Wavelet Transform and Scalogram

Instead of using windowed sinusoids as TF atoms, consider the wavelet atom

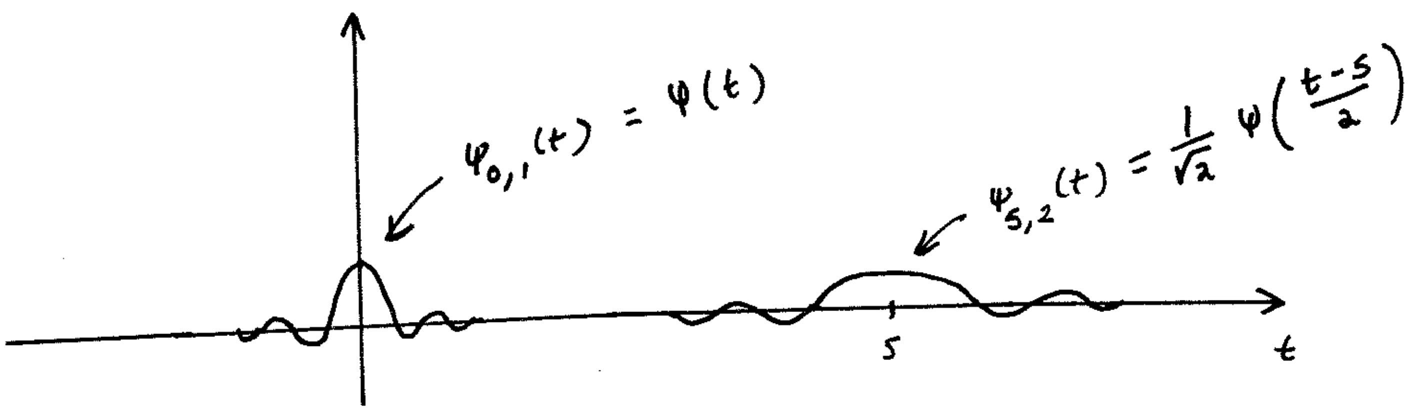
$$\psi_{u,s}(t) = \frac{1}{\sqrt{s}}\,\psi\left(\frac{t-u}{s}\right)$$

where the mother wavelet $\psi(t)$ satisfies

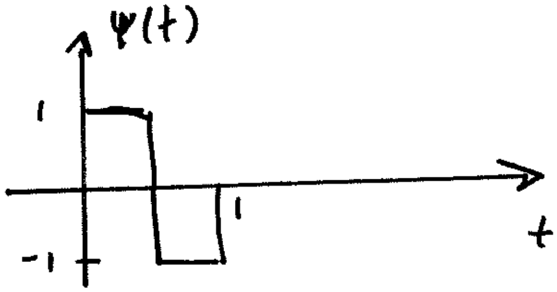
$$\int \psi(t)\,dt = 0$$

$$\int |\psi(t)|^2\,dt = 1$$

Ex.



Ex. Haar wavelet



Ex. "Mexican Hat" wavelet

$$\psi(t) = \frac{2}{\pi^{1/4}\sqrt{3\sigma}} \left(\frac{t^2}{\sigma^2} - 1 \right) \exp\left(- \frac{t^2}{2\sigma^2} \right)$$

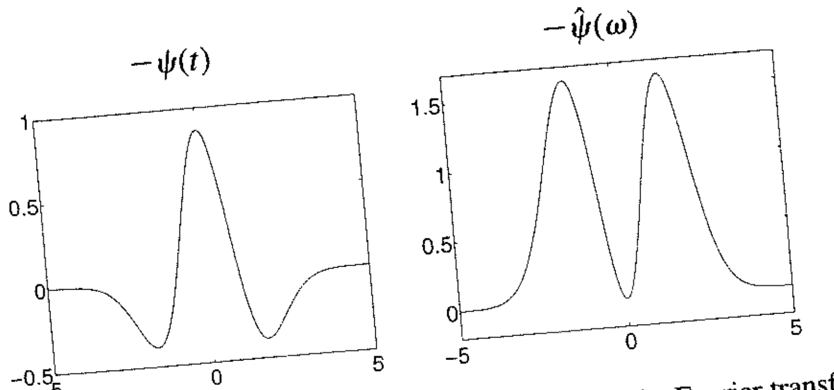


FIGURE 4.6 Mexican hat wavelet (4.35) for $\sigma = 1$ and its Fourier transform.

The wavelet transform is defined as

$$W x(u,s) = \langle x, \psi_{u,s} \rangle$$

$$= \int_{-\infty}^{\infty} x(t) \frac{1}{\sqrt{s}} \psi^*\left(\frac{t-u}{s}\right) dt$$

The <u>scalogram</u> is the time-frequency energy distribution associated with the wavelet transform:

$$P_{W_x}(u,s) = \left| \int_{-\infty}^{\infty} x(t) \frac{1}{\sqrt{s}} \psi^*\left(\frac{t-u}{s}\right) dt \right|^2$$

<u>Time-Frequency Resolution of Scalogram</u>

A change in scale s does not correspond to a simple shift in frequency since

$$\text{frequency} \propto \frac{1}{\text{scale}}$$

Also, a change in scale compresses or dilates $\psi(t)$, thus changing the temporal concentration. In other words, scale has both a time and frequency interpretation, and consequently the time-frequency concentration depends on scale.

Let $\psi_{u,s}(t) = \frac{1}{\sqrt{s}} \psi\left(\frac{t-u}{s}\right)$. Then

$$\Psi_{u,s}(f) = \sqrt{s}\, \Psi(sf)\, e^{-j2\pi f u}$$

$$\text{where } \Psi(f) = \int_{-\infty}^{\infty} \psi(t) e^{-j2\pi f t}$$

Now if $\psi(t)$ is centered at $t=0$, it follows that $\psi_{u,s}(t)$ is centered at $t=u$. and

$$\sigma_t^2(u,s) = \int_{-\infty}^{\infty} (t-u)^2 \left| \psi_{u,s}(t) \right|^2 dt = s^2 \sigma_t^2$$

$$\text{where } \sigma_t^2 \equiv \int_{-\infty}^{\infty} t^2 \left| \psi(t) \right|^2 dt$$

Let f_0 be the center frequency of $\Psi(f)$. Then it follows that the center frequency of $\Psi_{u,s}(f)$ is f_0/s. The energy spread about f_0/s is

$$\sigma_f^2(u,s) = \int_0^\infty \left(f - \frac{f_0}{s}\right)^2 \left|\Psi_{u,s}(f)\right|^2 df$$

$$= \frac{\sigma_f^2}{s^2}$$

where $$\sigma_f^2 \equiv \int_0^\infty (f-f_0)^2 \left|\Psi(f)\right|^2 df.$$

Conclusions:

time position depends on u alone, spread depends on s.

frequency position depends of f_0 and s, spread depends on s.

$$\sigma_t^2(u,s) \propto s^2 \quad , \quad \sigma_f^2(u,s) \propto \frac{1}{s^2}$$

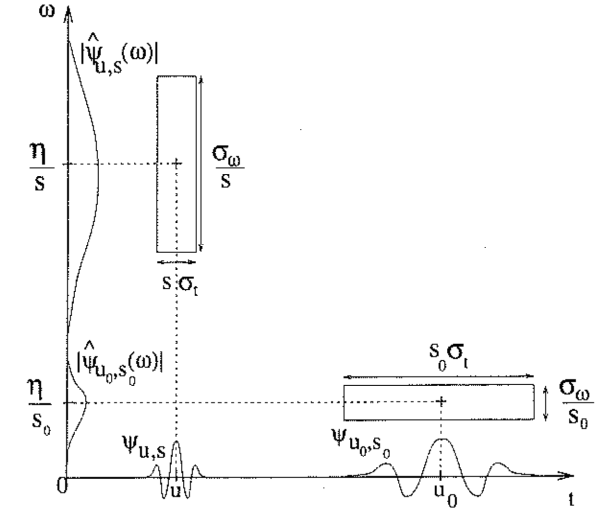
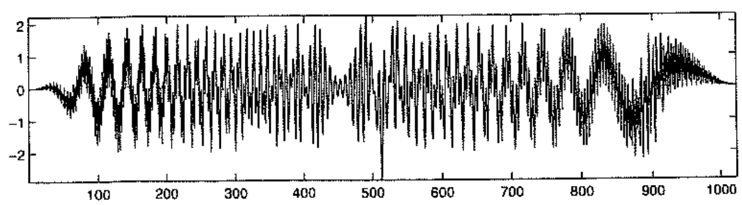
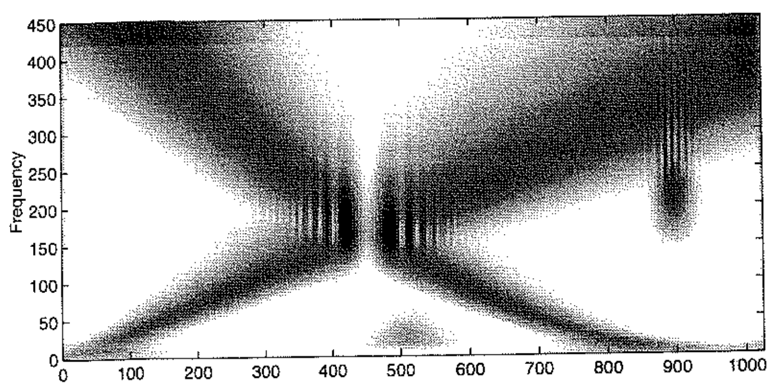


FIGURE 4.9 Heisenberg boxes of two wavelets. Smaller scales decrease the time spread but increase the frequency support, which is shifted towards higher frequencies.

Ex.



← signal



← scalogram

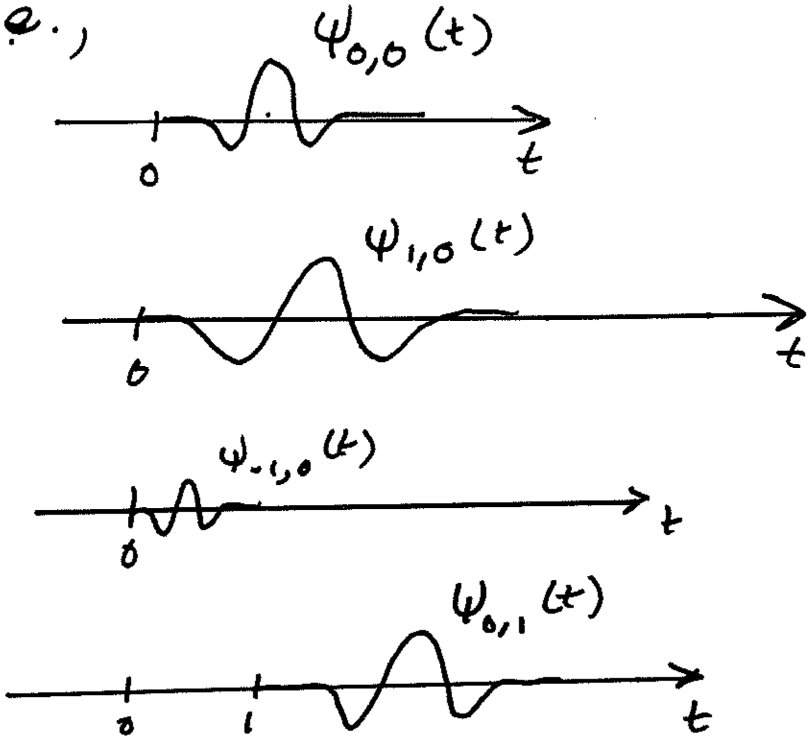
Wavelets and Wavelet Expansions

Let $f(t)$ be a finite energy signal. We write $f \in L_2(\mathbb{R})$

$$\Leftrightarrow \left\{ f : \int_{-\infty}^{\infty} f^2(t)\,dt < \infty \right\}$$

<u>Wavelets</u> (or wavelet basis functions) are localized waveforms  whose scaled and translated versions are all orthogonal to each other.

i.e.,



are orthogonal to each other

$$\int_{-\infty}^{\infty} \psi_{j,k}(t)\,\psi_{\ell,m}(t)\,dt = \delta_{j-\ell} \cdot \delta_{k-m}$$

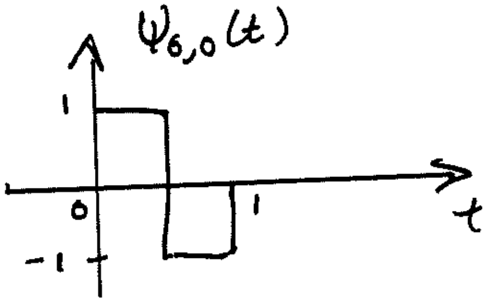
Consider expanding $f(t)$ in terms of wavelet basis functions

$$f(t) = \sum_k \sum_j a_{j,k}\, \psi_{j,k}(t)$$

What are $\{a_{j,k}\}$?

Simplest of all wavelets?

Haar wavelet

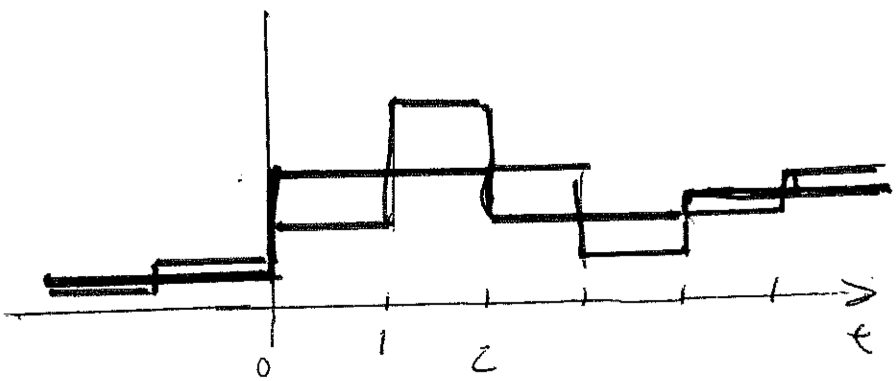


Connection to DSP

Suppose we have an f and a (Haar) approximation f_J

f_{J+1} is a even coarser approximation

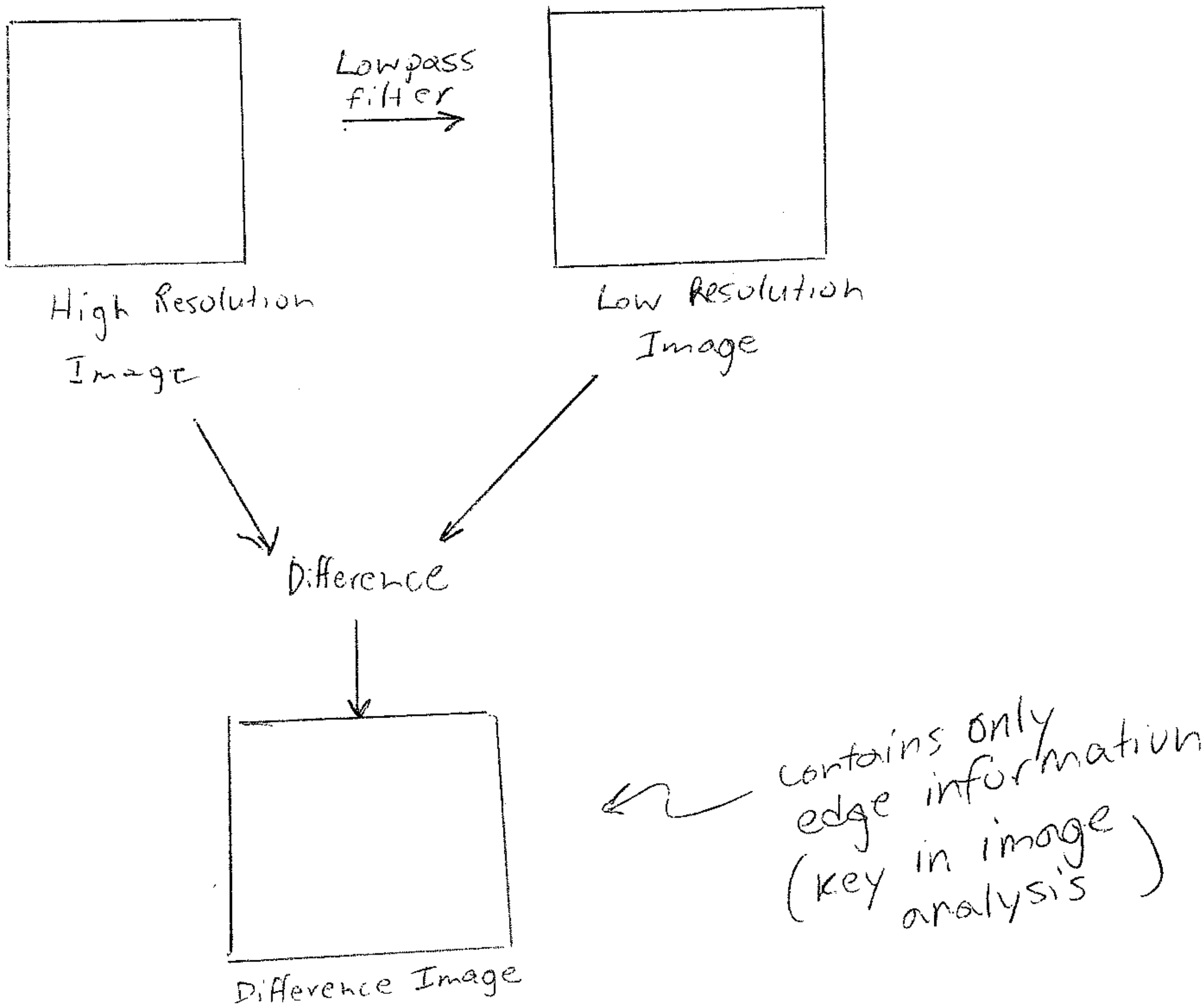
f_{J+1} can be easily expressed in terms of f_J



ex. $f_{J+1}(1) = \left(f_J\left(\frac{1}{2}\right) + f_J\left(\frac{3}{2}\right)\right)/2$

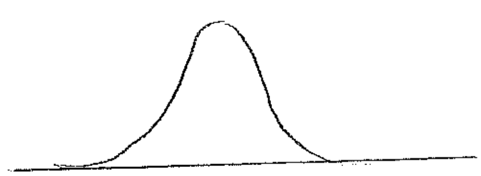
$\Rightarrow$ values of f_{J+1} approximation are just digital filtered $\left(\frac{1}{2}, \frac{1}{2}\right)$ versions of f_T values (wavelets = filter banks)

Multiscale Image Analysis



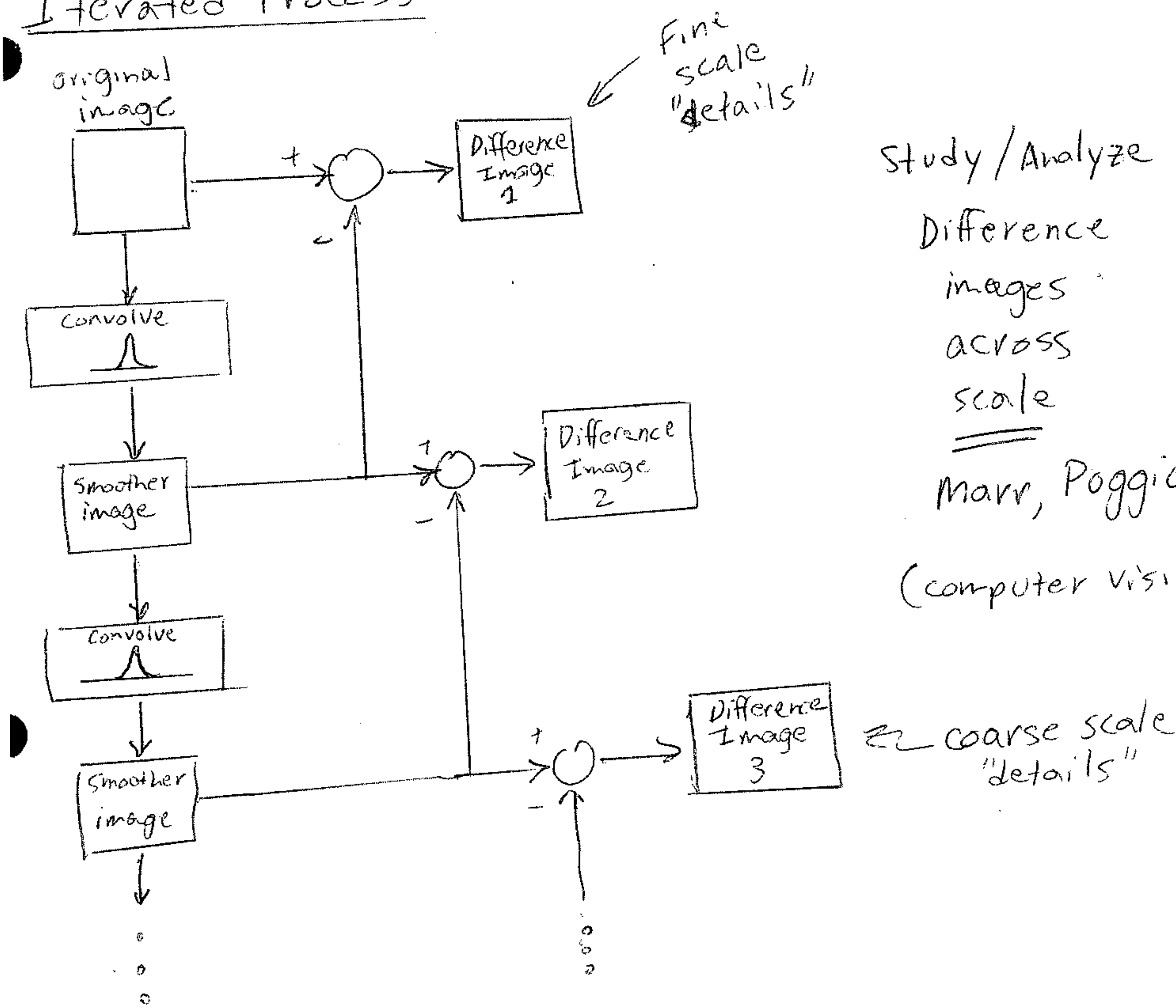
Lowpass filter

2-d Gaussian convolution kernel



(Multiscale Analysis)

Iterated Process



Study / Analyze Difference images across scale

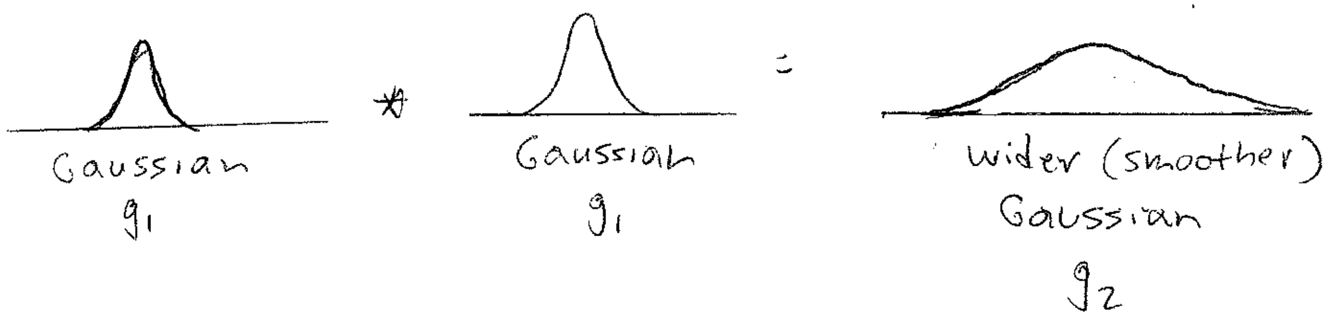
Marr, Poggio

(computer vision)

Key idea:

Burt & Adelson (1981) propose efficient image coding/compression scheme based on the fact that smoothed images can be subsampled (downsampled) without (significant) loss of information.

Equivalent Construction

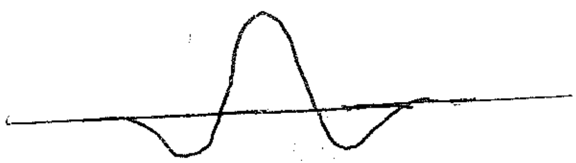


Difference Image

$$= g_1 * \text{Image} - g_2 * \text{Image}$$

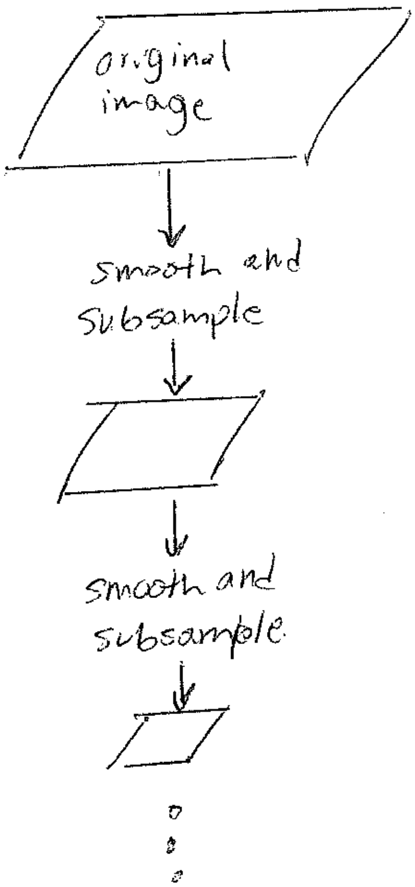
$$= (g_1 - g_2) * \text{Image}$$

$g_1 - g_2$ = difference of Gaussians
(also called Laplacian)



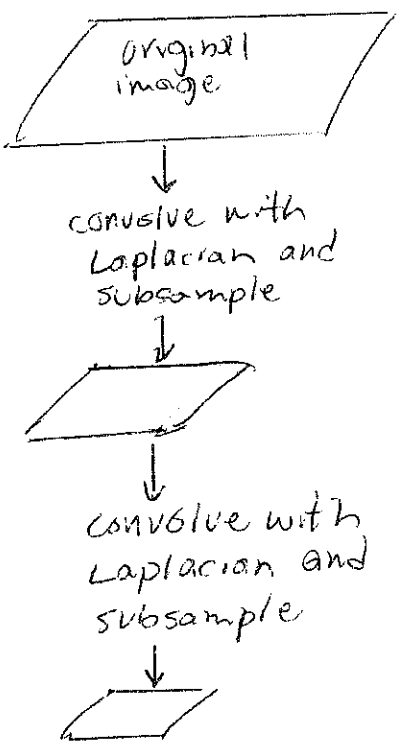
looks like
a wavelet!

Gaussian Pyramid (Burt & Adelson)



To compute difference images, interpolate low res/subsampled image and compute difference

... or equivalently Laplacian Pyramid



This is precisely the approach taken in wavelet analysis.

Burt & Adelson use Laplacian pyramid for image compression.

Applications

Approximation

$\{\phi_n\}_{n=0}^{\infty}$ orthonormal basis of $L^2(\mathbb{R})$
(e.g., wavelet basis)

Linear Approximation

$$f_M = \sum_{n=0}^{M-1} \langle f, \phi_n \rangle \phi_n \approx f$$

$$\text{error}(M) = \| f - f_M \|^2 = \sum_{n=M}^{\infty} |\langle f, \phi_n \rangle|^2$$

Non-linear Approximation

$$f_M = \sum_{n \in I_M} \langle f, \phi_n \rangle \phi_n$$

for some set I_M consisting of M basis vectors

$$\text{error}(M) = \| f - f_M \|^2 = \sum_{n \notin I_M} |\langle f, \phi_n \rangle|^2$$

Choose I_M to consist of indices of M basis functions with largest $|\langle f, \phi_n \rangle|$ values.

Adaptive Basis Selection

Suppose $\{\phi_n^1\}_n, \{\phi_n^2\}_n, \ldots \{\phi_n\}_n$

are all orthonormal bases

$$f_M^r = \sum_{n=0}^{M-1} \langle f, \phi_n^r \rangle \phi_n^j$$

$$\text{error}(M,j) = \| f - f_M^j \|^2 = \sum_{n \geq M} |\langle f, \phi_n^j \rangle|^2$$

Choose basis with smallest approximation error.

Estimation

Linear Estimation

We make measurements of a signal x in noise

$$\underbrace{Y_n}_{\text{observation}} = \underbrace{X_n}_{\text{signal}} + \underbrace{W_n}_{\text{noise}} \quad \text{Gaussian white}$$

$n = 0, \ldots, N-1$

Signal Estimation

$$y_n = x_n + w_n \qquad n = 0, \ldots, N-1$$

signal   Gaussian white noise

Vector Representation

$$\underline{Y} = \underline{X} + \underline{W} \; ; \quad \underline{Y} = \begin{pmatrix} y_0 \\ \vdots \\ y_{N-1} \end{pmatrix}$$

Filtering

1. - Compute transform of data $\underline{Y}$
2. - Process coefficients
3. - Transform back (inv. transform)

ex. Let U denote the DFT (matrix)

1. Compute $\underline{\theta} = U \underline{Y}$
2. Process coefficients $\underline{\theta} \xmapsto{\text{process}} \underline{\tilde{\theta}}$
3. $\underline{\hat{X}} = U^{-1} \underline{\tilde{\theta}}$

Frequency domain filter

Let's restrict our attention to coefficient-wise processing

(independently process each coefficient in $\underline{\theta}$)

We can express this as follows.

Let $\{\phi_n\}_{n=0}^{N-1}$ be an orthonormal basis for $\mathbb{R}^N$ (e.g., DFT basis, wavelet basis, etc.)

<u>Coefficient-wise Filtering</u>

$$\hat{\underline{x}} = \sum_{n=0}^{N-1} g_n\left(\langle \underline{y}, \underline{\phi_n} \rangle\right) \underline{\phi_n}$$

where g_n denotes an arbitrary processing operation ($g_n : \mathbb{R} \to \mathbb{R}$)

<u>Linear filtering</u>: $g_n(z) = a_n z$

simple weighting

<u>nonlinear filtering</u>: $g_n(z)$ is non-linear

Linear filtering is optimal
(in a Bayesian MSE sense)
if $\underline{x}$ is modeled as a Gaussian (zero-mean)
random vector.

Optimal weight

$$a_n^* = \frac{E\left[|\langle \underline{x}, \underline{\Phi}_n\rangle|^2\right]}{E\left[|\langle \underline{x}, \underline{\Phi}_n\rangle|^2\right] + E\left[|\langle \underline{w}, \underline{\Phi}_n\rangle|^2\right]}$$

Wiener Filter

Linear filtering is sub-optimal
(in a Bayesian MSE sense)
if $\underline{x}$ is modeled as
a non-Gaussian vector.
(think images)

Nonlinear Threshold Estimators

Since $\underline{W}$ is Gaussian white noise, each noise coefficient

$$\langle \underline{W}, \underline{\Phi}_n \rangle \sim N(0, \sigma^2)$$

Therefore, with high probability,

$$|\langle \underline{W}, \underline{\Phi}_n \rangle| < \tau \quad (\text{depending on } \sigma^2)$$

Hence; if

$$|\langle \underline{Y}, \underline{\Phi}_n \rangle| = |\langle \underline{X}, \underline{\Phi}_n \rangle + \langle \underline{W}, \underline{\Phi}_n \rangle| \geq \tau$$

then the signal component $\langle \underline{X}, \underline{\Phi}_n \rangle$ is large, otherwise it is "swamped" in the noise.

This suggest a <u>very</u> simple nonlinear estimator.

$$\underline{\hat{X}} = \sum_{n=0}^{N-1} \delta_T\left(\langle \underline{Y}, \underline{\Phi_n} \rangle \right) \underline{\Phi_n}$$

where

$$\delta_T(z) = \begin{cases} 0 , & \\ z , & \text{otherwise} \end{cases}$$

In many cases, this nonlinear estimator can significantly outperform linear estimators.

This is because many signals are not well modeled as a realization of a Gaussian process.

Wavelet-based threshold estimators are especially useful since the wavelet representation is often very sparse (i.e., a few large coefficients dominate the representation)

Compression

Basically, this is the approximation problem coupled with quantization/coding of coefficients $\{\langle f, \phi_n \rangle\}_{n=0}^{M-1}$.

To optimize bit allocation, coding etc., the signals to be compressed are often modeled as random processes

If we use a Gaussian process model, then the KL basis is optimal (useful in audio compression)

Images are non-Gaussian and the best known compression algorithms are based on non-linear approximations using wavelet bases.

Analysis of Signals and Processes

- Time-frequency analysis
- Edge / singularity detection / classification
- Analysis of fractional Brownian motion, fractal processes, and $1/f$ noise
- Wavelet transforms are approximately Karhunen-Loève type expansions for stochastically scale-invariant processes (including fBM and $1/f$ processes)
- multifractal Analysis
  (applications to study of turbulence, rainfall patterns images)

Modeling Random Processes

- Multiscale Markov Random Fields
- $1/f$ process modeling
- image modeling
- Bayesion signal and image estimation
- multiscale data segmentation
- detection / classification
- pattern recognition
- texture synthesis