Multi–resolution Analysis via Wavelets: An Introductory Tutorial

Evangelos Roussos

Program in Applied & Computational Mathematics
Princeton University
eroussos@math.princeton.edu

ISPRS Intl. Summer Sch. Advances in Medical Imaging — April 24–29, 2006

Why wavelets

- Wavelets have already had a remarkable impact.
- A lot of people are now applying wavelets to a lot of situations, and all seem to report favorable results.
- What is about wavelets that make them so popular?
- What is it that makes them so useful?

Wavelets?

- Wavelets are mathematical “Lego bricks” ...

Wavelets?

- Wavelets are mathematical “Lego bricks” ...
- ...carefully constructed so as to have special properties.

Wavelets?

- Wavelets are mathematical "Lego bricks" ...
- ... carefully constructed so as to have special properties.
- Wavelet analysis is a refinement of Fourier analysis ...

Wavelets?

- Wavelets are mathematical “Lego bricks” . . .
- . . . carefully constructed so as to have special properties.
- Wavelet analysis is a refinement of Fourier analysis . . .
- . . . incorporating the notions of
  - *Multiple scales* and
  - *Adaptive time–frequency localisation.*

Wavelets?

- Wavelets are mathematical "Lego bricks" . . .
- . . . carefully constructed so as to have special properties.
- Wavelet analysis is a refinement of Fourier analysis . . .
- . . . incorporating the notions of
  - *Multiple scales* and
  - *Adaptive time–frequency localisation*.
- An entire *family* of wavelets can be constructed from a single 'mother' wavelet.

What wavelets can do for you

- You can describe (almost) any function or signal in terms of wavelets . . .

What wavelets can do for you

- You can describe (almost) any function or signal in terms of wavelets ...
- ...and, actually, with very few of them ...

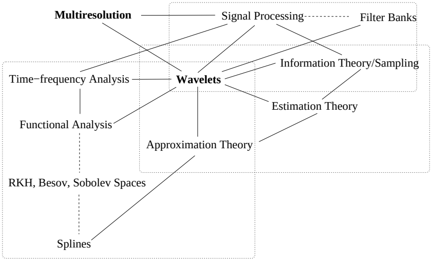
What wavelets can do for you

- You can describe (almost) any function or signal in terms of wavelets ...
- ... and, actually, with very few of them ...
- ... using an *extremely efficient* ($\mathcal{O}(n)$) algorithm.

What wavelets can do for you

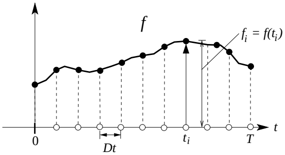
- You can describe (almost) any function or signal in terms of wavelets . . .
- . . . and, actually, with very few of them . . .
- . . . using an *extremely efficient* ($\mathcal{O}(n)$) algorithm.
- They are especially useful for describing *non–stationary* signals and signals with *singularities* (discontinuities, spikes, etc.).

Wavelets are multi–disciplinary



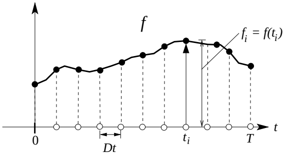
- Wavelets have an “interdisciplinary flavour”, integrating concepts from various scientific fields
- Functional analysis, signal processing, statistics, and others

Signals as functions and vectors



- Signals can be modelled as functions, $f : \Omega \to \mathcal{X}$, from the time– (or space–) domain to the space of amplitudes of the signal.
- Digital signals are from $\Omega = \mathbb{I} = \{(i_1, \ldots, i_d)\} \subseteq \mathbb{N}^d$, the space of d–tuples of integer indices, to $\mathcal{X} \subseteq \mathbb{R}$.
- Functions can be thought of as "vectors in a very high–dimensional space".
- Intuitively, we can understand this by *discretising* a function f with sampling interval Δt and letting $\Delta t \to 0$.

Physical Domain & Sampling



- Physical domain representation: expresses a function f on $\Omega \subseteq \mathbb{R}^d$ as a combination of an impulse-train of Dirac δ–functions at $\mathbf{r}_i = i\Delta\mathbf{r}$:

$$f(\mathbf{r}) = \sum_i f_i \delta(\mathbf{r} - \mathbf{r}_i) = \sum_i f(\mathbf{r}_i)\delta(\mathbf{r} - \mathbf{r}_i), \quad \mathbf{r} \in \Omega \subseteq \mathbb{R}^d.$$

- We can think of the δ–function as "picking" the value of a function, $f(\mathbf{r}_i) = f_i$, at each $\mathbf{r}_i$.
- The discrete equivalent is using the canonical basis, $\{\mathbf{e}_i\}_i$, $\mathbf{e}_i = (\ldots, 0, \ldots, 0, 1, 0, \ldots, 0, \ldots)$, with a 1 at position i.

Domains, representations, and transforms

- More generally, represent a signal as a sum of bases, $\{\mathbf{b}_k\}$:

$$f(\mathbf{r}) = \sum_k c_k b_k(\mathbf{r}), \quad \mathbf{r} \in \Omega \subseteq \mathbb{R}^d. \tag{1}$$

- *'Choosing a representation' means expressing our signal in a certain basis:* from the signal coefficients $\{f_i\}_i$ we get the coefficients $\{c_k\}_k$.
- Bases are "prototypical signals" and their amplitude is 'modulated' by their corresponding coefficient.

Frequency Domain (Fourier) Representation

- The basis functions are sinusoids, or complex exponentials, $\{e^{\iota\omega t}\}_\omega$, $\iota \stackrel{\text{def}}{=} \sqrt{-1}$:

$$f(t) = \int \mathrm{d}\omega\, \hat{f}(\omega) e^{\iota\omega t},$$

  where ω is the frequency
- The inverse representation is $f(t) \longmapsto \hat{f}(\omega)$, from t–space to ω–space:

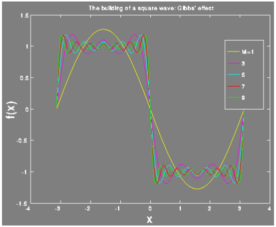
$$\hat{f}(\omega) \stackrel{\text{def}}{=} \int \mathrm{d}t\, e^{-\iota\omega t} f(t) = \langle e_\omega, f \rangle,$$

  where $\langle \cdot \rangle$ is the inner product.
- The Fourier domain is therefore useful for representing the *frequency content* of a signal.

Problems with the Fourier representation

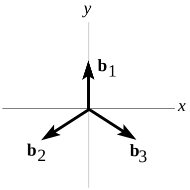
- Fourier bases are perfectly localised w.r.t. frequency, ω, but their support is the whole real axis, $(-\infty, \infty)$: *they are not localised in physical space*.
- This means that we cannot tell *when* a particular 'frequency event' happened.
- They are not very useful for time–varying signals.

The Gibbs effect



- Occurs in the representation of functions with *discontinuities* (jumps) with Fourier bases.
- An *infinite* number of functionsis needed to model the discontinuity.
- Using a finite ('truncated') series leads to fixed–size *oscillations* ("overshoots").
- *A fundamental issue with the Fourier transform: we are trying to model localised features with non–local bases!*.

From bases to frames



- We can represent a vector with more "bases" than the dimensionality of the vector space itself (under some conditions).
- This is called an *overcomplete* representation.
- It is very useful since it is a *robust* representation.
- The elements of a frame are generically called *'atoms'*.

Windowed Fourier Transform

- If we combine a Fourier basis $e^{\iota\omega t}$ with a *window* $g(t)$ that has finite support, we cut off the part of the signal outside the window.
- By shifting the window by u, in physical space, we get a family of atoms $\{g_{u,\omega}\}$:

$$g_{u,\omega}(t) \stackrel{\mathrm{def}}{=} e^{\iota\omega t} g(t-u).$$

- This leads to the *windowed–* or *short–time* Fourier Transform (STFT), $f(t) \longmapsto \tilde{f}(u,\omega)$:

$$\tilde{f}(u,\omega) = \langle g_{u,\omega}, f \rangle = \int \mathrm{d}t\, g^*(t-u) e^{-\iota\omega t} f(t).$$

- Gives information about signals in (t,ω) *simultaneously*.

Time–frequency Tiling

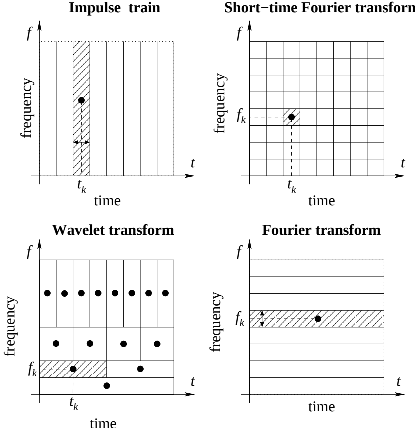
We say that the atoms *'tile'* the time–frequency space.

- We can visualise this by ploting the result of the transform in (u, ω)–space.
- The (effective) support of an atom localised at (u, ω) is

$$\sigma_t \times \sigma_\omega \doteq \left[u - \frac{\sigma_t}{2}, u + \frac{\sigma_t}{2}\right] \times \left[\omega - \frac{\sigma_\omega}{2}, \omega + \frac{\sigma_\omega}{2}\right]$$

- These are called 'Heisenberg boxes': *they represent the uncertainty, or trade–off, w.r.t. precise localisation in space versus frequency content.*

The tiling game



Tiles

- The Heisenberg boxes of δ functions are stripes with perfect localisation on the time axis, t, but infinite support on the frequancy axis.
- Fourier bases have the exactly oposite representation: perfect localisation on the frequency axis, ω, but zero "resolution" on the time axis.
- STFT–tiles are *identical* parallelograms, $\sigma_t \times \sigma_\omega$, shifted in time and space in order to cover the time–frequency plane.

Aren't we done after the STFT?

- The amount of localisation of STFT–atoms *remains fixed*.
- They introduce a *fixed scale* into the analysis: *width of the window*, σ.
- Signal features with time–scales $\Delta t < \sigma$ ($>$) *underlocalised* (*overlocalised*) in time.
- Must be obtained as a result of destructive (constructive) *interference* between the $g_{u,\omega}$'s.
- Many atoms must be used: $\tilde{f}(\omega, t)$ must be *spread out*.

Wavelets

- Wavelets are basis functions that *span the space of signals with finite energy*, therefore they can represent any function in this space.
- They are another kind of localised bases which *adapt* their time– and frequency–localisation.
- Wavelet analysis is a *scale–independent method*.
- Wavelets must satisfy certain requirements:
  - They integrate to zero: this property makes them 'wave–s',
  - They are *well localised* in space, i.e. they have 'compact support' ('–lets').

Wavelet construction: families

- Start with a window function $\psi(t)$: *mother wavelet*.
- Use ψ and *all possible scalings* of ψ:

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

where s: scale factor.

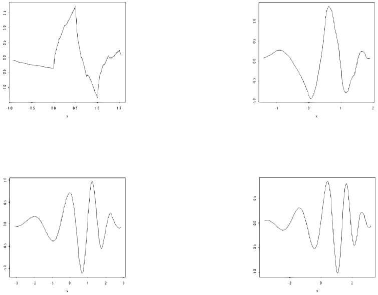
- Time localisation of signals: create translated versions of ψ:

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

General properties

- **Scale:** since wavelet bases are localised functions, and they can be formed by the dilation of a mother–wavelet, they naturally incorporate a notion of characteristic scale.
- **Compact support:** They are constructed such that they can be *identically zero* outside a certain range.
- **Smoothness:** wavelets trade-off locality of support and smoothness: the less localised they are, the smoother they become.

General properties, contd.



- **Multi–resolution:** they allow us to “zoom” on a certain level of detail in the signal, and add more detail as needed.
- **Variety:** There are many different types, depending on their other properties, like orthogonality, smoothness, locality of their support, their relation to equivalent digital filters, etc.

Continuous Wavelet Transform

- The generic wavelet transform of a function, $f(t) \longmapsto w(s,\tau)$, can then be written as

$$w(s,\tau) = \langle \psi_{s,\tau}, f \rangle = \int \mathrm{d}t\, \psi^*_{s,\tau}(t) f(t), \qquad (2)$$

- Again, this is an inner product, or correlation, of our signal with the wavelet function.
- *It is a measure of how much the details of our signal at that particular scale and position "look like" our basis.*
- By taking all translations τ and dilations s we get a very detailed picture of the *information content* of our signal, w.r.t. scale and spatial position.
- A visual representation of the transform in (s,τ)–space is called a *scalogram*.

Discrete Wavelet Transform

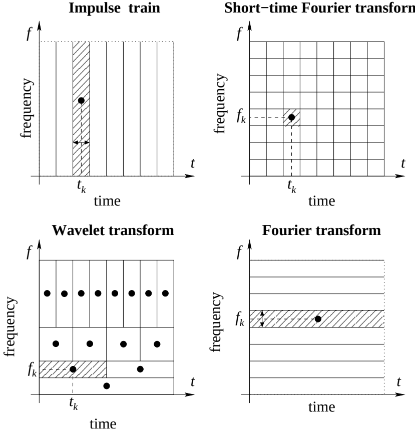
- The CWT is a *redundant* transform: to reconstruct the original signal from $\{w_{s,\tau}\}$ we do not need *all* dilations, s, and translations, τ.
- In many cases (e.g. signal compression) we do *not* want to have all wavelet coefficients.
- We can "sparsify" our set of "samples" $\{(s, \tau)\}$ to only a critical number of them, $\{(s_j, \tau_k)\}$.
- Translate and dilate at discrete steps:

$$\psi_{j,k}(t) = \frac{1}{\sqrt{{s_0}^j}} \psi\left(\frac{t - k\tau_0 {s_0}^j}{{s_0}^j}\right) \tag{3}$$

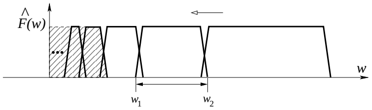
- The discrete wavelet transform is:

$$f(t) = \sum_{j,k} w_{j,k} \psi_{j,k}(t), \quad \forall j, k \in \mathbb{Z}, \quad t \in \mathbb{R}^d. \tag{4}$$

Discrete Wavelet Transform



The View from the Fourier Domain

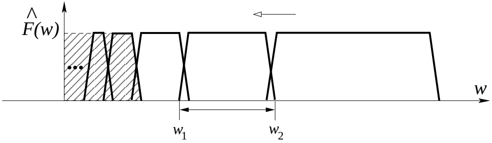


- Wavelets are signals too: they have a Fourier representation.
- Recall:
  - Contraction by a in physical domain causes a dilation by the same amount in frequency domain:

$$\mathcal{F}\left[f(at)\right] = 1/|a|\,\hat{f}\left[\omega/a\right], \quad a \in \mathbb{R},$$

  - Daughter–wavelets can be generated by scaling a mother–wavelet.
- The DWT amounts to adding a set of wavelet spectra in Fourier domain, in order to capture the *frequency content* of f.

Multiresolution Analysis (MRA)



- As we keep dillating the wavelets, $|\Delta\omega| \to 0$, $\Delta\omega = \omega_2 - \omega_1$,
- We need an infinite countable number of wavelet bases.
- One one could use another function instead, the *scaling function*, ϕ, with just the right frequency band.
- This can again represented in wavelets.

The discrete wavelet transform

- Combined with our wavelets, $\{\phi\} \cup \{\psi_j\}_j$ covers the whole spectrum of f.
- In physical domain,

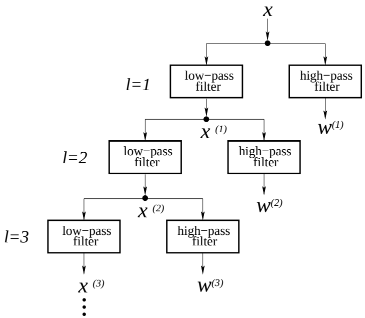
$$f(t) = \sum_k c_k \phi_k(t) + \sum_{j,k} d_{j,k} \psi_{j,k}(t)$$

- $\{c_k\}_k$ are the scaling coefficients and $\{d_{j,k}\}_{j,k}$ are the wavelet coefficients.

Implementation: The Fast Wavelet Transform

- A wavelet function has a band-like spectrum.
- Reconstructing a signal f can be interpreted as filtering with a set of *band–pass* filters.
- The action of the scaling function corresponds to *low–pass* filtering.

FDWT and Filter Banks



- The above analysis can be implemented *extremely efficiently*: computational cost $\mathcal{O}(n)$.
- The transform is implemented with a filter–bank, $(\mathbf{H}, \mathbf{G})$, via an iterated ('pyramid') algorithm.
- Iterative analysis of a signal into *several levels of detail.*

Wavelets for Images

- A digital image can be represented as a matrix $\mathbf{A} = [A_{ij}] \in \mathbb{R}^{N \times N}$ with 'grey level' A_{ij} at pixel (i, j).
- The application of the pyramid algorithm on an image can be done iteratively, in 'sub–bands'.
- Corresponds to the application of the wavelet transform with 2–D wavelets that are *tensor products* of 1–D ones.

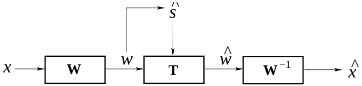
Wavelet Denoising

- Extract the desired signal from the noise.
- For the case of Gaussian white noise, the signal extraction problem can be stated as: *Given a set of noisy observations* $\{f_i^z\}_i$, $f_i^z = f^z(t_i)$, *sampled at times* $t_i = i/n$, *determine the 'true' values of the signal* f.
- The observation model is:

$$f_i^z = f_i + \sigma z_i, \qquad i = 1, \dots, n,$$

where σ^2 is the noise variance, and $z_i \sim \mathcal{N}(0, 1)$.

Wavelet Denoising II



- Re–write the observation model in the wavelet domain:

$$\mathbf{W}f_i^z = \mathbf{W}(f_i + \sigma z_i) = \mathbf{W}f_i + \sigma \mathbf{W}z_i.$$

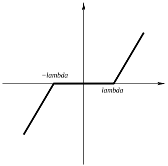
- If $\mathbf{W} = \{\psi_k\}_k$ is an orthonormal basis, the WT of Gaussian white noise, z_i, is Gaussian white noise, w_i, of the same amplitude. So,

$$\mathbf{W}f_i^z = \mathbf{W}f_i + \sigma w_i.$$

- Solving for f_i, gives

$$f_i = \mathbf{W}^{-1}(\mathbf{W}f_i^z - \sigma w_i).$$

Thresholding function



- *In general we do not know σw_i.* So, we need to estimate it: $\lambda = \widehat{\sigma w_i}$.
- Remove the estimated noise contribution from each of the wavelet coefficients $c_k = (\mathbf{Wf}^z)_k$. An appropriate way to do this is via the *soft–thresholding* function (see Fig. 6):

$$\eta_\lambda(x) = \begin{cases} x - \lambda, & x \geq \lambda \\ 0, & |x| < \lambda \\ x + \lambda, & x < -\lambda \end{cases}$$

Estimating the threshold λ

- Estimate the threshold, λ, using the 'universal' 'VisuShrink' method of Donoho and Johnstone:

$$\lambda = \sigma\sqrt{2\log(n)},$$

where n is the number of data samples.

- The value of the noise variance, σ^2, is not known. A robust estimate is

$$\hat{\sigma} = \frac{\mathrm{med}(\{|w_{J-1,k} - m|\}_k)}{0.6745},$$

where $m = \mathrm{med}(\{w_{J-1,k}\}_k)$, the median absolute value of the finest–scale $(J-1)$ wavelet coefficients.

Applying the method to 1–D functions

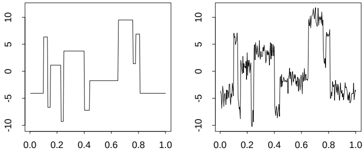


Figure: ‘Blocky’ function and noisy version with SNR = 5dB.

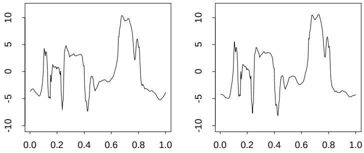


Figure: Wavelet denoising of the ‘Blocky’ function: left: VisuShrink, right: ‘analytic’ estimator.

Applying the method to 2–D functions

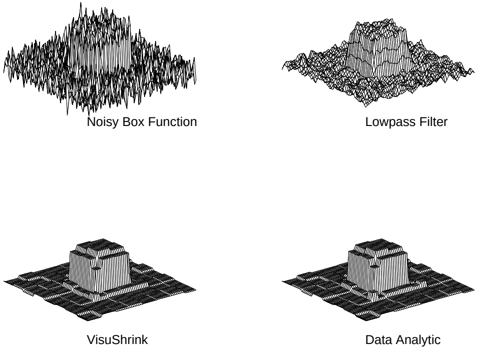


Figure: Two–dimenstional wavelet denoising