picture, the signal is given by pixels
    * Average and detail are represented by sum and difference of pixels
    * Implemented with a low-pass filter for average and high-pass filter for detail
- Provide foundation for a new approach to signal processing and analysis called multiresolution
  - Concerned with the representation and analysis of images at more than one resolution
  - May be able to detect features at different resolutions
  - At the *finest scale*, average and detail are computed by sum and difference of neighboring pixels
  - We move to a *coarser level* by taking sum and difference of the previous levels in a recursive/iterative manner

Background

- Objects in images are connected regions of similar texture and intensity levels
- Use high resolution to look at small objects; coarse resolution to look at large objects
  - If you have both large and small objects, use different resolutions to look at them
  - Figure 7.1 – Local histogram can vary over different areas of images
- Wavelet properties
  - Two important properties: admissibility and regularity
  - Admissibility
    * Stated as

$$\int_{-\infty}^{\infty} \frac{|\Psi(\omega)|^2}{|\omega|} d\omega < \infty$$

      where $\psi(t)$ is a wave in the time domain, and $\Psi(\omega)$ is the Fourier transform of $\psi(t)$

        * In practice, $\Psi(\omega)$ will always have sufficient decay so that the admissibility criterion reduces to the requirement that $\Psi(0) = 0$, or

$$\int_{-\infty}^{\infty} \psi(t)dt = \Psi(0) = 0.$$

        * Each wavelet transform must meet the requirement that it should integrate to zero
            · The transform *waves* above and below the x-axis and the average value of the wavelet in time domain must be zero
            · In addition, the transform is well localized in the time domain
        * A wavelet is defined over time t, $0 \leq t \leq N$
            · Provides a set of basis functions $\psi_{jk}(t)$ in continuous time
            · $\psi_{jk}(t)$ is a set of linearly independent functions that can be used to produce all admissible functions $f(t)$
            · The expression

$$f(t) = \sum_{j,k} b_{jk} \psi_{jk}(t)$$

where $\psi_{jk} = \psi(2^j \cdot t - k)$ indicates a wavelet that has been compressed j times and shifted k times, and b_{jk} is a coefficient
            · The shifted wavelet $\psi_{0k} = \psi(t - k)$ is defined over $k \leq t \leq k + N$, implying that the signal is shifted to the right (translated) by k
            · The rescaled wavelets $\psi_{j0} = \psi(2^j \cdot t)$ are defined over $0 \leq t \leq \frac{N}{2^j}$ implying that the signal is compressed by a factor of 2^j
    – Regularity
        * Imposed to ensure that the wavelet transform decreases quickly with decreasing scale
        * This condition also states that the wavelet function should have some smoothness and concentration in both time and frequency domains
    – Taken together, admissibility and regularity form the components *wave* and *let* in wavelet, respectively
        * *let* implies quick decay

- Image pyramids
    – Structure to represent images at more than one resolution
    – Collection of decreasing resolution images arranged in the shape of a pyramid
    – Figure 7.2a
        * Highest resolution image at the pyramid base
        * As you move up the pyramid, both size and resolution decrease
        * Base level of size $2^J \times 2^J$
        * General level j of size $2^j \times 2^j$, $0 \leq j \leq J$
        * Pyramid may get truncated at level P, $0 \leq P \leq J$
        * Number of pixels in a pyramid with $P + 1$ levels ($P > 0$) is

$$N^2 \left(1 + \frac{1}{4^1} + \frac{1}{4^2} + \cdots + \frac{1}{4^P} \right) \leq \frac{4}{3} N^2$$

    – Figure 7.2b
        * Building image pyramids
        * Level $j - 1$ *approximation* output provides the images needed to build an approximation pyramid
        * Level j *prediction residual* output is used to build a complementary *prediction residual pyramid*
    – Both approximation and prediction residual pyramids are computed in an iterative fashion
    – Three step procedure

1. Compute a reduced-resolution approximation of level j input image; done by filtering and downsampling the filtered result by a factor of 2; place the resulting approximation at level $j-1$ of approximation pyramid
2. Create an estimate of level j input image from the reduced resolution approximation generated in step 1; done by upsampling and filtering the generated approximation; resulting prediction image will have the same dimensions as the level j input image
3. Compute the difference between the prediction image of step 2 and input to step 1; place the result in level j of prediction residual pyramid

- Variety of approximation and interpolation filters
  * Neighborhood averaging producing mean pyramids
  * Lowpass Gaussian filtering producing Gaussian pyramids
  * No filtering producing subsampling pyramids
  * Interpolation filter can be based on nearest neighbor, bilinear, and bicubic
- Upsampling
  * Doubles the spatial dimensions of approximation images
  * Given an integer n and 1D sequence of samples $f(n)$, upsampled sequence is given by

$$f_{2\uparrow}(n) = \begin{cases} f(n/2) & \text{if } n \text{ is even} \\ 0 & \text{otherwise} \end{cases}$$

  * Insert a 0 after every sample in the sequence
- Downsampling
  * Halves the spatial dimensions of the prediction images
  * Given by

$$f_{2\downarrow}(n) = f(2n)$$

  * Discard every other sample
- Figure 7.3
  * Approximation pyramid produced by low-pass Gaussian smoothing
  * Lower-resolution levels can be used for the analysis of large structures; higher resolution images appropriate for analyzing individual object characteristics
  * Prediction residual levels produced by bilinear interpolation
  * Residual pyramid can be used to generate the complementary approximation pyramid without error
    · Begin with a level $j \times j$ image
    · Predict the level $(j+1) \times (j+1)$ image by upsampling and filtering
    · Add the level $j+1$ prediction residual
    · Prediction residual histogram in Figure 7.3b is highly peaked around zero; approximation histogram is not
    · Prediction residuals are scaled to make small prediction erros more visible

• Subband coding
- Subbands
  * A set of band-limited components as a result of decomposing an image
  * Decomposition performed such that subbands can be reassembled to reconstruct the original image without error
- Digital filter in Figure 7.4a
  * Built from three basic components: unit delays, multipliers, and adders
  * Unit delays are connected in series to create $K-1$ delayed (right shifted) versions of the input sequence $f(n)$

* Delayed sequence $f(n-2)$ is given by

$$f(n-2) = \begin{cases} & \vdots \\ f(0) & \text{for } n = 2 \\ f(1) & \text{for } n = 2+1 = 3 \\ & \vdots \end{cases}$$

* Input sequence $f(n) = f(n-0)$
* $K-1$ delayed sequences at the outputs of unit delays
* Delayed sequences multiplied by constants $h(0), h(1), \ldots, h(K-1)$ (filter coefficients) and summed to produce the filtered sequence

$$\begin{aligned} \hat{f}(n) &= \sum_{k=-\infty}^{\infty} h(k) f(n-k) \\ &= f(n) \star h(n) \end{aligned}$$

* Each coefficient defines a *filter tap*; filter is of order K
* If the input to the filter of Figure 7.4a is the unit discrete impulse of Figure 7.4b, we have

$$\begin{aligned} \hat{f}(n) &= \sum_{k=-\infty}^{\infty} h(k) \delta(n-k) \\ &= h(n) \end{aligned}$$

  · Substitute $\delta(n)$ for $f(n)$
  · Make use of sifting property of the unit discrete impulse
  · Impulse response of the filter is the K-element sequence of filter coefficients
  · Unit impulse is shifted from left to right across the top of the filter (delays)
  · There are K coefficients; impulse response is of length K, and filter is called a *finite impulse response* (FIR) filter
* Figure 7.5

– Two components of wavelet as analysis and synthesis
  * Two-band subband coding
  * Figure 7.6a – two filter banks; each containing two FIR filters
– Analyzing wavelet
  * Analog bandpass filter with its properties of scaling and translation
  * Facilitate implementation as a convolution operation
  * Analysis filter bank (filters $h_0(n)$ and $h_1(n)$ used to break input sequence $f(n)$ into two half-length sequences $f_{\text{lp}}(n)$ and $f_{\text{hp}}(n)$
– Synthesizing wavelet
  * Along with a scaling (smoothing) function, used to represent a signal from its lowpass features (background) and bandpass details (high frequency)
– Need to build a pair of analyzing and synthesizing wavelets, as well as a pair of scaling functions (lowpass and smoothing) so that the input and reconstructed signals remain the same
– *Orthogonality*
  * Property of wavelets such that their inner products are zero
  * Mathematically,

$$\int_{-\infty}^{\infty} \psi_{jk}(t) \cdot \psi_{j'k'}(t) dt = 0$$

- Orthogonal basis
  - Formed by wavelets for the space of admissible functions
  - Leads to a simple formula for the coefficient b_{jk}; defined earlier as

$$f(t) = \sum_{j,k} b_{jk}\psi_{jk}(t)$$

  - Multiplying above expression on both sides by $\psi_{j'k'}(t)$ and integrating, we have

$$\int_{-\infty}^{\infty} f(t)\psi_{j'k'}(t)dt = \int_{-\infty}^{\infty} \sum_{j,k} b_{jk}\psi_{jk}(t)\psi_{j'k'}(t)dt$$

  - Orthogonality property eliminates the integrals of the terms where $j \neq j'$ and $k \neq k'$; we get

$$\int_{-\infty}^{\infty} f(t)\psi_{j'k'}(t)dt = b_{jk}\int_{-\infty}^{\infty} (\psi_{j'k'}(t))^2 dt$$

    yielding the coefficient b_{jk} as

$$b_{jk} = \frac{\int_{-\infty}^{\infty} f(t)\psi_{j'k'}(t)dt}{\int_{-\infty}^{\infty} (\psi_{j'k'}(t))^2 dt}$$

Multiresolution

- Scaling function $\phi(2^j \cdot t - k)$ provides the basis for a set of signals (or average) at level j
- Similarly, the wavelet function $\psi(2^j \cdot t - k)$ provides the detail at level j
- Addition of ϕ and ψ at level j yields the signal at level $j+1$ providing for multiresolution,

$$\phi(2^j \cdot t - k) + \psi(2^j \cdot t - k) \Rightarrow \phi(2^{j+1} \cdot t - 2k)$$

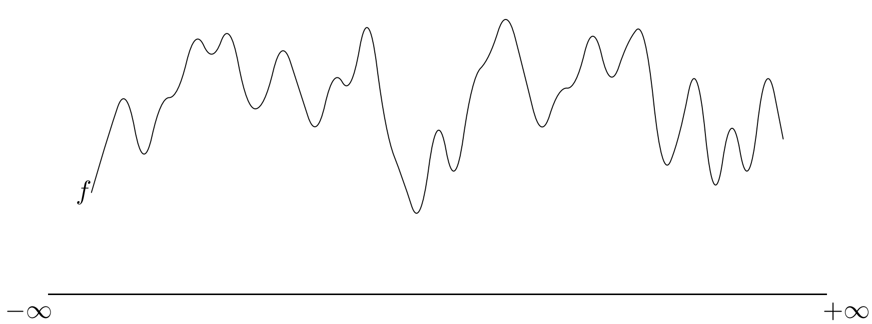
- Applying the above approach to all the signals at level j, we have

$$V_j \oplus W_j = V_{j+1}$$

  where V_j and W_j are the scaling space and wavelet space at level j
- Input signal is divided into different scales of resolution, rather than different frequencies
- Wavelets automatically match long time with low frequency and short time with high frequency

Haar Wavelet

- Consider a signal f in one dimension from $-\infty$ to $+\infty$



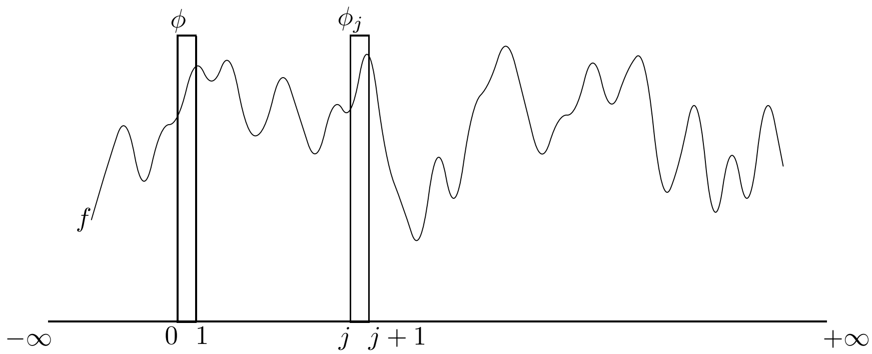
- Haar scaling function is denoted by $\phi(t)$ and Haar wavelet function is denoted by $\psi(t)$.
- Haar scaling function (averaging or lowpass filter) at level 0 (in the original signal) is given by

$$\phi(x) = \begin{cases} 1 & 0 \leq x < 1 \\ 0 & \text{otherwise} \end{cases}$$

- Translation by j is denoted by $\phi_j(x)$

$$\phi_j(x) = \phi(x - j)$$

- Figure below shows both $\phi(x)$ and $\phi_j(x)$



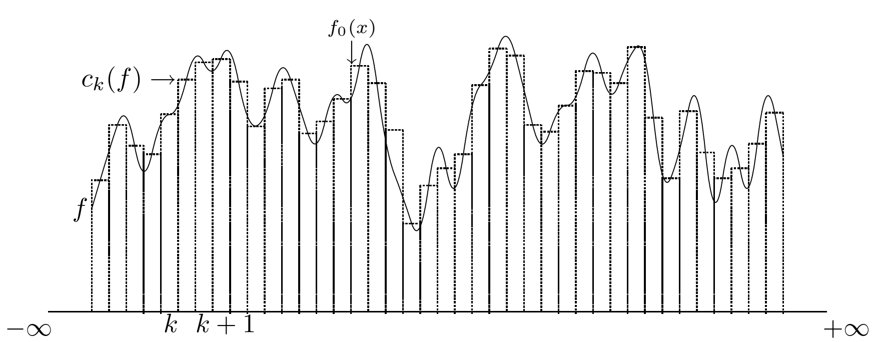
- Coefficients of the signal f indexed by j are given by

$$\begin{aligned} c_j(f) &= \int f(x)\phi_j(x)dx \\ &= \text{Average of } f \text{ over the interval } [j, j+1] \end{aligned}$$

- An approximate reconstruction of f from $c_j(f)$ is given by

$$f_0(x) = \sum_j c_j(f)\phi_j(x)$$

- Reconstruction of the signal



  - Ideally, we'll like to have a better resolution for sampling in Figure above and go to an appropriately finer scale
  - However, in images, the finest scale is given by the pixel, and we start at this level.
    - Sums and differences of neighboring pixels are considered to be at finest scale.

- Next, we go to a coarser level using the family $\{\phi_j^{(1)}\}_j$ where

$$\phi_j^{(1)}(x) = \phi\left(\frac{1}{2}x - j\right).$$

- Note that

$$\phi\left(\frac{1}{2}x - j\right) = \begin{cases} 1 & 2j \le x < 2(j+1) \\ 0 & \text{otherwise} \end{cases}$$

- Signals at level 1 are given by

$$\begin{aligned} c_j^1(f) &= \frac{1}{2}\int f(x)\phi_j^{(1)}(x)dx \\ &= \text{average of } f \text{ over } [2j, 2(j+1)] \end{aligned}$$

- Averaging over larger interval leads to a loss of information (detail)
  - Lost detail is preserved in wavelet transform
  - $\phi^{(0)}$ refers to ϕ at level 0, the original level.
  - Since $\phi_j^{(1)} = \phi_{2j}^0 + \phi_{2j+1}^0$, we see that

$$c_j^{(1)} = \frac{c_{2j}^{(0)} + c_{2j+1}^{(0)}}{2}.$$

  - Detail is preserved by introducing a new coefficient (highpass filter)

$$d_j^{(1)} = \frac{c_{2j}^{(0)} - c_{2j+1}^{(0)}}{2}$$

  - It is apparent that

$$\begin{aligned} c_j^{(1)} + d_j^{(1)} &= c_{2j}^{(0)} \\ c_j^{(1)} - d_j^{(1)} &= c_{2j+1}^{(0)} \end{aligned}$$

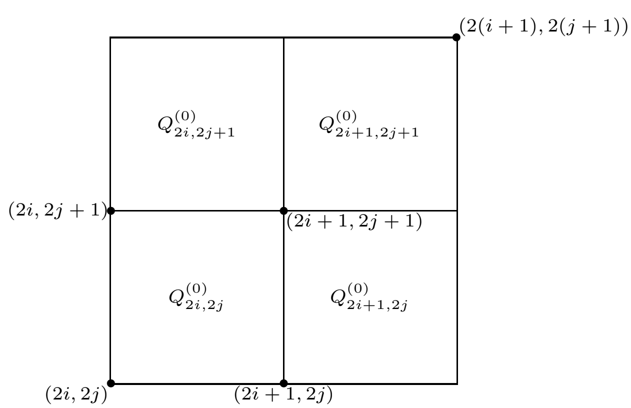
  - The average (ψ) and detail (d) coefficients for Haar wavelet at level 1 are given by

$$\begin{aligned} \psi_j^{(1)} &= \tfrac{1}{2}\left[\phi_{2j}^{(0)} - \phi_{2j+1}^{(0)}\right] \\ d_j^{(1)} &= \int f(x)\psi_j^{(1)}(x)dx \end{aligned}$$

  - Notice from above equation that the wavelet transform of one-dimensional signal is two-dimensional.

Extension of Haar wavelet to a signal in two dimensions

- Consider a sample of the 2D image as a box as shown below



- Sample is divided into four areas (squares)
- Q represents the signal coefficients
- Let (l,p) represent the center coordinates $(2i+1, 2j+1)$ in the sample
- The Haar coefficient is given by

$$C_{l,p}(f) = \int\int f(x,y)\chi_{Q^{(0)}_{l,p}}(x,y)dxdy$$

where the characteristic χ of Q at level 0 is given by

$$\phi^{(1)}_{i,j}(x,y) = \chi_Q(x,y) = \begin{cases} 1 & (x,y) \in Q \\ 0 & (x,y) \notin Q \end{cases}$$

$$Q^{(1)}_{j} = \bigcup_{\substack{l = 2i, 2i+1 \\ p = 2j, 2j+1}} Q^{(0)}_{lp}$$

- Also, with

$$Q^{(1)}_{(i,j)} = \left\{ (x,y) \middle| \begin{array}{l} 2i \le x < 2(i+1) \\ 2j \le y < 2(j+1) \end{array} \right\}$$

the Haar coefficient at level 1 is given by

$$\begin{aligned} C^{(1)}_{(i,j)}(f) &= \frac{1}{4}\int\int_{Q^{(1)}_{(i,j)}} f(x,y)dxdy \\ &= \int\int \phi^{(1)}_{i,j}(x,y)f(x,y)dxdy \end{aligned}$$

- The average and detail coefficients are now given by

$$\begin{aligned} C^{(1)}_{(i,j)}(f) &= C^{(0)}_{2i,2j}(f) + C^{(0)}_{2i+1,2j}(f) + C^{(0)}_{2i,2j+1}(f) + C^{(0)}_{2i+1,2j+1}(f) \\ D^{(1)(0,1)}_{(i,j)}(f) &= C^{(0)}_{2i,2j}(f) + C^{(0)}_{2i+1,2j}(f) - C^{(0)}_{2i,2j+1}(f) - C^{(0)}_{2i+1,2j+1}(f) \\ D^{(1)(1,0)}_{(i,j)}(f) &= C^{(0)}_{2i,2j}(f) - C^{(0)}_{2i+1,2j}(f) + C^{(0)}_{2i,2j+1}(f) - C^{(0)}_{2i+1,2j+1}(f) \\ D^{(1)(1,1)}_{(i,j)}(f) &= C^{(0)}_{2i,2j}(f) - C^{(0)}_{2i+1,2j}(f) - C^{(0)}_{2i,2j+1}(f) + C^{(0)}_{2i+1,2j+1}(f) \end{aligned}$$

  * Notice from the above equation that the wavelet transform of a two-dimensional signal is in four dimensions
- Adding the four coefficients in the above equation, we get

$$C^{(1)}_{(i,j)}(f) + \sum D^{(1)(\alpha,\beta)}_{(i,j)}(f) = C^{(0)}_{2i,2j}$$

$$\begin{aligned} \phi^{(1)}_{ij}(x,y) &= \sum_{l=2i}^{2i+1}\sum_{p=2j}^{2j+1} \phi^{(0)}_{l,p}(x,y) \\ \psi^{(}_{(i,j,k)}1)(x,y) &= \frac{1}{4}\sum_{l=2i}^{2i+1}\sum_{p=2j}^{2j+1} (\xi_{l,p,k})\phi^{(0)}_{l,p}(x,y) \end{aligned}$$

$\xi_{l,p,k}$					
$k \rightarrow$		0	1	2	3
l	$2i$	1	1	1	1
	$2i+1$	1	1	-1	-1
p	$2j$	1	-1	1	-1
	$2j+1$	1	-1	-1	1

$$d^{(1)}_{i,j,k} = \int f(x,y)\psi^{(1)}_{(i,j,k)}(x,y)dxdy$$

- $k = 0$ corresponds to

$$\phi^{(1)}_{(2i,2j)}(x,y) = \frac{1}{4}\phi\left(\frac{1}{2}x - i, \frac{1}{2}y - j\right)$$

Discrete Wavelet Transform

- CWT is redundant as the transform is calculated by continuously shifting a continuously scalable function over a signal and calculating the correlation between the two
- The discrete form is normally a [piecewise] continuous function obtained by sampling the time-scale space at discrete intervals
- The process of transforming a continuous signal into a series of wavelet coefficients is known as *wavelet series decomposition.*
- Scaling function can be expressed in wavelets from $-\infty$ to j
- Adding a wavelet spectrum to the scaling function yields a new scaling function, with a spectrum twice as wide as the first
  - Addition allows us to express the first scaling function in terms of the second
  - The formal expression of this phenomenon leads to multiresolution formulation or two-scale relation as

$$\phi(2^j t) = \sum_k h_{j+1}(k)\phi(2^{j+1}t - k)$$

  - This equation states that the scaling function (average) at a given scale can be expressed in terms of translated scaling functions at the next smaller scale, where the smaller scale implies more detail
  - Similarly, the wavelets (detail) can also be expressed in terms of translated scaling functions at the next smaller scale as

$$\psi(2^j t) = \sum_k g_{j+1}(k)\phi(2^{j+1}t - k)$$

  - The functions $h(k)$ and $g(k)$ are known as *scaling filter* and *wavelet filter*, respectively
    * These filters allow us to implement the *discrete wavelet transform* (DWT) as an iterated digital filter bank.
- *Subsampling property*
  - Gives a step size of 2 in the variable k for scaling and wavelet filters
  - Every iteration of filter banks reduces the number of samples by half so that in the
    * In the last case, we are left with only one sample

Implementation of Haar Wavelets

- Any wavelet implemented by the iteration of filters with rescaling
  - Set of filters form the *filter bank*
  - Let k be an integer

- Averaging and detail filters implemented using two $2^{k-1} \times 2^k$ filtering matrices H and G given by

$$H = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \qquad G = \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \frac{1}{2} & -\frac{1}{2} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \frac{1}{2} & -\frac{1}{2} \end{bmatrix}$$

- Let H^t and G^t denote the transpose of H and G, respectively
- Let I_k denote a $2^k \times 2^k$ identity matrix
- Then, the following facts about H and G are true:

$$\begin{aligned} H^t \times H + G^t \times G &= \tfrac{1}{2} I_k \\ H \times H^t = G \times G^t &= \tfrac{1}{2} I_{k-1} \\ H \times G^t = G \times H^t &= 0 \end{aligned}$$

- For simplicity, consider the original signal to be sampled as a vector of length 2^k
- The filtering process includes downsampling ($\downarrow 2$) and decomposes b into two vectors b_1 (for block average) and d_1 (for detail) given by

$$\begin{aligned} b_1 &= H \times b \\ d_1 &= G \times b \end{aligned}$$

- b_1 and d_1 can be combined to reconstruct the original signal b

$$b = 2 \times (H^t \times b_1 + G^t \times d_1)$$

  * A lossy compression can be achieved by discarding the detail vector d_1, or setting it to be zero.

• Haar filter is applied to an image by the application of H and G filters in a tensorial way

- Let P be a picture image represented as an $r \times c$ matrix of pixels
- Applying the H filter to P, we get a new image P' as

$$P' = H \times P \times H^t$$

- P' is an $r' \times c'$ matrix such that

$$\begin{aligned} r' &= \frac{r}{2} \\ c' &= \frac{c}{2} \end{aligned}$$

- Application of H and G filters results into four matrices given by

$$\begin{aligned} P_{11} &= H \times P \times H^t \\ P_{12} &= H \times P \times G^t \\ P_{21} &= G \times P \times H^t \\ P_{22} &= G \times P \times G^t \end{aligned}$$

  * P_{11} is called the *fully averaged picture*
  * P_{12} and P_{21} are called *partially averaged* and *partially differenced pictures*
  * P_{22} is called the *fully differenced picture*
- The four components can be used to reconstruct the original image P as

$$P = \begin{bmatrix} H^t & G^t \end{bmatrix} \times \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} \times \begin{bmatrix} H \\ G \end{bmatrix}$$

    * Above equation is known as a *synthesis filter bank*

- The matrix $[HG]^t$ is orthogonal as its inverse is the transpose, or

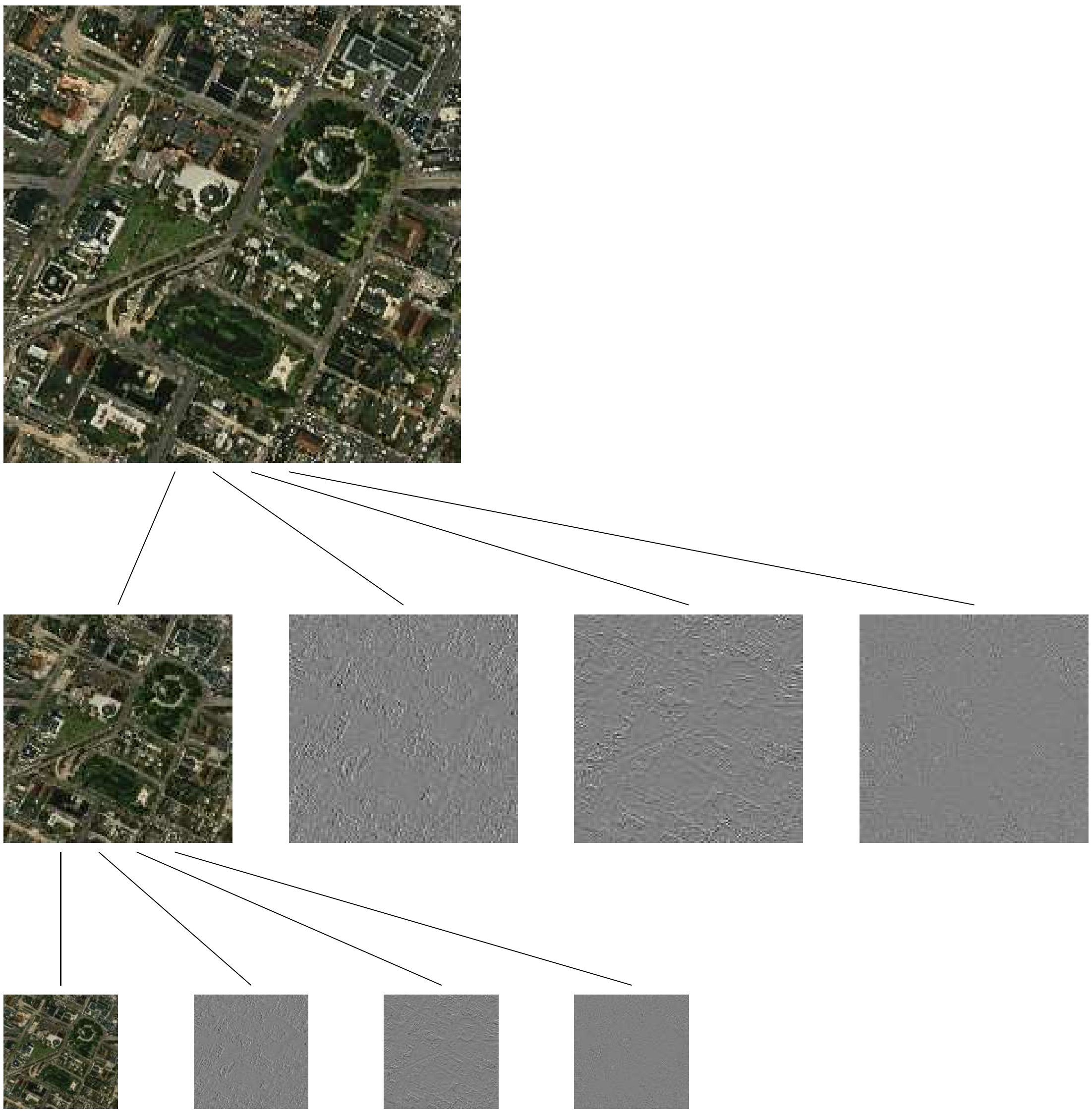
$$\begin{bmatrix} H \\ G \end{bmatrix}^{-1} = \begin{bmatrix} H^t G^t \end{bmatrix}$$

    * Matrices in synthesis bank are also known as *orthogonal filter bank*
    * Note that

$$\begin{bmatrix} H^t G^t \end{bmatrix} \begin{bmatrix} H \\ G \end{bmatrix} = H^t H + G^t G = I$$

    * The synthesis bank is the inverse of the analysis bank
    * Analysis bank contains the steps for filtering and downsampling
    * Synthesis bank reverses the order and performs upsampling and filtering

- Analysis of a picture (for two levels) is shown below



– Figure below shows an original texture, and its compression and reconstruction

Original image

Compression by one level

Compression by two levels

Reconstructed image

  * Top left shows the original 256×256 pixel texture
  * Application of Haar wavelet results into four 128×128 pixel components which are combined into a 256×256 pixel image shown on top right
    - Top left quarter of this image shows the fully averaged part
    - Top right quarter contains the partially averaged part
    - Bottom left quarter contains the partially differenced part
    - Bottom right quarter contains the fully differenced component
  * Haar wavelet is applied to the fully averaged part again and the assembled components are shown in the bottom left picture
  * This picture is then used for reconstruction of the texture and the reconstructed texture is shown in the bottom right picture.

- Lossy compression is achieved by discarding the differenced pictures (setting the matrices to zero) and retaining only P_{11} during the reconstruction phase
  - The process can be carried through several processing steps, thus removing a large amount of detail information.

Other wavelets

- Haar wavelet transform, as described above, may not be able to take good advantage of the continuity of pixel values within images

- Other wavelets may perform better at this, and achieve higher compression of textures, specially if the textures are smooth images.

JPEG 2000 Standard

- JPEG2000 standard is based on wavelets to achieve compression
- Divides an image into two-dimensional array of samples, known as *components*
  - As an example, a color image may consist of several components representing base colors red, green, and blue
- Image and its components are decomposed into rectangular *tiles*, which form the basic unit of original or reconstructed image
- All the components (for example different color components) that form a tile are kept together so that each tile can be independently extracted/decoded/reconstructed.
- Tiles are analyzed using wavelets to create multiple decomposition levels
  - Yields a number of coefficients to describe the horizontal and vertical spatial frequency characteristics of the original tiles, within a local area.
  - Different decomposition levels are related by powers of 2
- Information content of a large number of small-magnitude coefficients is further reduced by quantization, giving *code-blocks*
- Additional compression is achieved by entropy coding of bit-planes of the coefficients in code-blocks to reduce the number of bits required to represent quantized coefficients
- JPEG2000 allows the formation of regions of interest (ROI) by selective coding of some coefficients.