Wavelets and edge detection

WTBV WS 2016/17

February 7, 2018

1D edge detection

- Approximating derivatives: numerical instability
- Continuous wavelet transform (CWT)
- Wavelet functions and derivatives
- CWT, scaling and wavelet identities
- The algorithme-à-trous

2D edge detection

- Canny's method
- 2D separable CWT and gradients
- Gradient computation using scaling and wavelet identities

- Initial event:
  A. Grossmann and J. Morlet,
  Decompositions of Hardy functions into square integrable wavelets of constant shape, *SIAM J. Math. Analysis*, 1984
  (Analysis of seismic signals)
- ... but there were precursors . . . e.g.
  A. P. Caldéron,
  Intermediate Spaces and Interpolation, the Complex Method, *Studia Mathematica*, 1964
- see:
  S. Jaffard, Y. Meyer, R. Ryan,
  Wavelets, Tools for Science and Technology, SIAM 2001,
  in particular: Chap. 2: Wavelets from a Historical Perspective

- J. Canny,
  A computational approach to edge detection, *IEEE Trans Patt. Recog. and Mach. Intell.* 6, 961–1005, 1986.
- S. G. Mallat and S. Zhong,
  Characterization of signals from multiscale edges, *IEEE Trans Patt. Recog. and Mach. Intell.*, 14, 710–732, 1992.

- When dealing with a discretized version of a function $f(t)$

$$\ldots f(t_0), f(t_0+h), f(t_0+2h), \ldots$$

  (step size h for sampling) one may take the difference quotient as a numerical approximation of the derivative

$$f'(t_0) = \frac{d}{dt} f(t_0) \approx \frac{f(t_0+h) - f(t_0)}{h}$$

- From the numerical point of view this is a highly dangerous method (in particular if it is used iteratively) if the step size h is small
- As a rule: first apply a *smoothing* operator to the data before taking differences
- See the notebook *ramp-en.pdf* for illustration

- See handout *cwt-en.pdf* for the details and more information
- See notebook *CWT-15-en.pdf* for illustrations of the countinuous wavelet transform
- Relevant notebooks for illustration of edge detection
  - *ramp-en.pdf*
  - *atrous-poly-en.pdf*
  - *sobel-en.pdf*
  - *circletest.pdf*
  - *wvedges-en.pdf*

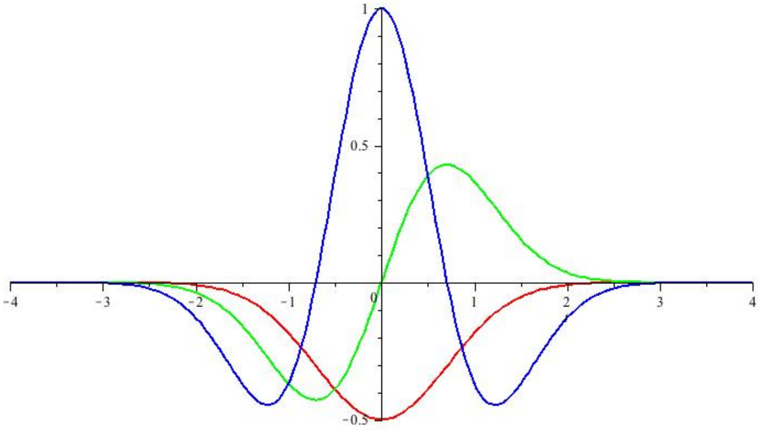


Figure: mexican-hat wavelet as second derivative of a Gaussian

- Generally: wavelet transforms consist in both
  - *smoothing* operations (approximation, low-pass filtering)
  - *differencing* operations (detail, high-pass filtering)
- $\psi(t)$ a "suitable" wavelet function, e.g.,
  (1) $\psi(t)$ is continuous and has vanishing zero-th moment

$$\widehat{\psi}(0) = \int_{\mathbb{R}} \psi(t)\, dt = 0$$

  (2) $\psi(t)$ decays rapidly as $t \to \pm\infty$

$$|\psi(t)| \leq A\, e^{-B|t|} \;\; (t \in \mathbb{R}) \;\; \text{for some constants } A, B > 0$$

  - normalization (for convenience) $\|\psi\|^2 = 1$
- Localization (mean and variance)

$$\mu = \int_{\mathbb{R}} t\, |\psi(t)|^2 dt \qquad \sigma^2 = \int_{\mathbb{R}} (t-\mu)^2\, |\psi(t)|^2 dt$$

- continuous scaling (dilation and translation) of $\psi(t)$

$$\psi_{s,a}(t) = \frac{1}{\sqrt{|s|}}\psi(\frac{t-a}{s}) \qquad (s, a \in \mathbb{R})$$

(this notation differs from the one used in the DWT context)

- Localization

$$\mu_{s,a} = \int t\,|\psi_{s,a}(t)|^2\,dt = ... = s\mu + a$$

$$\sigma^2_{s,a} = \int (t - \mu_{s,a})^2\,|\psi_{s,a}(t)|^2 dt = ... = s^2\sigma^2$$

- Fourier transform

$$\widehat{\psi_{s,a}}(\lambda) = \sqrt{s}\,e^{-2\pi i a\lambda}\,\widehat{\psi}(s\lambda)$$

- Definition of the continuous wavelet transform (CWT)

$$f(t) \longmapsto f^{\psi}(s,a) = \langle f \mid \psi_{s,a} \rangle$$
$$= \int_{\mathbb{R}} f(t)\, \overline{\psi_{s,a}(t)}\, dt$$

- Note

$$\|f - \psi_{s,a}\|^2 = \|f\|^2 + \|\psi\|^2 - 2\,\Re\left[f^{\psi}(s,a)\right]$$

- Intuitively: $f^{\psi}(s,a)$ correlates the behavior of $f(t)$ with that of $\psi(t)$ in the vicinity of $a \in \mathbb{R}$ (if $\mu = 0$) in resolution (scaling) $s \in \mathbb{R}$
- The CWT transform data $\left\{f^{\psi}(s,a)\right\}_{s,a\in\mathbb{R}}$ are highly redundant!

- Caldéron's reconstruction formula

  $f(t)$ can be reconstructed from its CWT transform data $\left\{f^{\psi}(s,a)\right\}_{s,a\in\mathbb{R}}$ under suitable conditions

$$f(t) = \frac{1}{C_\psi} \int_{s\in\mathbb{R}} \int_{a\in\mathbb{R}} f^{\psi}(s,a)\, \psi_{s,a}(t)\, da\, \frac{ds}{s^2}$$

- Here the number

$$C_\psi = \int_{\lambda\in\mathbb{R}} \frac{|\widehat{\psi}(\lambda)|^2}{|\lambda|}\, d\lambda$$

  must be finite and > 0

- This holds if conditions (1) and (2) for $\psi(t)$ are satisfied

- The HAAR wavelet function $\psi_{haar}(t)$ can be regarded as a derivative

$$\psi_{haar}(t) = \frac{d}{dt}\Delta(t) \quad \text{where} \quad \Delta(t) = \begin{cases} t & 0 \leq t \leq 1/2 \\ 1 - t & 1/2 \leq t \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

  $\Delta(t)$ is a smoothing function

- The mexican-hat wavelet function $\psi_{mex}(t)$ is a derivative

$$\psi_{mex}(t) = \frac{d}{dt}\left(t\, e^{-t^2}\right) = -\frac{1}{2}\frac{d^2}{dt^2}e^{-t^2}$$

  but $t\, e^{-t^2}$ is not really a smoothing function

- Take $\psi(t)$ as the derivative of a smoothing function $\theta(t)$

$$\psi(t) = \frac{d}{dt}\,\theta(t)$$

and define the scaled (dilated) and reversed version of $\theta(t)$ as

$$\overleftarrow{\theta_s}(t) = \frac{1}{\sqrt{s}}\,\theta(-\frac{t}{s})$$

- Then one has (simple exercise in differentiating under the integral)

$$f^{\psi}(s,a) = -s\,\frac{d}{da}(f \star \overleftarrow{\theta_s})(a)$$

- Note: the convolution $f \star \overleftarrow{\theta_s}$ is a $\overleftarrow{\theta_s}$-smoothed version of f
- Interpretation: Edges in the graph of $f(t)$ (high absolute values of the derivative) can be recognized by absolutely large values of the wavelet coefficients $f^{\psi}(s,a)$ over many scales (s values)

- Assume that for the wavelet function $\psi(t)$ one has a scaling function $\phi(t)$ (as in the MRA situation)
- Scaling and wavelet identities are

$$\phi(t) = \sqrt{2} \sum_{k \in \mathbb{Z}} h_k \, \phi(2t - k)$$

$$\psi(t) = \sqrt{2} \sum_{k \in \mathbb{Z}} g_k \, \phi(2t - k)$$

- Consider dyadic scaling for $\phi(t)$ and $\psi(t)$, i.e.,

$$\phi_{2^m,a}(t) = \frac{1}{2^{m/2}} \phi(\frac{t-a}{2^m}) \quad \psi_{2^m,a}(t) = \frac{1}{2^{m/2}} \psi(\frac{t-a}{2^m})$$

- Scaling and wavelet identities turn into

$$\phi_{2^{m+1},a}(t) = \sum_{k \in \mathbb{Z}} h_k \, \phi_{2^m,a+k\,2^m}(t)$$

$$\psi_{2^{m+1},a}(t) = \sum_{k \in \mathbb{Z}} g_k \, \phi_{2^m,a+k\,2^m}(t)$$

- Approximation and detail coefficients of a function $f(t)$, for dyadic scaling and integer translation
$(s, a) = (2^m, n) \quad (m, n \in \mathbb{Z})$

$$a_{m,n} = \langle f \mid \phi_{2^m,n} \rangle \qquad d_{m,n} = \langle f \mid \psi_{2^m,n} \rangle$$

- Recursion formulas for approximation and wavelet coefficients

$$a_{m+1,n} = \sum_{k \in \mathbb{Z}} h_k \, a_{m,n+k\,2^m} \quad (n \in \mathbb{Z})$$
$$d_{m+1,n} = \sum_{k \in \mathbb{Z}} g_k \, a_{m,n+k\,2^m} \quad (n \in \mathbb{Z})$$

- Written as filtering operations

$$\boldsymbol{a}^{(m+1)} = (a_{m+1,n})_{n \in \mathbb{Z}} = \overleftarrow{[(\uparrow_2)^m \, \boldsymbol{h}]} \star \boldsymbol{a}^{(m)}$$
$$\boldsymbol{d}^{(m+1)} = (d_{m+1,n})_{n \in \mathbb{Z}} = \overleftarrow{[(\uparrow_2)^m \, \boldsymbol{g}]} \star \boldsymbol{a}^{(m)}$$

- Here $(\uparrow_2)^m \, \boldsymbol{h}$ is the filter constructed from $\boldsymbol{h}$ by using m-fold upsampling with factor 2 ("spreading")

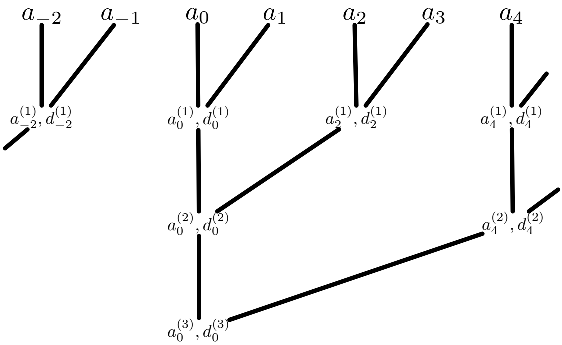


Figure: Scheme of the Haar transform

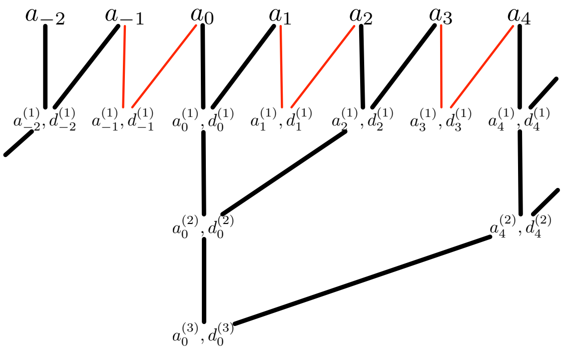


Figure: *à-trous* scheme (one level) for the Haar transform

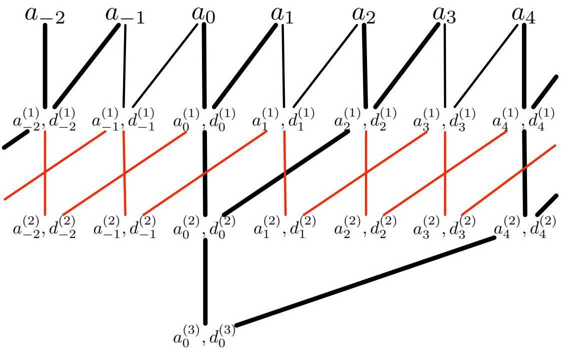


Figure: *à-trous* scheme (two levels) for the Haar transform

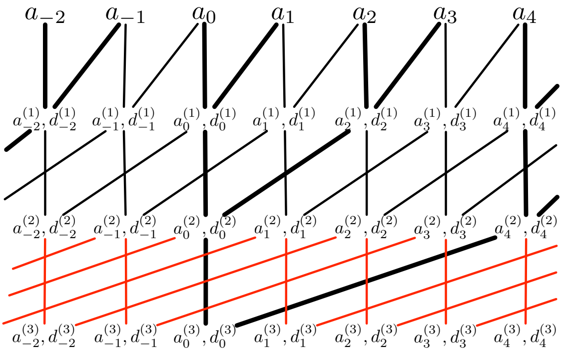


Figure: *à-trous* scheme (three levels) for the Haar transform

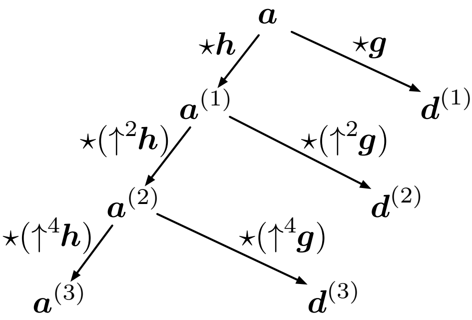


Figure: *à-trous* scheme (three levels)

high-pass filter: $\boldsymbol{g}$ low-pass filter: $\boldsymbol{h}$ signal: $\boldsymbol{a} = (a_n)_{n\in\mathbb{Z}}$

filtered signals: $\boldsymbol{a}^{(k)} = \left(a_n^{(k)}\right)_{n\in\mathbb{Z}}$ (approximation),

$\boldsymbol{d}^{(k)} = \left(d_n^{(k)}\right)_{n\in\mathbb{Z}}$ (detail)

Consider a function (an "image") $f(x, y) \in \mathcal{L}^2(\mathbb{R}^2)$.
$\partial_x f$, $\partial_y f$: partial derivatives of f,

$$\nabla f(x_0, y_0) = (\partial_x f(x_0, y_0), \partial_y f(x_0, y_0)) \qquad \text{gradient of } f$$

- CANNY's edge definition

$(x_0, y_0) \in \mathbb{R}^2$ is an *edge vertex* of $f(x, y)$ if the function

$$(x, y) \longmapsto |\nabla f(x, y)| = \sqrt{(\partial_x f(x, y))^2 + (\partial_y f(x, y))^2}$$

has a local maximum in (x_0, y_0) when running through this point in the direction $(\nabla f)(x_0, y_0)$ of steepest ascent/descent, formally:

$$|\nabla f\,[(x_0, y_0) + \varepsilon \cdot \nabla f(x_0, y_0)]| \leq |\nabla f(x_0, y_0)| \quad \text{for} \quad \varepsilon \approx 0$$

- Discretization and approximation of the gradient
  - Discretized function

$$A = \left[a_{p,q}\right]_{\substack{1 \le p \le m \\ 1 \le q \le n}} \qquad \text{where} \quad a_{p,q} = f(\xi_{p,q})$$

  - Approximation of the gradient

$$D^x = \left[d^x_{p,q}\right]_{\substack{1 \le p \le m \\ 1 \le q \le n}} \qquad \text{where} \quad d^x_{p,q} \approx (\partial_x f)(\xi_{p,q})$$
$$D^y = \left[d^y_{p,q}\right]_{\substack{1 \le p \le m \\ 1 \le q \le n}} \qquad \text{where} \quad d^y_{p,q} \approx (\partial_y f)(\xi_{p,q})$$

    computed using
    - $A \longmapsto D^x$: "derivation" in x-direction, smoothing in y-direction
    - $A \longmapsto D^y$: "derivation" in y-direction, smoothing in x-direction

- Example: The SOBEL operators

$$S_h : A \longmapsto D^x = A \star \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$S_v : A \longmapsto D^y = A \star \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

- Written as Kronecker products:

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$$

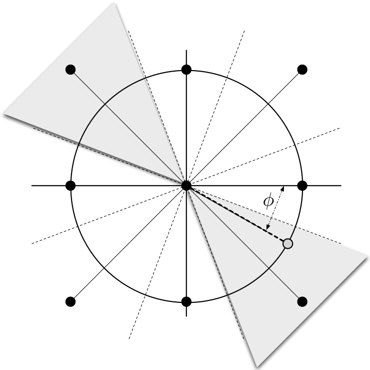
$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$$

- From rectangular coordinates to polar coordinates

$$\langle x, y \rangle \longmapsto \langle r, \phi \rangle = [\,\sqrt{x^2 + y^2}, \arctan(y/x)\,]$$

  where $-\pi < \arctan(y/x) \leq \pi$

- Discretizing the directions



- The 8 neighbors $\langle \tilde{p}, \tilde{q} \rangle$ of a vertex $\langle p, q \rangle$ with integer coordinates are

$$\langle \tilde{p}, \tilde{q} \rangle = \left\{ \begin{array}{l} \langle p \pm 1, q \pm 1 \rangle \\ \langle p \pm 1, 0 \rangle \\ \langle 0, q \pm 1 \rangle \end{array} \right\} = \langle p, q \rangle + \delta$$

$$\text{where} \quad \delta \in \Delta = \{ \langle \pm 1, \pm 1 \rangle, \langle \pm 1, 0 \rangle, \langle 0, \pm 1 \rangle \}$$

- They define 8 conic sectors rooted in $\langle p, q \rangle$ and of 45^{o} angular width,
symmetric w.r.t. to the respective straight lines joining $\langle p, q \rangle \longleftrightarrow \langle \tilde{p}, \tilde{q} \rangle$
- For an integer vertex $\langle p, q \rangle$ any direction $[1, \phi]$ defines a unique sector containing the straight line with direction $[1, \phi]$ rooted in $\langle p, q \rangle$, hence a neighbor $\langle \tilde{p}, \tilde{q} \rangle$ and the vector

$$\delta_{p,q}(\phi) = \langle \tilde{p}, \tilde{q} \rangle - \langle p, q \rangle = \langle \tilde{p} - p, \tilde{q} - q \rangle \in \Delta$$

- From the gradient matrices

$$D^x = \left[d^x_{p,q}\right] \qquad D^y = \left[d^y_{p,q}\right]$$

one obtains

  - the matrix of absolute length of the gradient vectors

$$R = \left[r_{p,q}\right] = \left[\sqrt{(d^x_{p,q})^2 + (d^y_{p,q})^2}\right]$$

  - the matrix of discretized gradient directions

$$S = \left[\delta_{p,q}(\phi)\right] \quad \text{where} \quad \phi = \arctan(d^y_{p,q}/d^x_{p,q})$$

- Given f, A, D^x, D^y, R, S one defines
  - $\langle p, q \rangle$ is an *edge-candidate* if

$$r_{p,q} = \max\{\, r_{p,q}, r_{\langle p,q\rangle \pm \delta_{p,q}(\phi)} \,\}$$

  - $\langle p, q \rangle$ is a *level-λ edge vertex* (for $\lambda \in [0,1]$) if

$$r_{p,q} = \max\{\, r_{p,q}, r_{\langle p,q\rangle \pm \delta_{p,q}(\phi)} \,\} \;\text{ and }\; r_{p,q} \geq \lambda \cdot \max_{p',q'} r_{p',q'}$$

    (or $r_{p,q} \geq \lambda \cdot \text{aver}_{p',q'}\; r_{p',q'}$)
- Two-level method with $0 < \lambda_{low} < \lambda_{high} \leq 1$:
  - $\langle p, q \rangle$ is a *strong edge vertex* if it is a level-λ_{high} edge vertex
  - $\langle p, q \rangle$ is a *weak edge vertex* if it is a level-λ_{low} edge vertex, but not a strong one
  - Weak edge vertices are iteratively turned into strong edge vertices
    if they are neighbors of strong edge vertices

- For a 1D wavelet function $\psi(x)$ let

$$\Psi(x, y) = \psi(x)\,\psi(y)$$

  be the 2D separable wavelet function constructed from it
- The 2D CWT of a function $f(x, y)$ is defined as

$$f^{\Psi}(a, b, s) = \frac{1}{s} \iint_{x,y \in \mathbb{R}\times\mathbb{R}} f(x, y)\,\Psi\left(\frac{x-a}{s}, \frac{y-b}{s}\right) dx\,dy$$

- As in the 1D case, let $\psi(x) = \frac{d}{dx}\,\theta(x)$ be the derivative of a "smoothing function" $\theta(x)$
- The 2D separable smoothing function constructef from $\theta(x)$ is

$$\Theta(x,y) = \theta(x)\,\theta(y)$$

- The 2D partial wavelet functions are

$$\Psi^x(x,y) = \psi(x)\,\theta(y) = \partial_x\,\Theta(x,y)$$

$$\Psi^y(x,y) = \theta(x)\,\psi(y) = \partial_y\,\Theta(x,y)$$

- The 2D partial continuous wavelet transform (CWT) is defined as

$$f^{\Psi^x}(a,b,s) = \frac{1}{s}\iint f(x,y)\,\Psi^x(\frac{x-a}{s},\frac{y-b}{s})\,dx\,dy$$

$$f^{\Psi^y}(a,b,s) = \frac{1}{s}\iint f(x,y)\,\Psi^y(\frac{x-a}{s},\frac{y-b}{s})\,dx\,dy$$

- Defining a smoothed version of f using an s-scaling of Θ

$$f^{\Theta}(a,b,s) = \iint_{x,y\in\mathbb{R}\times\mathbb{R}} f(x,y)\,\Theta(\frac{x-a}{s},\frac{y-b}{s})\,dx\,dy$$

one has

$$\begin{bmatrix} -f^{\Psi^x}(a,b,s) \\ -f^{\Psi^y}(a,b,s)) \end{bmatrix} = \begin{bmatrix} \partial_a f^{\Theta}(a,b,s) \\ \partial_b f^{\Theta}(a,b,s) \end{bmatrix} = \nabla f^{\Theta}(a,b,s)$$

- This shows how to to compute gradient values of f^{Θ} using the 2D CWT

- Assume that 1D scaling, wavelet functions are described by

$$\phi(x) = \sqrt{2}\sum_k h_k\,\phi(2x-k) \qquad \psi(x) = \sqrt{2}\sum_k g_k\,\phi(2x-k)$$

- and that the smoothing function $\theta(x)$ also satisfies a scaling identity

$$\theta(x/2) = \sqrt{2}\sum_\ell r_\ell\,\theta(x-\ell/2)$$

- A scaling function $\Phi^x(x,y)$ for the wavelet function $\Psi^x(x,y) = \psi(x)\,\theta(y)$ can be defined as

$$\Phi^x(x,y) = \phi(x)\,\theta(y/2)$$

  which then satisfies a 2D scaling identity

$$\Phi^x(x,y) = 2\sum_{k,\ell} h_k\, r_\ell\, \Phi^x(2x-k, 2y-\ell)$$

- The 2D wavelet identity for $\Psi^x(x,y)$ is simply

$$\Psi^x(x,y) = \sqrt{2}\sum_k g_k\,\Phi^x(2x-k,2y) = 2\sum_{k,\ell} g_k\,\epsilon_\ell\,\Phi^x(2x-k,2y-\ell)$$

  with $\epsilon_\ell = \frac{1}{\sqrt{2}}\,\delta_{\ell,0}$. Similarly for $\Phi^y(x,y)$ and $\Psi^y(x,y)$

- Example: The HAAR wavelet function $\psi_{haar}(t)$ is the derivative of the smoothing function $\theta(t) = \Delta(t)$:

$$\psi_{haar}(t) = \frac{d}{dt}\Delta(t) \quad \text{where} \quad \Delta(t) = \begin{cases} t & 0 \leq t \leq 1/2 \\ 1 - t & 1/2 \leq t \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

- The function $\Delta(t)$ satisfies

$$\Delta(x) + 2\,\Delta(x - 1/2) + \Delta(x - 1) = 2\Delta(x/2)$$

- which can be written as a scaling equation

$$\Delta(x) = \frac{1}{2}\left(\Delta(2x) + 2\,\Delta(2x - 1) + \Delta(2x - 2)\right)$$

- so that

$$\boldsymbol{r} = \frac{1}{2\sqrt{2}}\,\langle 1, 2, 1\rangle$$

  is a B-spline filter

- Approximation and detail coefficients are defined as usual

$$a^x_{m;k,\ell} = \langle f \mid \Phi^x_{2^m,k,\ell} \rangle = \iint f(x,y) \frac{1}{2^m} \Phi^x(\frac{x-k}{2^m}, \frac{y-\ell}{2^m})\, dx\, dy$$
$$d^x_{m;k,\ell} = \langle f \mid \Psi^x_{2^m,k,\ell} \rangle = \iint f(x,y) \frac{1}{2^m} \Psi^x(\frac{x-k}{2^m}, \frac{y-\ell}{2^m})\, dx\, dy$$

  and analogously for $a^y_{m;k,\ell}$ and $d^y_{m;k,\ell}$
- Recursion formulas for the approximation coefficients

$$a^x_{m+1;p,q} = \sum_{k,\ell} h_k\, r_\ell\, a^x_{m;p+k2^m,q+\ell 2^m}$$
$$a^y_{m+1;p,q} = \sum_{k,\ell} r_k\, h_\ell\, a^y_{m;p+k2^m,q+\ell 2^m}$$

- Formulas for the detail coefficients

$$d^x_{m+1;p,q} = \sum_{k,\ell} g_k\, \epsilon_\ell\, a^x_{m;p+k2^m,q+\ell 2^m} = \tfrac{1}{\sqrt{2}} \sum_k g_k\, a^x_{m;p+k2^m,q}$$
$$d^y_{m+1;p,q} = \sum_{k,\ell} \epsilon_k\, g_\ell\, a^y_{m;p+k2^m,q+\ell 2^m} = \tfrac{1}{\sqrt{2}} \sum_k g_\ell\, a^y_{m;p,q+\ell 2^m}$$

- Computational scheme (*à trous* algorithm)

$$A_m^x = \left[f^{\Phi^x}(2^m; p, q)\right]_{p,q} \qquad A_m^y = \left[f^{\Phi^y}(2^m; p, q)\right]_{p,q}$$
$$D_m^x = \left[f^{\Psi^x}(2^m; p, q)\right]_{p,q} \qquad D_m^y = \left[f^{\Psi^y}(2^m; p, q)\right]_{p,q}$$

where $A_0 = A_0^x = A_0^y = [f(p,q)]_{p,q}$