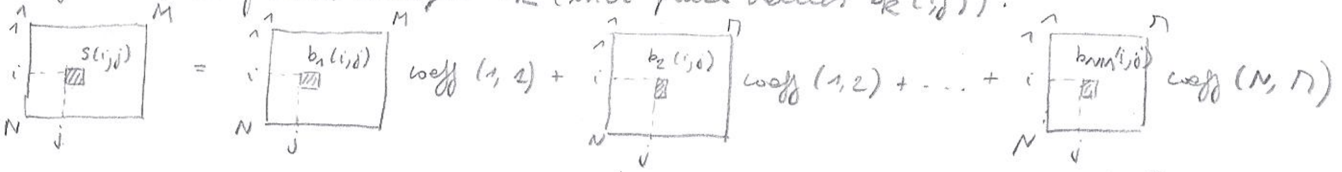
CHAPITRE 1 : GLOBAL TRANSFORMS

GENERAL MODEL

The basic concept of any transform is to represent an image s (with pixel values $s(i,j)$) as weighted sum of basis images b_k (with pixel values $b_k(i,j)$).



The matrix equation is : $s(i,j) = \sum_{k=1}^{L} \text{coeff}_k \, b_k(i,j)$ or $s(i,j) = \sum_{k=1}^{N}\sum_{l=1}^{M} \text{coeff}(k,l)\, b_{kl}(i,j)$

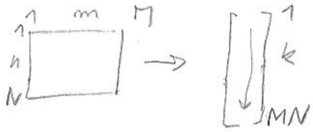
$\underline{s} = \underline{b} \cdot \text{coeff}$ (*)

where : L is the number of basis images to represent the full image s

$N \times M$ is the size of the original image and thus the number of basis vectors needed to perfectly represent it ($\mathbb{R}^{N\times M} = \text{span}\{v_i, i = 1 \dots N\times M \mid v = \sum_i \alpha_i v_i\}$)

forward transform { s corresponds to the spatial domain
coeff : corresponds to the transformed domain } inverse transform

The recipe for matrix to vector conversion is : (2D → 1D) $\begin{cases} s(n,m) \Rightarrow s(k) \\ k = m + (n-1)M \end{cases}$



vector to matrix conversion is (1D → 2D) $k \Rightarrow \begin{cases} m = \text{mod}_M(k) \\ n = 1 + \left\lfloor \frac{k}{M} \right\rfloor \end{cases}$

The physical meaning of the expansion into basis is the projection of s on a plane defined by b_i ($i = 1, \dots L$) linearly independent vectors to form $s' = \sum b_i \,\text{coeff}_i$

If b_i are orthogonal : $\text{coeff}_i = b_i^T \cdot s'$. To recover s from its projections on b_i, we need a number of linear independent vector b_i equal to the number of component of the vectors. Given : $s' = \underline{b}\,\text{coeff}$ and $s = \underline{b} \cdot \text{coeff}$ (from here above *)

with s': [P, 1] ; b_k : [P, 1], k = 1 ... L ; $\underline{b}$: [P, L] ; coeff [L, 1] , if

- L = P (L = N.M), it is always possible to define a set of linearly independent basis images so that no signal information is lost ($s = s'$)
- L < P (L < NM), signal information is lost ($s \neq s'$).

The goal of every transform is to minimize this error, generally through the minimization of the least square error : $LSE = \|s - s'\|^2 = (s - s')^T(s - s')$ and $\frac{\delta(LSE)}{\delta \text{coeff}} = 0$ for minimization, where LSE and coeff are vector following the

properties : $\frac{\delta(\underline{A}\,\text{coeff})}{\delta\,\text{coeff}} = \underline{A}^T$ and $\frac{\delta(\text{coeff}^T \cdot \underline{A})}{\delta\,\text{coeff}}$ ...

Démo :

$LSE = (s - s')^T(s - s') = s^T s - s^T \underline{b}\,\text{coeff} - \text{coeff}^T \underline{b}^T s + \dots$

$\frac{\delta(LSE)}{\delta\text{coeff}} = \begin{bmatrix} \frac{\delta(LSE)}{\delta \text{coeff}_1} \\ \vdots \\ \frac{\delta(LSE)}{\delta\text{coeff}_L} \end{bmatrix} = 0 - (s^T\underline{b})^T - (\underline{b}^T s) + (\underline{b}^T \underline{b}\,\text{coeff} \dots$

$= -\underline{b}^T s - \underline{b}^T s + \underline{b}^T \underline{b}\,\text{coeff} + \underline{b}^T \underline{b} \dots$

$\Leftrightarrow \underline{b}^T \underline{b}\,\text{coeff} = \underline{b}^T \underline{s} \Rightarrow \text{coeff} = \underbrace{(\underline{b}^T\underline{b})^{-1}\underline{b}^T}_{=:\ \text{pseudo inverse of} \dots} s = \underline{b}^T \dots$

Proof :
$A \in \mathbb{R}^{m\times n}$, $x \in \mathbb{R}^n$
↳ coeff is a vector in this case

$A.x$ → product by rows of A : $\tilde{\alpha}_i^T$ (convention) · column vector x
if convention changes, result will be the transpose of the result found here.

$\frac{\partial A x}{\partial x} = \begin{bmatrix} \frac{\partial \tilde{\alpha}_1^T x}{\partial x} \\ \vdots \\ \frac{\partial \tilde{\alpha}_m^T x}{\partial x} \end{bmatrix} = [\tilde{\alpha}_1 \; \tilde{\alpha}_2 \cdots \tilde{\alpha}_m] = A^T$

Lire slide 18 à 25 si intérêt mais l'étude pour l'examen commence à la Karhunen - Loève transform

DISCRETE KARHUNEN LOEVE TRANSFORM

Let b_k be the eigenvectors and λ_k the corresponding eigenvalues of $\underline{C}_s$ the covariance matrix $\left(= \frac{1}{K-1} \sum_{k=1}^{K} (s_k - m_s)(s_k - m_s)^T \right.$ where $m_s = \frac{1}{K} \sum_{k=1}^{K} s_k$ is the mean$\left. \right)$, and assume that the eigenvalues have been ordered in decreasing order: $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_p$.

The Karhunen Loève transformation matrix is given by $\underline{b} = (b_1, b_2, \dots b_p)$

The Discrete Karhunen Loève Transform is defined as: $coeff = \underline{b}^T (s - m_s)$

and calculate the L basis images so that for each $L \leq P$, the $MSE := E(\|s - s'\|^2)$ between S and S' is minimal (where E is the expected value operator)

The properties of the DKLT are:

1) $m_{coeff} = 0$

2) $\underline{C}_{coeff} = diag(\lambda_1, \lambda_2, \dots \lambda_p)$

3) the elements of coeff are uncorrelated

4) λ_i is equal to the variance of $coeff_i$ along vector b_i

5) since $\underline{C}_s$ is a real symmetric matrix, it is always possible to find a set of orthonormal eigenvectors. It therefore follows that s can be reconstructed as follows: $s = \underline{b}\ coeff + m_s$ (inverse KLT)

6) if $L < P$, $s \cong \underline{b}_L\ coeff + m_s$ ($\underline{b}_L = (b_1, \dots b_L, 0 \dots 0)$), the MSE is given by

$$MSE = \sum_{j=1}^{P} \lambda_j - \sum_{j=1}^{L} \lambda_j = \sum_{j=L+1}^{P} \lambda_j$$

$\hookrightarrow$ or $coeff = (coeff_1, coeff_2, \dots, 0, \dots 0)$ thus we either zero out the basis or the coefficients.

Proofs of properties

Assumptions:

- Let s be a square image: $[N^2, 1]$ and s' be: $[L, 1]$ with $\begin{cases} L \leq N^2 \\ E(s) = 0 \\ E(s') = 0 \end{cases}$
- we impose that $E(\|s - s'\|^2)$ is minimal $\forall L$
- the basis $\underline{b}$ is orthonormal: $\underline{b}^T \underline{b} = I$

(1) $m_{coeff} = E(coeff) = E(\underbrace{b^T (s - m_s)}_{\text{deterministic, out of expectation operator}}) = b^T (\underbrace{E(s)}_{=: m_s} - m_s) = 0$

(3) As $\underline{C}s$ is diagonal, the diagonals terms are zero thus the crosscorrelation is zero, hence they are uncorrelated.

(6) In the case where $L = N^2$,

$$C_s = E(ss^T) = \begin{bmatrix} \sigma_s^2(1) & \dots \text{other terms} \\ \vdots & \\ & \sigma_s^2(N^2) \end{bmatrix} \quad C_{coeff} = E(coeff\ coeff^T) = \begin{pmatrix} \sigma_{coeff}(1) & \dots \text{other terms} \\ \vdots & \\ & \sigma^2_{coeff}(N^2) \end{pmatrix}$$

$$= E(b^T s s^T b) = b^T E(ss^T) b = b^T C_s b$$

thus $tr(C_s) = tr(C_{coeff}) = \sum_{i=1}^{N^2} \sigma^2_{coeff}(i)$ is basis invariant

In the case where $L \leq N^2$ and in the case we zero out some coefficients.

$$C_{coeff}' = \left[\begin{array}{cc|c} \sigma^2_{coeff}(1) & \dots \text{other terms} \leq 0 & 0 \\ \vdots & \sigma^2_{coeff}(L) & \vdots \\ 0 & 0 & 0 \end{array} \right]$$

thus $tr(C_{coeff}') = tr(C_s) - (\sigma^2_{coeff}(L+1) + \dots + \sigma^2_{coeff}(N^2))$

the MSE is:

$$E(\|s-s'\|^2) = E((s-s')^T(s-s')) = \underbrace{E(s^Ts)}_{1} + \underbrace{E(s'^Ts')}_{2} - \underbrace{E(s^Ts')}_{3} - \underbrace{E(s'^Ts)}_{4}$$

given at exam

1. $E(s^Ts) = E(s^2(1) + \dots + s^2(N^2)) = \sigma_s^2(1) + \dots + \sigma_s^2(N^2) = tr(C_s)$ — represents the energy of s

$E(s^Ts) = E((b.coeff)^T(b\,coeff)) = E(coeff^T \underbrace{b^Tb}_{I} coeff) = E(coeff^T coeff) = tr(C_{coeff})$ — only valid for orthonormal b because $b^Tb = I$

2. $E(s'^Ts') = E((b\,coeff')^T(b\,coeff')) = tr(C_{coeff}')$

3. $E(s^Ts') = E((b\,coeff)^T(b\,coeff')) = E(coeff^T b^T b\,coeff') = E(coeff^T coeff')$

$= E(coeff^2(1) + \dots + coeff^2(L)) = \sigma^2_{coeff}(1) + \dots + \sigma^2_{coeff}(L) = tr(C_{coeff}')$

4. $E(s'^Ts) = E((b\,coeff')^T(b\,coeff)) = tr(C_{coeff}')$

* the results $tr(C_s) = tr(C_{coeff})$ means that the energy in time domain C_s is conserved in the transform domain C_{coeff}. this follows the Parseval theorem.

$$= tr(C_{coeff}) + tr(C_{coeff}') - tr(C_{coeff}') - tr(C_{coeff}')$$

$$= tr(C_{coeff}) - tr(C_{coeff}')$$

by dims of property ⑥

$$= \sigma^2_{coeff}(L+1) + \dots \sigma^2_{coeff}(N^2) \text{ and this should be minimal}$$

Because the energy is fixed, it means that $\sigma^2(1) + \dots + \sigma^2(L)$ should have a maximal energy. The minimization of the MSE also impose thus that the variance of the projection of s on $\underline{b}$ should be maximum. As an example, if $L = 1$:

$$\begin{cases} \sigma^2 coeff(1) = C_{coeff}(1,1) = b_1^T C_s b_1 \text{ to be max} \xrightarrow{\text{lagrange}} \frac{d}{db_1}(b_1^T C_s b_1 - \lambda(b_1^T b_1 - 1)) = 0 \\ b_1^T b_1 = 1 \text{ by constraint of orthonormal basis} \end{cases}$$

By $\frac{\partial(Ax)}{\partial x} = A^T$: $\frac{d}{db_1}(b_1^T(C_s - \lambda I)b_1) = (b_1^T(C_s - \lambda I))^T$

By $\frac{\partial(x^TA)}{\partial x} = A$: $\frac{d}{db_1}(b_1^T(C_s - \lambda I)b_1) = (C_s - \lambda I)b_1$

$$(b_1^T(C_s - \lambda I))^T + (C_s - \lambda I)b_1 = (C_s - \lambda I)^T b_1 + (C_s - \lambda I)b_1$$

$$= 2(C_s b_1 - \lambda b_1) = 0 \text{ as } C_s = C_s^T \text{ because } C_s \text{ is symmetric}$$

$\Rightarrow C_s b_1 = \lambda b_1 \rightarrow b_1$ is an eigenvector of C_s and as $\underbrace{b_1^T C_s b_1}_{=\lambda}$ must be max, b_1 must be the eigenvector with the largest possible eigenvalue λ_1. By generalization, λ_2 and the following should be the 2nd -- highest eigenvalues such that

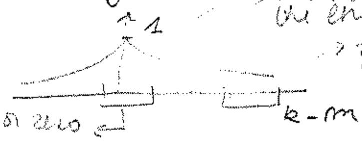
$$C_s b = \underline{b}\,\Lambda \text{ where } \Lambda = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_L \end{bmatrix} \text{ with } \lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_L$$

The DKLT is constructed iteratively by finding these largest eigenvalues to minimize the MSE. Indeed, because the eigenvector dropped corresponds to the last eigenvalues and as these last eigenvalues are always the smallest, the error is for sure the smallest possible. As a conclusion, the DKLT is a unitary transform where the b_i are orthonormal ($\underline{b}$ is unitary, $b^T = b^{-1}$, $b_i b_j = \delta_{ij}$) as they are the eigenvectors of a real symmetric matrix C_s. The coefficients are decorrelated as $C_{coeff} = b^T C_s b = b^{-1} b \Lambda = I \Lambda = \Lambda$ is a diagonal matrix. The coefficients are ordered by decreasing variance and the reconstruction error is ensuring a minimum MSE $\forall L$

To solve the problem of compression of big data or images, we [illegible] into smaller partitions with the assumption that is the pixels are far or if they are neighbours but belonging to different partition, they are not correlated. The covariance is an exponential function of the distance:

$$C_{(k,l)(m,n)} = e^{-0,125|k-m| - 0,249|l-n|}$$

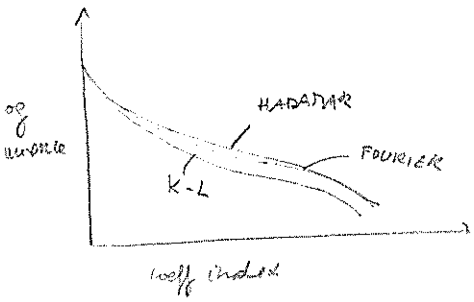
// $F_X(x) = \frac{1}{2} e^{-\lambda|x|}$ laplacian distribution



→ this behaviour comes from the energy compaction
→ few values are large

The transformation is done independently for every partition and every partition has the same KLT images and eigenvalues.

The DKLT gives the best efficiency curve for ensemble of images but the problem is that it is complex to implement as coeff + basis have to be sent while for other transfor, only the coeff are sent. Moreover, the Fourier / DCT transform achieve almost same results with simpler implementation. Thus the KLT is a theoretical minimum.



DISCRETE FOURIER TRANSFORM (DFT)

Definitions

1-D IDFT $s(i) = \sum_{k=0}^{N-1} coeff(k)\, b_k(i)$ with $b_k(i) = \frac{1}{\sqrt{N}} e^{\frac{j2\pi ik}{N}}$

1-D DFT $coeff(k) = \sum_{i=0}^{N-1} s(i)\, b_k^*(i)$ with $b_k^*(i) = \frac{1}{N} e^{-\frac{j2\pi ik}{N}}$

A NE PAS CONNAITRE

2-D IDFT $s(i,j) = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} coeff(k,l)\, b_k(i)\, b_l(j)$ with $b_{kl}(i,j) = b_k(i) b_l(j) = \frac{1}{N} e^{\frac{j2\pi(ik+jl)}{N}}$

2-D DFT $coeff(k,l) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} s(k,l)\, b_k^*(i)\, b_l^*(j)$ with $b_{kl}^*(i,j) = b_k^*(i)\, b_l^*(j) = \frac{1}{N} e^{-\frac{j2\pi(ik+jl)}{N}}$

Starting from $s(i,j)$ if N is finite, $coeff(k,l)$ always exists. This can be verified by replacing 2-D DFT into 2-D IDFT and making use of the orthogonality property of the basis functions. A similar property exists for the continuous Fourier transform ($X(\omega) = \int x(t) e^{-j\omega t} dt$) where the existence is guaranteed if the spatial image function is square integrable, i.e. the signal must be of finite energy

$coeff(k,l) = coeff_{RE}(k,l) + j\, coeff_{IM}(k,l)$ OU/ET $coeff(k,l) = |coeff(k,l)| \exp(j\,\phi(k,l))$ where

$|coeff(k,l)| = \sqrt{(coeff_{RE}(k,l))^2 + (coeff_{IM}(k,l))^2}$ and $\phi(k,l) = \tan^{-1}\left(\frac{coeff_{IM}(k,l)}{coeff_{RE}(k,l)}\right)$

↳ Fourier amplitude spectrum ↳ phase spectrum

The power spectrum density is $P(k,l) = |coeff(k,l)|^2$

Properties

1) Symmetric separability

2) Linearity : $s_1(i,j) + s_2(i,j) \Leftrightarrow coeff_1(k,l) + coeff_2(k,l)$

$a\, s(i,j) \Leftrightarrow a\, coeff(k,l)$

3) Scaling : $s(ai, bj) \Leftrightarrow \frac{1}{ab} coeff\left(\frac{k}{a}, \frac{l}{b}\right)$

4) translation in spatial/frequency domain : $s(i-i_0, j-j_0) \Leftrightarrow coeff(k,l) \exp\left(\frac{-j2\pi(ki_0 + lj_0)}{N}\right)$

$s(i,j) \exp\left(\frac{j2\pi(ik_0 + jl_0)}{N}\right) \Leftrightarrow coeff(k-k_0, l-l_0)$

Demo :

2) $\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (s_1(k,l) + s_2(k,l))\, b_k^*(i)\, b_l^*(j) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (s_1(k,l)\, b_k^*(i)\, b_l^*(j) + s_2(k,l)\, b_k^*(i)\, b_l^*(j))$

$= \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} s_1(k,l)\, b_k^*(i)\, b_l^*(j) + \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} s_2(k,l)\, b_k^*(i)\, b_l^*(j) = coeff_1(k,l) + coeff_2(k,l)$

3) $\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} s(ai,bj) \frac{1}{N} e^{-\frac{j2\pi(ik+lj)}{N}}$ let $i' = ai$ $j' = bj$

$\sum_{i'=0}^{(N-1)a} \sum_{j'=0}^{(N-1)b} s(i',j') \frac{1}{N} e^{-\frac{j2\pi(i'k/a + j'l/b)}{N}} = \frac{1}{|ab|} coeff\left(\frac{k}{a}, \frac{l}{b}\right)$

→ comes from the division of the sum upper values $(N-1)a$ ... to have $N-1$

4) $s(i-i_0, j-j_0) = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} coeff(k,l)\, b_k(i-i_0)\, b_l(j-j_0) = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} coeff(k,l) \frac{1}{N} e^{\frac{j2\pi([i-i_0]k + [j-j_0]l)}{N}}$

$= \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} coeff(k,l) \frac{1}{N} e^{\frac{j2\pi(ik+jl-i_0k-j_0l)}{N}} = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} coeff(k,l) \frac{1}{N} e^{\frac{j2\pi(ik+jl)}{N} - \frac{j2\pi(i_0k+j_0l)}{N}} = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} \underbrace{coeff(k,l)\, e^{\frac{-j2\pi(i_0k+j_0l)}{N}}}_{(QFD)} b_k(i) b_l(j)$

5) periodicity : $coeff(k,l) = coeff(k+N, l) = coeff(k, l+N) = coeff(k+N, l+N)$

If we apply the inverse transform, we obtain a periodic function $s_p(i,j)$ from which one period is equal to $s(i,j)$

.) Conjugate symmetry: if $s(i,j)$ is real $\Rightarrow$ $coeff(k,l) = coeff^*(-k,-l)$ (symmetry)

$$|coeff(k,l)| = |coeff(-k,-l)| \;\&\; \phi(k,l) = -\phi(-k,-l)$$

using properties (4) & (5) in visualization $k_0 = l_0 = \frac{N}{2}$: $s(i,j)(-1)^{i+j} \Leftrightarrow coeff\left(k-\frac{N}{2}, 1-\frac{N}{2}\right)$

.) If $s(i,j)$ is rotated over an angle θ then $coeff(k,l)$ is rotated over the same angle θ
for discrete images, this is only approximately true: a pixel is not mapped to a new pixel position after rotation. Rotation of a discrete image automatically involves interpolation as pixel at position pixel + θ might not exist.

.) Assume $sl(i,j)$ is the Laplacian ($s(i,j)$ in freq domain): $sl(i,j) \Leftrightarrow -\left(\frac{2\pi}{N}\right)^2 (k^2+l^2)\, coeff(k,l)$
The Laplacian emphasizes high frequency components and hence it also emphasizes edges

Dems.

(7) $s(i,j)\cdot e^{j\theta} \Rightarrow \sum\sum coeff(k,l)\, b_k(i) b_l(j)\, e^{j\theta} = \sum\sum \overbrace{\left[coeff(k,l)\cdot e^{j\theta}\right]}^{\text{(CFD.}} b_k(i) b_l(j)$

(8) $\Delta s(i,j) = \frac{\partial^2 s(i,j)}{\partial i^2} + \frac{\partial^2 s(i,j)}{\partial j^2}$ and $\frac{\partial^2 s(i,j)}{\partial i^2} = \sum_k \sum_l coeff(k,l) \frac{1}{N} \frac{\partial^2}{\partial i^2}\left(e^{\frac{j2\pi(ik+lj)}{N}}\right)$

and $\frac{\partial}{\partial i}\left(e^{\frac{j2\pi ik}{N}} e^{\frac{j2\pi lj}{N}}\right) = e^{\frac{j2\pi lj}{N}} \frac{j2\pi k}{N} e^{\frac{j2\pi ik}{N}} \rightarrow \frac{\partial^2}{\partial i^2}\left(e^{\cdots}\right) = e^{\frac{j2\pi lj}{N}} - \left(\frac{2\pi}{N}\right)^2 k^2 e^{\frac{j2\pi ik}{N}}$

by doing same with $\frac{\partial^2}{\partial j^2}$ we obtain: $\Delta s(i,j) = \sum_k \sum_l \underbrace{coeff(k,l)\left(-\left(\frac{2\pi}{N}\right)^2 (k^2+l^2)\right)}_{\text{(CFD}} b_k(i) b_l(j)$

In plots of Fourier coefficient images
the coeff (m,n) have a high dynamic range. The image spectrum usually decreases very rapidly with increasing frequency but the high frequency components are important since they are mainly due to sharp transitions. To adapt the contrast in the displayed images to the dynamic range of the display and the eye, a logarithmic rule is applied on the coefficients:

$disp(m,n) = \log(1+|coeff(m,n)|) \geqslant 0$ ($disp(m,n) = 0 \Leftrightarrow coeff(m,n) = 0$)

$\frac{F_{max}}{F_{min}} \rightarrow R = \frac{\log 1 + KF_{max}}{\log 1 + KF_{min}}$; what is meaning ?

?

DISCRETE COSINE TRANSFORM (DCT)

Definitions

1-D IDCT $\quad s(i) = \sum_{k=0}^{N-1} coeff(k)\, b_k(i)$ with $b_k(i) = c(k) \cos \frac{(2i+1)k\pi}{2N}$, $c(k) = \begin{cases} \frac{1}{\sqrt{N}} & \text{for } k = 0 \\ \frac{\sqrt{2}}{\sqrt{N}} & \text{for } k = 1 \ldots N \end{cases}$

1-D DCT $\quad coeff(k) = \sum_{i=0}^{N-1} s(i)\, b_k^{c}(i)$ with $b_k^{c}(i) = c(k) \cos \frac{(2i+1)k\pi}{2N}$

2-D IDCT $\quad s(i,j) = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} coeff(k,l) \left(c(k) \cos \frac{(2i+1)k\pi}{2N} \right) \left(c(l) \cos \frac{(2j+1)l\pi}{2N} \right)$

2-D DCT $\quad coeff(i,j) = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} s(k,l) \left(c(k) \cos \frac{(2k+1)i\pi}{2N} \right) \left(c(l) \cos \frac{(2l+1)j\pi}{2N} \right)$

Basis functions

slide 63 & 64 à imprimer !!!

+ pg de note (?) [illegible]

MSE of DCT

The cosine transform is almost as good as the KLT without having to send the basis and compute it, as it is already predefined. Only the coefficients have to be sent.

imprimer slide 70

Applications

Noise reduction

through the appropriate choice of a limited set of basis images and projection of the measured image on this subset of basis images, noise reduction may be obtained (if noise is white noise). Indeed, typical images have large signal energy contributions at low frequencies and decreasing towards high frequencies. white noise has a broad spectrum. As a result of this, generally, the SNR at high frequencies is much lower than at low frequencies. The goal of the projection is then to retain the low frequencies components of the measured image, i.e. the projection of the measured image on the basis low-freq. basis images

Transformation to a feature space

Sometimes the basis images and/or their associated coefficients carry a physical interpretation.

→ coefficients: in the case where the basis images have been fixed a priori and have not been derived from the data itself (KLT), only the coefficients are meaningful. Ex: in FT (Fourier), the frequency spectrum allows for analysis of the amplitude, the phase and the energy distribution

↳ basis images: in the case where the basis images are derived from data, both coeff & basis images are meaningful. Ex: KLT.

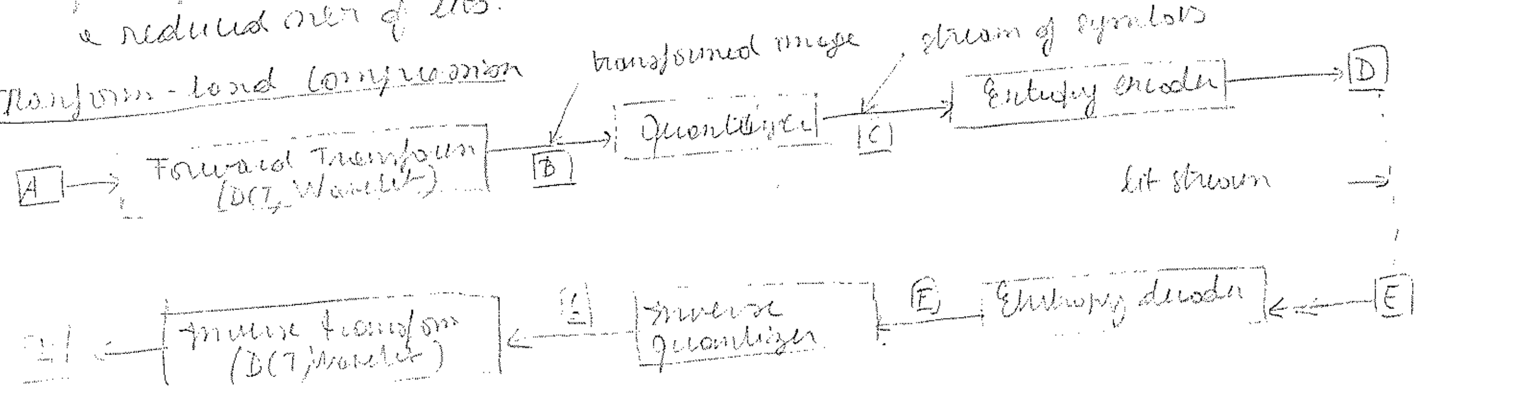
Data compression

Since neighbouring grey values in an image with N×M pixels are usually highly correlated, data compression must be possible. For global transforms, the image is written as a weighted sum of L basis images. These basis images are calculated in the same way as they can either be calculated for a class of representative images (KLT) or postulated a priori (DFT, DCT, WHT). Hence, each individual image is characterized by the set of L weighting coefficients which can be ordered in a new image: the transformed image

→ if L = NM → no loss of information occurs but there is no data compression

↳ if L < NM there is loss of information and data compression which is most efficient for coefficients with low mutual correlation (i.e. if coeff are uncorrelated, the minimal number of coeff is reached), and with low variance (i.e. a peak probability density function) since this allows coding (quantization) with a reduced nr of bits.

Transform-based compression



Ex: DCT based encoder

DC →	AC1	AC5 →	AC6
AC2	AC4	AC7	AC12
AC3	AC8	AC11	AC13
AC9 →	AC10	AC14 →	AC15

DC components & AC components for the DCT of a 4x4 block

- partition of the input image into blocks
- Apply the DCT transform on N×N blocks
- Apply scalar quantization of the DCT coefficients
- Scan and entropy encode the resulting quantization indices
  - ↳ scanning order to be defined (here by →)

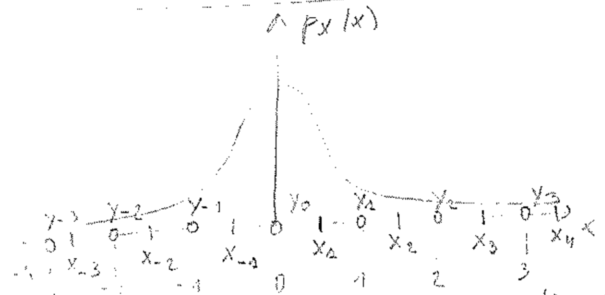
Resolution scalability:

To obtain resolution scalability, multiresolution transforms such as wavelet transform can be used. Progressively sending the wavelets subbands from lowest to highest resolution level naturally provides resolution scalability. (cf. section 5.4 slide 86/124 in part 3 - wavelets):

- sending & decoding only the lowest resolution LL subband (corner top left) corresponds to the lowest resolution version of the original image
  Ex: Lena is small in top-left subband: resolution is 1/16 of the original both horizontally and vertically
- sending 3 high frequency subbands next to the LL subband (having each the same number of samples as the LL subband) and performing a one level inverse wavelet transform will enable reconstructing a higher resolution version of the image, which will have a resolution twice as large as the LL subband on each direction (1/8 of the original resolution)

The DCT transform can also be used to provide resolution scalability. Say each DCT block has N×N samples, each DCT coefficient corresponds to a specific horizontal and vertical frequency. The top left corner corresponds to the DC component. If you send only $\frac{N}{2} \times \frac{N}{2}$ DCT coefficient in each block and perform an $\frac{N}{2} \times \frac{N}{2}$ inverse DCT transform, will give you an image which is half the size of the original image both horizontally and vertically.

SCALAR QUANTIZATION



x_i : decision points

$[x_i, x_{i+1})$: uncertainty interval

y_i : reconstruction points

(x_{-1}, x_1) : deadzone → all points are mapped to zero and seen the distribution, lots of chance to observe a zero

$p_Y(i) = P(Y = y_i) = \int_{x_i}^{x_{i+1}} p_X(x)\,dx$

if $x_i \leq X < x_{i+1}$: $Q(x) = i$, $|i| \geq 1$

if $x_{-1} < X < x_1$: $Q(x) = 0$

$Q^{-1}(i) = y_i \Leftrightarrow Q^{-1}(Q(X)) = y_i$

Based on a given distribution, the quantizer is chosen such that it minimizes the mean error distortion given by : $D_n = \sum_i \int_{x_i}^{x_{i+1}} \underbrace{|x - y_i|^n}_{\text{n-th distortion (error)}} \underbrace{p_X(x)}_{\text{chance of occurrence of the distortion}} dx$

$\sum_i$: sum on all intervals ; $\int_{x_i}^{x_{i+1}}$: interval

Such quantizer are called Lloyd-Max quantizers and satisfy :

$\int_{x_i}^{y_i} |x - y_i|^{n-1} p_X(x)\,dx = \int_{y_i}^{x_{i+1}} |x - y_i|^{n-1} p_X(x)\,dx$

for $n = 1$: $\int_{x_i}^{y_i} p_X(x)\,dx = \int_{y_i}^{x_{i+1}} p_X(x)\,dx = \frac{1}{2}\int_{x_i}^{x_{i+1}} p_X(x)\,dx$

$n = 2$: $x_k = \frac{y_{k-1} + y_k}{2}$, $k = 1, 2, \dots N-1$ (x_k are median points)

$y_k = \frac{\int_{x_k}^{x_{k+1}} x\,p_X(x)\,dx}{\int_{x_k}^{x_{k+1}} p_X(x)\,dx}$ $k = 0, 1, \dots N-1$ (y_k are centroids) (**)

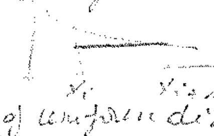
However, Lloyd-Max quantizer are not optimal in the entropy coded case because, if coding the quantized indices, $R = H(Y) = -\sum_{i=0}^{N-1} p_Y(i) \log_2 p_Y(i) \leq \log_2 N$ the minimal rate, if the coding the quantized indices, is ... the optimal quantizer should minimize the MSE subject to a ... constraint on the entropy :

$J(\lambda) = \Big(\underbrace{\sum_{i=0}^{N-1} \int_{x_i}^{x_{i+1}} (x - y_i)^2 p_X(x)\,dx}_{D_2 \to MSE} - \lambda \underbrace{\sum_{i=0}^{N-1} \int_{x_i}^{x_{i+1}} p_X(x)\,dx \left(\log_2 \int_{x_i}^{x_{i+1}} p_X(x)\,dx\right)}_{\text{constraint on entropy}} \Big)$

$\frac{\partial J}{\partial y_k} = 0 \Rightarrow y_k = \text{result (**)}$ and $\frac{\partial J(\lambda)}{\partial x_k} = 0 \Rightarrow (x_k - y_{k-1})^2 - (x_k - y_k)^2 - \lambda(\log_2 p_Y(i-1) - \log_2 p_Y(i)) = 0$

→ iterative algorithm for solution

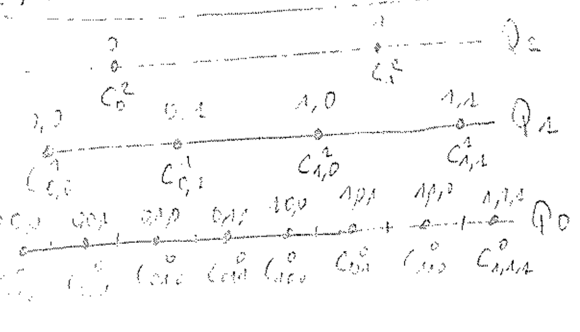
At high rates and for smooth pdf (gaussian/laplacian) the simplest quantizer which is optimal or close to the optimum in MSE sense is the uniform quantizer, as at high rates, the intervals $x_i - x_{i+1}$ are small and the pdf can be seen as uniform



In this case : $D_2 = \frac{\Delta^2}{12}$ $(\Delta = x_{i+1} - x_i, \forall i)$ (can be found by replacing p_X of uniform dist. in formula of D_n with $n = 2$)

EMBEDDED QUANTIZATION

(embedded means each level contains previous one)



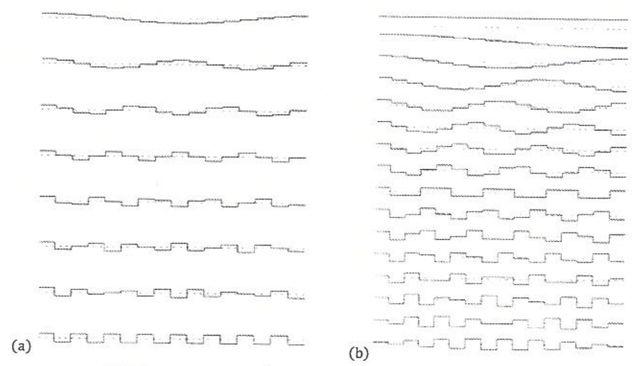
It contains a set of quantizers $Q = \{Q_N, Q_{N-1}, \dots Q_0\}$ with $Q_N(X) = k_N$, $Q_{N-1}(X) = k_N, k_{N-1}$, … , $Q_j(X) = k_N, k_{N-1}, \dots k_{j+1}$

At every step, the previous code k_N is refined to k_N, k_{N-1} until the precision level wanted is reached. Such quantizer allows adaptation of the source to the available rate.

CH 1

Basis functions for DFT and DCT

1D



DFT (Real part)

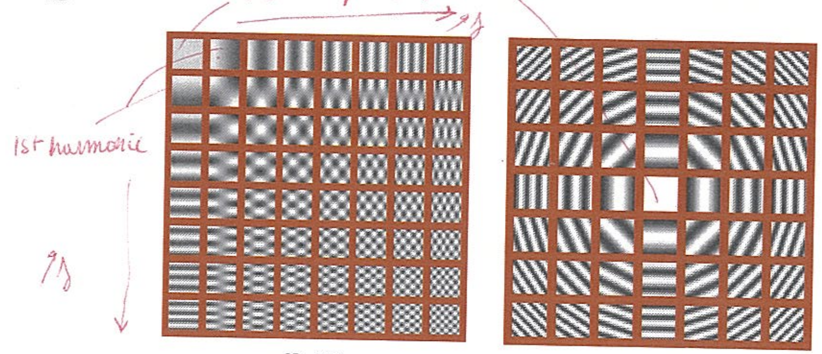
DCT

Basis functions for DFT and DCT

2D

DC component

1st harmonic

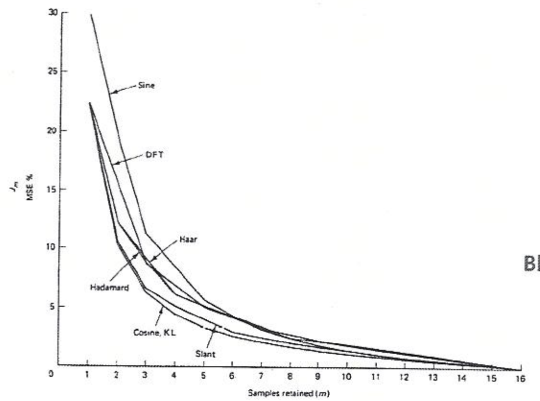


2D - DCT

2D - DFT

DCT = real part of DFT, it is a 2D separable transform but not symmetric. It compacts energy in a few components

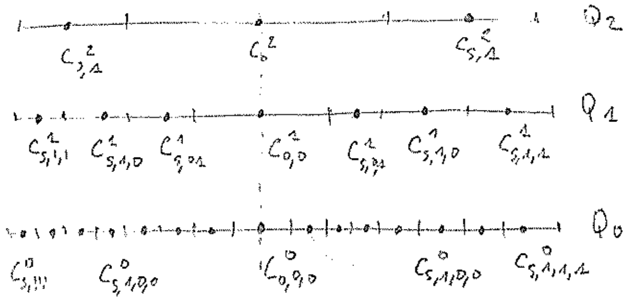
Mean Square Error of truncation compression



Block sizes 4x4

4b

EMBEDDED DEADZONE SCALAR QUANTIZATION → cf. slides Andreopoulos pg 33.



$Q_2(x) = s, 1$ / $Q_1(x) = s, 1, 0$ / $Q_0(x) = s, 1, 0, 0$ (sign)

A practical use of such quantizers:

quantization: $i_p = Q_p(X) = \begin{cases} \mathrm{sign}(X) \left\lfloor \frac{|X|}{2^p \Delta} + \frac{\xi}{2^p} \right\rfloor & \text{if } \frac{|X|}{2^p\Delta} + \frac{\xi_0}{2^p} \\ 0 & \text{else} \end{cases}$

inverse quantization: $y_i^p = Q_p^{-1}(i_p) = \begin{cases} 0 & \text{if } i_p \\ \mathrm{sign}(i_p)\left(i_p - \frac{\xi}{2^p} + \delta\right) 2^p \Delta & \text{if } i_p \end{cases}$

where $\xi < 1$ controls the size of deadzone
$0 < \delta < 1$ controls the placement of the reconstruction points in their respective cells.

$i_0 = Q_0(X)$ and $i_0 = \mathrm{sign}(i_0) \sum_{j=0}^{N-1} 2^j k_j$

$|\, s \,|\, k_{N-1} \,|\, k_{N-2} \,|\, \cdots \,|\, k_{p-1} \,|\, \cdots \,|\, k_1 \,|\, k_0 \,|$ p

i_p can be obtained by dropping the p least significant bitplanes from i_0.

proof: writing i_0 from definition: $|i_0| = \left\lfloor \frac{|X|}{\Delta} + \xi \right\rfloor$ and dropping p least significant bitplanes

$i_0^p = \mathrm{sign}(i_0) \left\lfloor \frac{|i_0|}{2^p} \right\rfloor = \mathrm{sign}(i_0) \left\lfloor \frac{\lfloor \frac{|X|}{\Delta} + \xi \rfloor}{2^p} \right\rfloor$ and by writing $\left\lfloor \frac{|X|}{\Delta} + \xi \right\rfloor = 2^p m + n$, $m \in \mathbb{Z}$, $0 \le n \le 2$

not understood

?? $(i) = \mathrm{sign}(i_0) \left\lfloor \frac{2^p m + n}{2^p} \right\rfloor \underset{?}{=} \mathrm{sign}(x) . m$ and by writing $\frac{|X|}{\Delta} + \xi = \left\lfloor \frac{|X|}{\Delta} + \xi \right\rfloor + \varepsilon$, $0 \le \varepsilon < 1$

? $= 2^p m + n + \varepsilon$

$\underset{?}{=} i_0^p$

$= \mathrm{sign}(X) \left\lfloor \frac{2^p m + n + \varepsilon}{2^p} \right\rfloor \underset{?}{=} \mathrm{sign}(X) . m$

Hence the quantization level p (i_p) is obtained by dropping the least non-significant bitplanes from the binary representation of X.

SUCCESSIVE APPROXIMATION QUANTIZATION

same scheme as embedded deadzone scalar quantization and same formulas but with $\xi = 0$ and $\delta = \frac{1}{2}$ in the definitions. SAQ can be simply implemented via thresholding, by applying a monotonically decreasing set of thresholds of the form: $T_{p-1} = \frac{T_p}{2}$

SCALABLE CODING → thèse Andreopoulos pg 18.

quality scalable → a fait en quantize

↳ apply same steps as pg 5 recto

↳ improve quality by taking more quantization steps (augmented precision) for a fixed nb of DCT coefficients

A high compression rate induces "blocking effect"

Resolution scalable → Q à 1 seul pas(?)

↳ apply same steps as pg 5 recto

↳ improve resolution by taking more DCT coefficients for a fixed number quantization steps.

↳ cf. pg 5

CHAPITRE 2 : HOUGH TRANSFORM

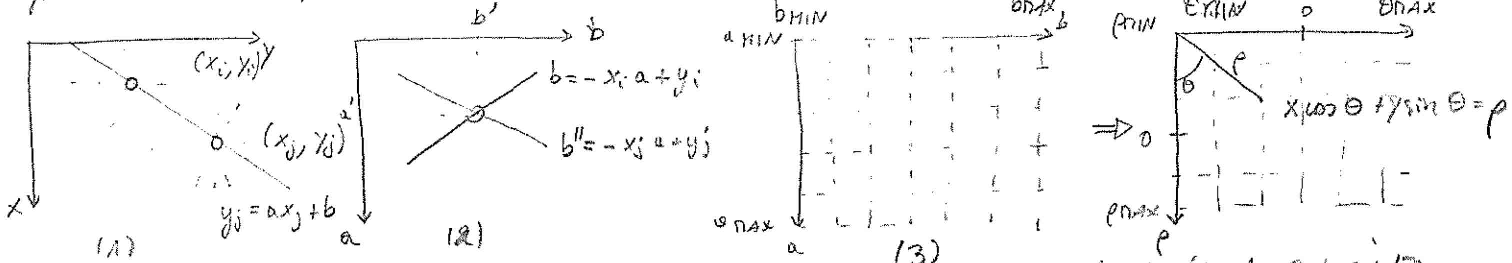
INTRODUCTION

The Hough transform is a curve detection technique that can replace edge tracking techniques by defining an algorithm that draws the line between aligned points defining an edge. The algorithm works on a thresholded (i.e binary) edge image. The advantage is that results are almost unaffected by gaps and by noise in the curves.

CASE : STRAIGHT LINES

To define a straight line, 2 approaches exist:

- find all lines determined by sets of 2 points number of lines: $\frac{n(n-1)}{2} \sim n^2$
- find out which points lie on a straight line: number of comparisons: $\frac{n^2(n-1)}{2} \sim n^3$



The basic principle of the Hough transform comes from the fact that 2 points (x_i, y_i) and (x_j, y_j) are aligned if (a, b) the line parameters match for both of them. This corresponds to an intersection in the (a, b) space. Indeed:

- the point (x_i, y_i): $y_i = ax_i + b$ equation in (x, y) space
  $b = -x_i a + y_i$ equation in (a, b) space
- the point (x_j, y_j): $y_j = ax_j + b$ equation in (x, y) space
  $b'' = -x_j a + y_j$ equation in (a, b) space

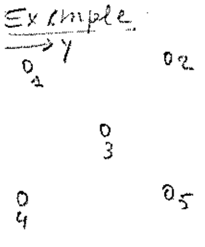
In the (a, b) space discretized of the Hough transform (3), the sum of the values of (x_i, y_i) sharing the same (a, b) give the number of points sharing the same (a, b) i.e lying on the same line in the (x, y) space. However, which values to give to a_{MIN}, b_{MIN}, a_{MAX} and b_{MAX} is an issue as they might go from $-\infty$ to $+\infty$. The polar representation of a line solves this problem: $\theta \in [0, \pi[$ & $\rho \leq$ distance of pixel from origin.

The algorithm follows:

1) quantorize the parameter space between appropriate minimum and maximum values of a & b
2) form an accumulator array $A(a, b)$ which is initialised to zero dimensions of $A(a, b)$: $a: 1, \dots K$, $b: 1, \dots L$
3) for each point (x, y) in the thresholded edge image and for each of the allowed values a_k $(k = 1 \dots K)$, calculate b and round it to the nearest allowable value b_l then increment the accumulator $A(a_k, b_l) = A(a_k, b_l) + 1$
4) local maxima in the accumulator array correspond to collinear points; the value in the accumulator array corresponds to the number of points on the line. The number of lines retained can be set by adjusting a threshold value T.

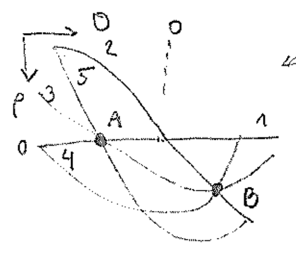
Amount of calculation: NK (usually $K \ll N$)

Example



$\theta = \pm\frac{\pi}{2}$

$\rho = \pm\sqrt{2}D$



representation of the points 1, 2, 3, 4, 5 in the ρ, θ space. The intersection of A & B of the curves means that the points 1, 3, 5 ($\cap$ A) are aligned and the points 4, 3, 2 ($\cap$ B) also.

VRIJE UNIVERSITEIT BRUSSEL — ETRO

Image Processing

The Wavelet Transform

ADRIAN MUNTEANU

Contents

1.1 Vector spaces and Inner Products

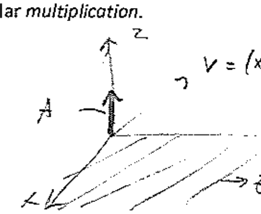
Linear Algebra:

- Vectors over *R* or *C* that are of a finite dimension *n*: $\mathbb{R}^n$ or $\mathbb{C}^n$
- Given $\{v_k\}$ a set of vectors in these spaces:
  - Does the set span the spaces, i.e. can every vector be represented as a linear combination of vectors from $\{v_k\}$?
  - Are the vectors linearly independent?
  - How can we find bases for the spaces to be spanned?
  - Given a subspace in $\mathbb{R}^n$ or $\mathbb{C}^n$ and a general vector *x*, can we find an approximation of *x* in the least-square sense that lies in the subspace?

1.1 Vector spaces and Inner Products

- A *vector space* over *R* or *C* is a set of vectors *E*, together with *addition* and scalar *multiplication*.
- Properties:
  - Commutativity: $x+y=y+x$
  - Associativity: $(x+y)+z=x+(y+z)$
  - Distributivity: $\alpha(x+y)=\alpha x+\alpha y$
  - Additive Identity: there exists 0 in *E*, such that: $x+0=x, \forall x\in E$
  - Multiplicative Identity: $1\cdot x=x, \forall x\in E$
  - Additive inverse: for all *x* in *E*, there exists a (-*x*) in *E*, such that: $x+(-x)=0$
- A subset *M* of *E* is a *subspace* of *E* if:
  - For all *x* and *y* in *M*, *x+y* is in *M*
  - For all *x* in *M*, and α in *R* or *C*, αx is in *M*.
- Given $S\subset E$, the *span* of *S* is the subspace of *E* consisting of all linear combinations of vectors in *S*.
  - Example:

$$span(S)=\left\{\sum_{i=1}^{n}\alpha_i x_i \mid \forall\alpha_i\in C, \forall x_i\in S\right\}$$



$v=(x_1,y_1,z_1)\in\mathbb{R}^3$

subspace $E=span(x,y)\subset\mathbb{R}^2$

$A\perp E \Rightarrow A\oplus E=\mathbb{R}^2$

A is the orthogonal complement of E

axe z — plan xy

allows to make measurements in a space

1.1 Vector spaces and Inner Products

- The vectors $\{x_1, x_2, \dots x_n\}$ are *linearly independent* if $\sum_i \alpha_i x_i = 0 \Rightarrow \alpha_i = 0, \forall i$
- A *basis* in *E* is a subset of linearly independent vectors $\{x_1, x_2, \dots x_n\}$ for which: $E = span(x_1 \dots x_n)$

- *E* is *infinite dimensional* if it contains an infinite linearly independent set of vectors;
  eg.: the space of infinite sequences is spanned by the infinite set $\{\delta(n-k)\}_{k\in Z}$
- The *inner product* on *E* over *C* is a function defined on *E* X *E* with the properties:
  - $\langle x+y, z\rangle = \langle x,z\rangle + \langle y,z\rangle$
  - $\langle x, \alpha y\rangle = \alpha\langle x,y\rangle$
  - $\langle x,y\rangle^* = \langle y,x\rangle$
  - $\langle x,x\rangle \geq 0$ and $\langle x,x\rangle = 0$ if and only if $x \equiv 0$

- Examples:
  - The inner product for complex valued functions over R: $\langle f,g\rangle = \int_{-\infty}^{\infty} f^*(t)g(t)dt$
  - The inner product for complex valued sequences over Z: $\langle x,y\rangle = \sum_{n=-\infty}^{\infty} x^*[n]y[n]$

- The *norm* of a vector is defined as: $\|x\| = \sqrt{\langle x,x\rangle}$

1.1 Vector spaces and Inner Products

- A *vector x* is said to be *orthogonal* to a set of vectors $S = \{y_i\}$, *if* $\langle x, y_i\rangle = 0, \forall i$
  - Notation: $x \perp S$
- *Two spaces* S_1 and S_2 are called to be *orthogonal* if for $\forall x_i \in S_1,\ x_i \perp S_2$
  - Notation: $S_1 \perp S_2$
- A *set of vectors* $\{x_1, x_2 \dots\}$ is called *orthogonal* if $\forall i,j, i\neq j,\ x_i \perp x_j$
- A *orthonormal* set of vectors is an orthogonal set with unit norm: $\langle x_i, x_j\rangle = \delta[i-j]$
- A vector space equipped with an inner product is called an *inner product space.*
- A sequence of vectors $\{x_n\}$ in *E* converges to a vector *x* in *E* if: $\|x_n - x\| \to \infty|_{n\to\infty}$
- A sequence of vectors $\{x_n\}$ is called a *Cauchy* sequence if: $\|x_n - x_m\| \to 0,\ as\ n,m \to \infty$
- If every Cauchy sequence in *E* converges to a vector in *E*, then *E* is called *complete*
- *A complete inner product space is called a Hilbert space.*

↳ Hilbert space – complete – with inner product.

$\langle x_k, y\rangle = \langle x_k, \sum_k \alpha_k x_k\rangle$
$= \sum_k \alpha_k \langle x_k, x_k\rangle$
$\to\ = \alpha_k$
1 pour 1k en produit

1.1 Vector spaces and Inner Products

- Given a Hibert space *E* and a subspace *S*, the *orthogonal complement* of *S* in *E*, denoted by $S^\perp$ is the set:

$$\{x \in E, x \perp S\}$$

- *S* is *closed* if it contains all the limits of sequences of vectors in *S*.
- Given a vector *x* in *E*, there exists a unique *y* in *S* and a unique *z* in $S^\perp$ such that: $x = y + z$.
  - Result: $E = S \oplus S^\perp$
- Examples of Hilbert spaces
- *Space of square-summable sequences* - $l^2(Z)$
  - inner product: $\langle x,y\rangle = \sum_{n=-\infty}^{\infty} x[n]^* y[n]$
  - norm: $\|x\| = \sqrt{\langle x,x\rangle} = \sqrt{\sum_{n\in Z} |x[n]|^2} (< \infty$ → finite energy
- *Space of square-integrable functions* - $L^2(R)$
  - inner product: $\langle f,g\rangle = \int_{t\in R} f(t)^* g(t)dt$
  - norm: $\|f\| = \sqrt{\langle f,f\rangle} = \sqrt{\int_{t\in R} |f(t)|^2 dt}$

1.1 Vector spaces and Inner Products

Orthonormal Bases in Hilbert spaces

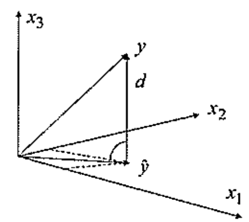
- $S = \{x_i\}$ form an orthonormal basis in E if $\langle x_i, x_j\rangle = \delta[i-j]$ and $\forall y \in E, \exists \alpha_k$, such that:

$$y = \sum_k \alpha_k x_k \qquad \alpha_k = \langle x_k, y\rangle$$

to prove (using def of y and property of ⊥)

Orthogonal Projection and Least-square Approximation

- Given *E* a Hilbert space, *S* a closed subspace of *E*, $\{x_1, x_2, \dots\}$ an orthonormal basis in *S*, and a vector $y \in E$, find the best approximation of y in *S*.
  - $d = y - \hat{y}$
  - $\|d\| = \|y - \hat{y}\|$ is minimal for $\hat{y} = \sum_i \langle x_i, y\rangle x_i$
  - Note that: $d \perp S$
  - Note that: $\|y\|^2 = \|\hat{y}\|^2 + \|d\|^2$



99

1.1. Vector spaces and Inner Products

Biorthogonal Bases

A system $\{x_k, \tilde{x}_k\}$ constitutes *biorthogonal base* in a Hibert space *E* if and only if:

- For all *i*, *j* in *Z*: $\langle x_i, \tilde{x}_j \rangle = \delta[i-j]$
- The sets $\{x_k\}$ and $\{\tilde{x}_k\}$ constitute each a *frame* in *E*, that is, for all *y* in *E*, there exist strictly positive constants (called *frame bounds*) such that: $A, B, \tilde{A}, \tilde{B}$

$$A\|y\|^2 \le \sum_k |\langle x_k, y\rangle|^2 \le B\|y\|^2$$ → if $A = B = 1$ → parseval theorem: energy in transform domain = energy in spatial domain ⇒ energy conservation

$$\tilde{A}\|y\|^2 \le \sum_k |\langle \tilde{x}_k, y\rangle|^2 \le \tilde{B}\|y\|^2$$ → energy of signal coefficient should be bounded.

- Bases that satisfy these constraints are called *Riesz bases*.
- If $A = B$ the frame $\{x_k\}$ is called a *tight frame*.

Expansion formula: $y = \sum_k \langle x_k, y\rangle \tilde{x}_k = \underbrace{\sum_k \langle \tilde{x}_k, y\rangle x_k}$

→ reconstruction basis ≠ construction basis of the transform does not have to be the same

2. Why Wavelets?

2.1. Fourier Theory

The signal to be expanded is either continuous or discrete in time.

The expansion involves an integral (a transform) or a summation (a series).

⇒ Four possible combinations of continuous/discrete time and integral/series expansions

General Case

$\{\psi_\omega\}$, $\{\psi_k\}$ a continuous and a discrete set of basis functions respectively.

(a) ***Continuous-time Integral Expansion***

$$x(t) = \int X_\omega \psi_\omega(t) d\omega, \quad X_\omega = \langle \tilde{\psi}_\omega(t), x(t) \rangle$$

(b) ***Continuous-time Series Expansion***

$$x(t) = \sum_k X_k \psi_k(t), \quad X_k = \langle \tilde{\psi}_k(t), x(t) \rangle$$

(c) ***Discrete-time Integral Expansion***

$$x[n] = \int X_\omega \psi_\omega[n] d\omega, \quad X_\omega = \langle \tilde{\psi}_\omega[n], x[n] \rangle$$

(d) ***Discrete-time Series Expansion***

$$x[n] = \sum_k X_k \psi_k[n], \quad X_k = \langle \tilde{\psi}_k[n], x[n] \rangle$$

2.1. Fourier Theory

Fourier Case

a) Continuous-time Fourier Transform **(CTFT)** - Fourier Transform

b) Continuous-time Fourier Series **(CTFS)** - Fourier Series

c) Discrete-time Fourier Transform **(DTFT)**

d) Discrete-time Fourier Series **(DTFS)**

All cases: $\{\psi\} = \{\tilde{\psi}\}$

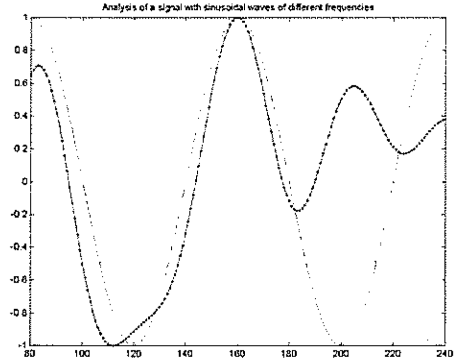
2.1.1 Fourier Transform

The Fourier Transform in $L^1(\mathbb{R})$

$$F(\omega)=\int_{-\infty}^{\infty} f(t)\,e^{-j\omega t}\,dt, \qquad (1)$$

$$\psi_\omega(t)=e^{j\omega t} \qquad F(\omega)=\langle e^{j\omega t}, f(t)\rangle$$

It measures "the intensity" of the oscillations at the frequency ω in $f(t)$



2.1.1. Fourier Transform

The Inverse Fourier Transform in $L^1(\mathbb{R})$

If $f \in L^1(\mathbb{R})$ and $F \in L^1(\mathbb{R})$ then: $f(t)=\frac{1}{2\pi}\int_{-\infty}^{\infty} F(\omega)\,e^{j\omega t}d\omega$ (2)

- Synthesize $f(t)$ as a sum of sinusoidal waves $e^{j\omega t}$ of amplitude $F(\omega)$

Note 1: the inversion formula is exact if f is continuous.

Note 2: the inversion formula is exact if $f(t)$ is defined as $\left(f(t^+)+f(t^-)\right)/2$ at a point of discontinuity

Fourier Transform in $L^2(\mathbb{R})$

The formulas above hold in the L^2 sense: if $\hat{f}(t)$ is the result of (1) followed by (2), then:

$$\left\|f(t)-\hat{f}(t)\right\|_2=0$$

2.1.1. Fourier Transform

Properties

Linearity, Shifting, Scaling, Diferentiation/Integration...

Convolution theorem

$$h(t)=f(t)*g(t)=\int f(\tau)g(t-\tau)d\tau \Rightarrow \quad H(\omega)=F(\omega)\cdot G(\omega)$$

$$f(t)*g(t)\leftrightarrow F(\omega)\cdot G(\omega)$$

$$f(t)\cdot g(t)\leftrightarrow \frac{1}{2\pi}F(\omega)*G(\omega)$$

Alternative view of the convolution theorem: the complex exponentials $e^{j\omega t}$ are eigenfunctions of the convolution operator.

$$g(t)*e^{j\omega t}=\int_{-\infty}^{\infty} e^{j\omega(t-\tau)}g(\tau)d\tau=e^{j\omega t}\int_{-\infty}^{\infty} e^{-j\omega\tau}g(\tau)d\tau=G(\omega)e^{j\omega t}$$

Parseval's formula $\int_{-\infty}^{\infty} f^*(t)g(t)dt=\frac{1}{2\pi}\int_{-\infty}^{\infty} F^*(\omega)G(\omega)d\omega$

Energy conservation property $\int_{-\infty}^{\infty}|f(t)|^2dt=\frac{1}{2\pi}\int_{-\infty}^{\infty}|F(\omega)|^2d\omega$

2.1.2. Fourier Series

A periodic signal, $f(t)=f(t+kT), k\in Z$ can be expanded in its Fourier series:

$$f(t)=\sum_{k=-\infty}^{\infty}F[k]e^{jk\omega_0 t},\ \omega_0=\frac{2\pi}{T}; \qquad (3)$$

$$x(t)=\sum_k X_k\psi_k(t)\Rightarrow \psi_k(t)=e^{jk\omega_0 t}$$

- Linear combination of complex exponentials with frequencies $k\omega_0$

Fourier Coefficients: $F[k]=\frac{1}{T}\int_{-T/2}^{T/2} f(t)e^{-jk\omega_0 t}dt$ (4)

Note 1: If $f(t)$ is continuous, then the series converges uniformly to $f(t)$

Note 2: If $f(t)$ is square-integrable over a period, but not necessarily continuous, then the series converges to $f(t)$ in L^2 sense.

$$k|_{-N}^{N}\Rightarrow \hat{f}_N(t); \qquad \left\|f(t)-\hat{f}_N(t)\right\|\to 0\Big|_{N\to\infty}$$

- Convergence plagued by Gibbs phenomenon

79

2.1.2. Fourier Series

Properties

The set of functions used in (3) is a complete orthonormal system for the interval $[-T/2, T/2]$:

$$\varphi_k(t) = \left(1/\sqrt{T}\right)e^{jk\omega_0 t},\ t \in [-T/2, T/2], k \in Z \Rightarrow \langle \varphi_k(t), \varphi_l(t)\rangle_{[-T/2,T/2]} = \delta[k-l]$$

Parseval relation:

$$\langle g(t), f(t)\rangle = T \cdot \langle F[k], G[k]\rangle \Rightarrow \int_{-T/2}^{T/2} |f(t)|^2 dt = T \cdot \sum_{k=-\infty}^{\infty} |F[k]|^2$$

Best Approximation Property

$$\left\| f(t) - \sum_{k=-N}^{N} \langle \varphi_k, f\rangle \varphi_k(t) \right\| \le \left\| f(t) - \sum_{k=-N}^{N} \alpha_k \varphi_k(t) \right\|$$, α_k an arbitrary set of coefficients.

- The Fourier series coefficients are the best ones for an approximation in the span of $\{\varphi_k(t)\}, k \in [-N, N]$

Fourier series can be used for non-periodic signals via periodization.

Problem: discontinuities at the boundaries.

2.1.3. Discrete-Time Fourier Transform

Given a sequence $\{f[n]\}_{n\in Z}$ in $l^1(Z)$, its discrete-time Fourier transform is defined by

$$F\left(e^{j\omega}\right) = \sum_{n=-\infty}^{\infty} f[n]e^{-j\omega n}, \text{ which is } 2\pi \text{ periodic.} \qquad (5)$$

Inverse transform:

$$f[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} F\left(e^{j\omega}\right)e^{j\omega n} d\omega \qquad (6)$$

$$x[n] = \int X_\omega \psi_\omega[n] d\omega \Rightarrow \psi_\omega[n] = e^{j\omega n}$$

Note 1: If $f[n]$ is in $l^2(Z)$, we have convergence in l^2 sense as the summation limits in (5) go to infinity.

Note 2: $f[n]$ is obtained by sampling a continuous time signal $f(t)$ at instants nT.

Result: DTFT is related to the Fourier transform $F_c(\omega)$ of $f(t)$:

$$F\left(e^{j\omega}\right) = \frac{1}{T}\sum_{k=-\infty}^{\infty} F_c\left(w - k\frac{2\pi}{T}\right)$$

2.1.3. Discrete-Time Fourier Transform

Properties

The properties of the FT are carried over by DTFT

Convolution:

$$f[n]*g[n] = \sum_{l=-\infty}^{\infty} f[n-l]g[l] = \sum_{l=-\infty}^{\infty} f[l]g[n-l] \leftrightarrow F\left(e^{j\omega}\right)\cdot G\left(e^{j\omega}\right)$$

Parseval's relation:

$$\sum_{n=-\infty}^{\infty} f^*[n]g[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} F^*\left(e^{j\omega}\right)G\left(e^{j\omega}\right)d\omega$$

Energy conservation:

$$\sum_{n=-\infty}^{\infty} |f[n]|^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi} \left|F\left(e^{j\omega}\right)\right|^2 d\omega$$

2.1.4. Discrete-Time Fourier Series

A periodic discrete signal, $f[n] = f[n + lN], l \in Z$ has its discrete-time Fourier series representation given by:

$$F[k] = \sum_{n=0}^{N-1} f[n]e^{-jn \cdot k 2\pi/N}, \quad k \in Z \qquad (7)$$

Inverse DTFS representation: $$f[n] = \frac{1}{N}\sum_{k=0}^{N-1} F[k]e^{jn \cdot k 2\pi/N}, \quad n \in Z \qquad (8)$$

$$x[n] = \sum_k X_k \psi_k[n], \Rightarrow \psi_k[n] = e^{jn \cdot k 2\pi/N}$$

All the properties of the FT hold.

Convolution $\Rightarrow$ Circular Convolution

- $f_0[n], g_0[n]$, one period of $f[n], g[n]$
- $f_0[n] = \begin{cases} f[n], 0 \le n \le N-1 \\ 0, \textit{otherwise} \end{cases}$

2.1.4. Discrete-Time Fourier Series

Circular Convolution

$$f[n]*g[n]=\sum_{l=0}^{N-1} f[n-l]g[l]=\sum_{l=0}^{N-1} f_0[(n-l)\bmod N]g_0[l]=$$
$$=f_0[n]\circ g_0[n]\leftrightarrow F[k]\cdot G[k]$$

Parseval's relation:

$$\sum_{n=0}^{N-1} f^*[n]g[n]=\frac{1}{N}\sum_{k=0}^{N-1} F^*[k]G[k]$$

Discrete Fourier Transform (DFT)

The same formulas as (7) and (8), except that $f[n]$ and $F[k]$ are defined only for: $n,k\in\{0,\ldots,N-1\}$

Convolution matrix C:

$$C=\begin{pmatrix} f[0] & f[1] & . & . & f[N-1] \\ f[N-1] & f[0] & . & . & f[N-2] \\ . & . & . & . & . \\ . & . & . & . & . \\ f[1] & f[2] & . & . & f[0] \end{pmatrix}$$

2.1.4. Discrete-Time Fourier Series

Circular Convolution

$$f\circ g=\mathbf{C}g=\mathbf{F}^{-1}\Lambda\mathbf{F}g,\quad \mathbf{F}[n][k]=e^{-jn\cdot k2\pi/N},\qquad \mathbf{F}^{-1}[n][k]=(1/N)e^{jn\cdot k2\pi/N}$$

Λ is a diagonal matrix with $F[k]$ on its diagonal.

Another view:

- **C** is diagonalized by **F**
- The complex exponential sequences $\{e^{j(2\pi/N)nk}\}$ are eigenvectors for the convolution matrix **C**, with eigenvalues $F[k]$.

$$\hat{f}=\mathbf{F}f,\quad \hat{g}=\mathbf{F}g$$

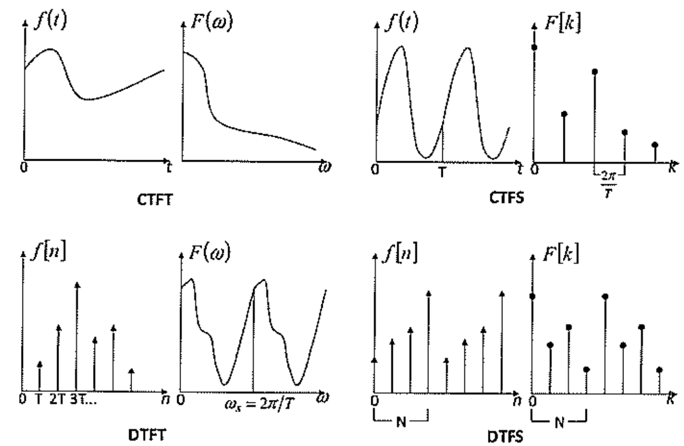
Parseval's relation:

$$\hat{f}^*\hat{g}=(\mathbf{F}f)^*(\mathbf{F}g)=f^*\mathbf{F}^*\mathbf{F}g=Nf^*g;\qquad \mathbf{F}^{-1}=\frac{1}{N}\mathbf{F}^*$$

2.1.5 Various Flavors of Fourier Transforms

Transform	Time	Frequency	Analysis Synthesis
Fourier Transform CTFT	Continuous	Continuous	$F(\omega)=\int_t f(t)e^{-j\omega t}dt$ $f(t)=1/2\pi\int_\omega F(\omega)e^{j\omega t}d\omega$
Fourier Series CTFS	Continuous Periodic	Discrete	$F[k]=1/T\int_{-T/2}^{T/2} f(t)\,e^{-j2\pi kt/T}dt$ $f(t)=\sum_k F[k]\,e^{j2\pi kt/T}$
Discrete-Time Fourier Transform DTFT	Discrete	Continuous Periodic	$F(e^{j\omega})=\sum_n f[n]e^{-j\omega n}$ $f[n]=1/2\pi\int_{-\pi}^{\pi} F(e^{j\omega})e^{j\omega n}d\omega$
Discrete-Time Fourier Series DTFS	Discrete Periodic	Discrete Periodic	$F[k]=\sum_{n=0}^{N-1} f[n]e^{-jn\cdot k2\pi/N}$ $f[n]=1/N\sum_{k=0}^{N-1} F[k]e^{jn\cdot k2\pi/N}$

2.1.5 Various Flavors of Fourier Transforms



CASE: PARAMETRIC CURVES

In this case, the goal is, given n points v_i in an image, find subsets of these points that lie on curves $c(x, y, p_1 \ldots p_N) = 0$ parameters p_i. The dimension of the ρ, θ space depends on the number of parameters (of the order) of the curve. The higher the order, the higher the dimension of the accumulator.
The algorithm is similar but the 3rd step has to be modified because there is an exponential growth of the calculations and of the size of the accumulator with the number of parameters. To save computations in the accumulator, gradient orientation information is used.

Ex: a 3 parameter curve (a, b, r): $(x-a)^2 + (y-b)^2 = r^2$: for each discrete value of r, restrict the range and quantization steps of the circle centers $(a,$

CHAPTER 3: WAVELET TRANSFORM

FUNDAMENTALS OF SIGNAL DECOMPOSITION (from slide 1-84: not for exam, only read)

important part 4x4 slide 84 85

INTRODUCTION TO WAVELET TRANSFORM

The Fourier transform expresses a signal from exponentials, extracting the frequency component of the function in time $f(t)$. The problem is that, if this gives a very precise frequency localization of the signal (Dirac pulse at right frequency), the localization in time is inexistent because of the analysis from $-\infty$ to $+\infty$ of the signal. The goal of the wavelet transform is thus to localize the analysis via localized basis functions in time and frequency by capturing signal transient. For example:

graph A

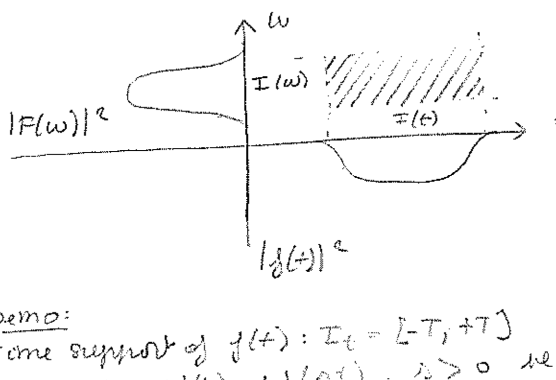
The part 1 of the signal should have a different frequency content than part 2, thus it be separated into 2 different analysis rather than 1 fit on the whole signal such as Fourier.

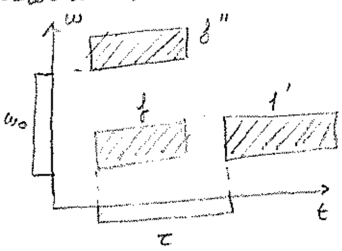
$$F(\omega) = \underbrace{\int_{-\infty}^{+\infty}}_{\text{time interval}} f(t)\, \underbrace{e^{-j\omega t}}_{\text{extraction in frequency}} dt$$

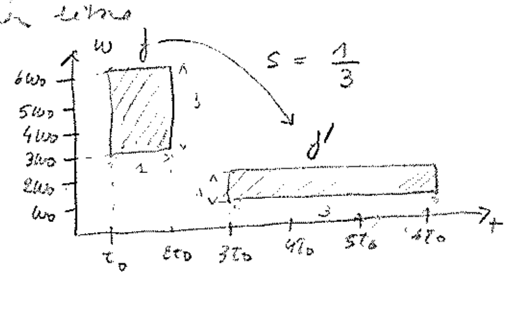
TIME-FREQUENCY REPRESENTATION

A tile is defined in the time-frequency plane as containing 90% of the energy of the time and frequency domain functions. It is centred around the centers of gravity of $|f(t)|^2$ and $|F(\omega)|^2$. The basis function defined by it is of finite localization in time and frequency. Its properties are:

- time shift τ → shifting of the tile by τ in time axis
- modulation by $e^{j\omega_0 t}$ → shifting of the tile by ω_0 in frequency axis
- scaling by s ($f'(t) = f(st)$) → $I'_t = \frac{I_t}{s}$ and $I'_\omega = s I_\omega$ i.e. resolution in frequency is traded for resolution in time







Demo:

Time support of $f(t)$: $I_t = [-T, +T]$

By defining $f'(t) = f(st)$, $s > 0$ we have

$-T \leq t \leq T \Rightarrow -T \leq t' = st \leq T \Rightarrow -\frac{T}{s} \leq t \leq \frac{T}{s}$ thus

time support of $f'(t)$: $I'_t = \frac{I_t}{s}$

- frequency support of $F(\omega)$: $I_\omega = [\omega_{min}, \omega_{max}]$

By defining $F'(\omega) = \int_{-T}^{T} f'(t)\, e^{-j\omega t} dt = \int_{-T/s}^{T/s} f(st)\, e^{-j\omega t} dt$

change $st \to t'$

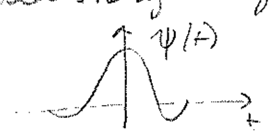
$$= \frac{1}{s}\int_{-T}^{T} f(t')\, e^{-j\left(\frac{\omega}{s}\right)t'} = \frac{1}{s} F\left(\frac{\omega}{s}\right)$$

$\omega_{min} \leq \omega \leq \omega_{max}$, $\omega_{min} \leq \omega' = \frac{\omega}{s} \leq \omega_{max}$

$\Rightarrow s\,\omega_{min} \leq \omega \leq s\,\omega_{max}$

frequency support of $f'(t)$ is $I'_\omega = s I_\omega$

The basis functions used for the wavelet transform uses these properties. They are defined as: $\psi_{a,b}(t) = \underbrace{\frac{1}{\sqrt{a}}}_{3}\, \psi\Big(\underbrace{\frac{t-b}{a}}_{2}\Big)^{2}$ and $\text{Transform}_{a,b} = \langle f(t), \psi_{a,b}(t)\rangle$



1. Scaling by a

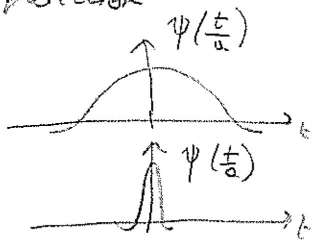
The notion of scale is similar to that used in geographical maps < large scale = global view; small scale = detailed view

The scale can also be seen as a frequency analyzer in the case when the basis function act as a band-pass filter < large scale = low frequency basis function; small scale = high frequency basis function

Thus the scale is closely related to the inverse frequency.

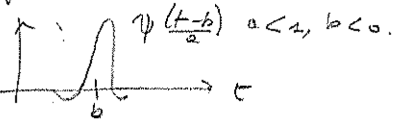
→ large scale : $a > 1$, $\psi_{a,b}(t)$ identifies long term trend in $f(t)$
↳ 1 on graph A

→ small scale : $a < 1$, $\psi_{a,b}(t)$ identifies short term behaviour in $f(t)$
↳ 2 on graph A



2. Shifting by b

It allows to move the basis function in time



3. Normalization factor

This factor ensures that the scaled ψ_a has the same energy as the original ψ

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

$\int |\psi_{a,0}(t)|^2 dt = \int \frac{1}{\sqrt{a}}\psi\left(\frac{t}{a}\right)\cdot \frac{1}{\sqrt{a}}\psi^*\left(\frac{t}{a}\right) dt = \int \psi(t')\psi(t'^*)\, dt' = \int |\psi(t)|^2 dt$ → = ok

The tiles cannot be made arbitrarily as small as possible to have sharp analysis both in time and in frequency simultaneously. The uncertainty principle defines the limitation on the width and length of the tiles. It is defined as

$$\boxed{\Delta t^2 \cdot \Delta\omega^2 \geq \frac{1}{4}}$$

if $\psi(t)$ vanishes faster than $\frac{1}{\sqrt{t}}$ as $t \to \infty$

where $\Delta t^2 = \dfrac{\int t^2 |\psi(t)|^2 dt}{\int |\psi(t)|^2 dt}$ is the resolution in time

$\Delta\omega^2 = \dfrac{\int \omega^2 |\Psi(\omega)|^2 d\omega}{\int |\Psi(\omega)|^2 d\omega}$ is the resolution in frequency

$\psi(t)$ is a basis function with Fourier transform $\Psi(\omega)$

$\psi(t)$ is a basis function centered around origin in time & frequency satisfying

$\int t|\psi(t)|^2 dt = 0$ and $\int \omega |\Psi(\omega)|^2 d\omega = 0$

Equality holds for Gaussian functions, $\psi(t) = \sqrt[4]{\frac{2\alpha}{\pi}}\, e^{-\alpha t^2}$

Demo:

$$\Delta t^2 \Delta\omega^2 \geq \frac{1}{4} \Leftrightarrow \left(\int_{-\infty}^{+\infty} t^2 |\psi(t)|^2 dt\right)\left(\int_{-\infty}^{+\infty} \omega^2 |\Psi(\omega)|^2 d\omega\right) \geq \frac{1}{4}\left(\int_{-\infty}^{+\infty} |\psi(t)|^2 dt\right)\left(\int_{-\infty}^{+\infty} |\Psi(\omega)|^2 d\omega\right)$$

By Parseval's formula : $\int_{-\infty}^{+\infty} |\psi(t)|^2 dt = \frac{1}{2\pi}\int_{-\infty}^{+\infty} |\Psi(\omega)|^2 d\omega$

$$\geq \frac{\pi}{2}\left(\int_{-\infty}^{+\infty} |\psi(t)|^2 dt\right)^2$$

By $\psi'(t) \to j\omega\Psi(\omega)$ and $\psi'(t)^* \to -j\omega\Psi^*(\omega)$ and Parseval : $\int_{-\infty}^{+\infty} |\psi'(t)|^2 dt = \frac{1}{2\pi}\int_{-\infty}^{+\infty} \omega^2 |\Psi(\omega)|^2 d\omega$

$$\Leftrightarrow \left(\int_{-\infty}^{+\infty} t^2 |\psi(t)|^2 dt\right)\left(\int_{-\infty}^{+\infty} |\psi'(t)|^2 dt\right) \geq \frac{1}{4}\left(\int_{-\infty}^{+\infty} |\psi(t)|^2 dt\right)^2$$

By Cauchy-Schwartz inequality : $\left(\int_{-\infty}^{+\infty} |f(t)|^2 dt\right)\left(\int_{-\infty}^{+\infty} |g(t)|^2 dt\right) \geq \left|\int_{-\infty}^{+\infty} f(t) g(t)\, dt\right|^2$

By replacing the variables : $f(t) = t\psi(t)$ and $g(t) = \psi'(t)$

$$\Leftrightarrow \left(\int_{-\infty}^{+\infty} t^2 |\psi(t)|^2 dt\right)\left(\int_{-\infty}^{+\infty} |\psi'(t)|^2 dt\right) \geq \left(\left|\int_{-\infty}^{+\infty} t\,\psi(t)\psi'(t)\, dt\right|\right)^2 =: I^2$$

thus $I : \int_{-\infty}^{+\infty} t\,\psi(t)\psi'(t)\, dt = t\psi^2(t)\Big|_{-\infty}^{+\infty} - \int_{-\infty}^{+\infty} (\psi(t) + t\psi'(t))\psi(t)\, dt = 0 - \int_{-\infty}^{+\infty} \psi^2(t)\, dt - I$

↳ vanish from hurdle

$$I = \frac{1}{2}\int_{-\infty}^{+\infty} \varphi^2(t)\,dt \Rightarrow I^2 = \frac{1}{4}\left(\int_{-\infty}^{+\infty} \varphi^2(t)\,dt\right)^2$$

The equality holds for: $\left.\begin{array}{l} \varphi'(t) = kt\,\varphi(t) \\ k = -2\alpha \\ \int_{-\infty}^{+\infty} \varphi^2(t)\,dt = 1 \end{array}\right\} \Rightarrow \varphi(t) = \sqrt[4]{\frac{2\alpha}{\pi}}\, e^{-\alpha t^2}$

CONTINUOUS SHORT-TIME FOURIER TRANSFORM

The goal is to improve the localization properties of the Fourier transform

Forward transform: $STFT_f(\tau,\omega) = \int_{-\infty}^{+\infty} f(t)\, w^*(t-\tau)\, e^{-j\omega t}\,dt$ window time shift

$= \langle g_{\omega,\tau}(t), f(t)\rangle$ where $g_{\omega,\tau}(t) = w(t-\tau)\,e^{j\omega t}$ is the windowed complex exponential

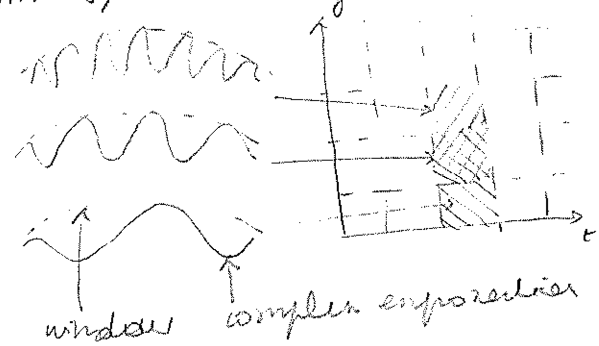
It is the inner product between the signal and the "shifts and modulates of our elementary window"

Inverse transform: $f(t) = \underbrace{\frac{1}{2\pi\|w(t)\|^2}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}}_{\text{superposition}} \underbrace{STFT_f(\omega,\tau)}_{\text{coefficient of function in transform domain}}\, g_{\omega,\tau}(t)\,d\omega\,d\tau$

Any classic window used for Fourier analysis is suitable for STFT transform. Smooth windows are preferred (e.g. Hanning window $w(t) = \begin{cases} [1+\cos(2\pi t/T)]/2 & t\in[-\frac{T}{2},\frac{T}{2}] \\ 0 & \text{else} \end{cases}$)
It is also convenient to choose the window such that $\|w(t)\| = \int |w(t)|^2 dt = 1$, i.e. normalized window. Gaussian window (e.g. Gabor: $w(t) = \beta e^{-\alpha t^2}$, $\alpha,\beta > 0$ with α controlling the width and β normalization factor) allow to reach equality in the uncertainty principle.

The Parseval's formula is written as: $\int_{-\infty}^{+\infty} |f(t)|^2 dt = \frac{1}{2\pi}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} |STFT_f(\omega,\tau)|^2\, d\omega\, d\tau$

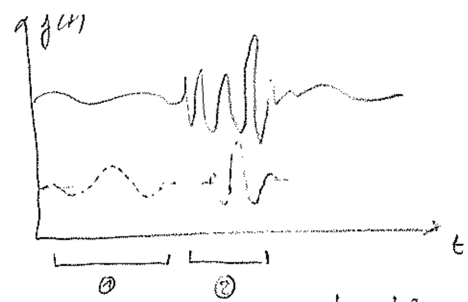
and as the time-frequency resolution of each elementary function is constant, it is natural to discretize the STFT on a rectangular grid $(m\omega_0, n\tau_0)$.



Once the window size is chosen, all frequencies are analyzed with the same time and frequency resolution, unlike what happens in the Wavelet Transform. It is not possible to distinguish different behaviours within a window spread. The alternative is to use a generalized STFT with multiple window sizes but this implies overcomplete representations (too much info)

CONTINUOUS WAVELET TRANSFORM

The drawback introduced by the STFT is the constant resolution in time and frequency. To overcome this, the wavelet transform supposes the equality in the Heisenberg principle (uncertainty principle): $\Delta t \cdot \Delta\omega = \frac{1}{2}$ which makes constant-relative bandwidth analysis: $\frac{\Delta f}{f} = cte \Rightarrow \Delta f = c \cdot f \cdot \alpha\ \Delta\omega = 2\pi \Delta f \Rightarrow$
$\Delta t\ 2\pi c f = \frac{1}{2} \Rightarrow \Delta t = \frac{1}{4\pi c \cdot f}$ thus $\begin{cases} HF \Rightarrow f\uparrow & \Delta f\uparrow \text{ (bad localization in frequency)} \quad \Delta t\downarrow \text{ (good loc. in time)} \\ LF \Rightarrow f\downarrow & \Delta f\downarrow \text{ (good localization in frequency)} \quad \Delta t\uparrow \text{ (bad loc. in time)} \end{cases}$



① good localization in frequency: discern between closed low-frequencies, the transform is able to discriminate the neighbouring frequencies

② good localization in time: discern between closed spikes in time

The transform can be seen as a real band-pass filter with impulse response $\psi(t)$, zero mean, $\int_{-\infty}^{+\infty}\psi(t)\,dt=0$ and unit energy.

Forward transform: $CWT_f(a,\tau)=\frac{1}{\sqrt{a}}\int_{-\infty}^{+\infty} f(t)\,\psi^*\left(\frac{t-\tau}{a}\right)dt$ for $f(t)\in L^2(\mathbb{R})$

The CWT measures the similarity between the signal and the scaled and shifted version of the elementary basis function $\psi_{a,\tau}(t)=\frac{1}{\sqrt{a}}$

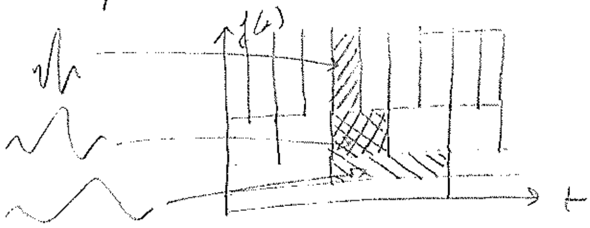
Inverse transform: if the wavelet $\psi(t)$ satisfies the admissibility condition with $\Psi(\omega)$ the Fourier transform of $\psi(t)$,

$CWT_f(a,\tau)=\langle f(t),\psi_{a,\tau}(t)\rangle$

$$C_\psi=\int_{-\infty}^{+\infty}\frac{|\Psi(\omega)|^2}{|\omega|}\,d\omega<\infty$$ (imposes that $|\Psi(\omega)|^2$ goes down faster than ω, not too demanding)

$$f(t)=\frac{1}{C_\psi}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} CWT_f(a,\tau)\,\psi_{a,\tau}(t)\,\frac{da\,d\tau}{a^2}$$

Any $f(t)\in L^2(\mathbb{R})$ can be written as a superposition of shifted and dilated wavelets. The reconstruction is in the L^2 sense (the L^2 norm of the reconstruction error is 0). The discretization of time-frequency space uses large time steps for large a ($a>1$) as in this case, $\psi_{a,\tau}(t)$ is "long" and of low frequency and fine time steps for small a ($a<1$) as in this case $\psi_{a,\tau}(t)$ is "short" and of high frequency. This defines the tiling of the frequency-time plane (a,τ) in the form $(a_0^m;\ a_0^m\cdot\tau_0)$



Properties

① Linearity: follows from the linearity of the inner product

② Shift property: if $g(t)=f(t-\tau')\Rightarrow CWT_g(a,\tau)=CWT_f(a,\tau-\tau')$ $\quad t'=t-\tau'$

Proof: $CWT_g(a,\tau)=\frac{1}{\sqrt{a}}\int_{-\infty}^{+\infty} f(t-\tau')\,\psi^*\left(\frac{t-\tau}{a}\right)dt=\frac{1}{\sqrt{a}}\int_{-\infty}^{+\infty} f(t')\,\psi^*\left(\frac{t'+\tau'-\tau}{a}\right)dt'$

$u=t-\tau'$, $du=dt$

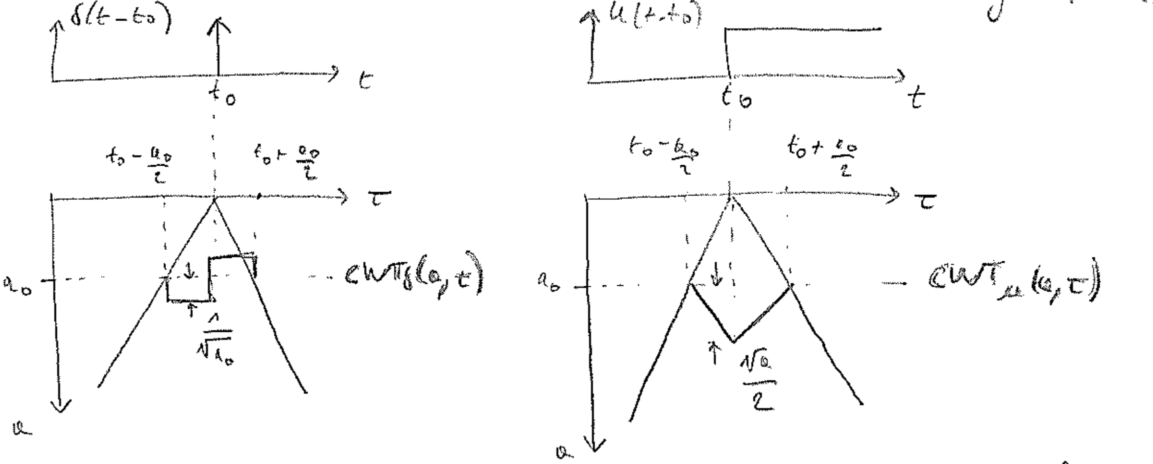
$=\frac{1}{\sqrt{a}}\int_{-\infty}^{+\infty} f(t')\,\psi^*\left(\frac{t'-(\tau-\tau')}{a}\right)dt'$, comme $u=t-\tau'\Rightarrow =: CWT_f(a,\tau-\tau')$

③ Scaling property: if $g(t)=\frac{1}{\sqrt{s}}f\left(\frac{t}{s}\right)\Rightarrow CWT_g(a,\tau)=CWT_f\left(\frac{a}{s},\frac{\tau}{s}\right)$ $\quad \to t'=\frac{t}{s}$, $dt'=dt/s$

Proof: $CWT_g(a,\tau)=\frac{1}{\sqrt{a}}\int_{-\infty}^{+\infty}\frac{1}{\sqrt{s}}f\left(\frac{t}{s}\right)\psi^*\left(\frac{t-\tau}{a}\right)dt=\frac{1}{\sqrt{a}}\int_{-\infty}^{+\infty}\frac{1}{\sqrt{s}}f(t')\,\psi^*\left(\frac{st'-\tau}{a}\right)s\,dt'$

$=\frac{1}{\sqrt{a}}\int\frac{1}{\sqrt{s}}f(t')\,\psi^*\left(\frac{t'-\frac{\tau}{s}}{a/s}\right)s\,dt'=\sqrt{\frac{s}{a}}\int f(t')\,\psi^*\left(\frac{t'-\frac{\tau}{s}}{a/s}\right)dt'=CWT_f\left(\frac{a}{s},\frac{\tau}{s}\right)$

) Time localization: Ex: the CWT of a Dirac pulse at time t_0 is: $CWT_\delta(a,\tau) = \frac{1}{\sqrt{a}}\psi\left(\frac{t_0-\tau}{a}\right)$
the wavelet transform is equal to the scaled wavelet reversed in time and centered at the location of the Dirac



uno phase Haar wavelet

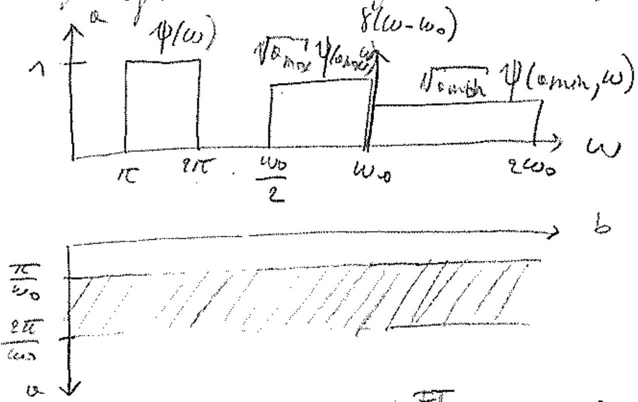
The transform "zooms-in" to the Dirac or onto the step function with a very good localization for very small scales a

Proof: the basic Haar wavelet transform is $\psi(t) = \begin{cases} 1 & 0 \le t \le \frac{1}{2} \\ -1 & \frac{1}{2} \le t < 1 \\ 0 & \text{else} \end{cases}$

Hence it gives a triangle since $u(t)$ as a sum of rectangle (rate) gives a triangle (rate) the line is first ↗ because of positive part of Haar wavelet convoluted with u then decreasing because convolution of negative part of Haar wavelet.

+ démo sur page agrafée → doc 1558816731771317 pg 4

) Frequency localization: Ex: A complex sinusoid of unit magnitude at frequency ω_0. The "sinc wavelet" (a perfect bandpass filter) with magnitude 1 for $\omega \in [\pi, 2\pi]$



The highest frequency wavelet that can analyse the signal has a scale $a_{min} = \frac{\pi}{\omega_0}$ and the lowest frequency wavelet that can analyze the signal has the scale $a_{max} = \frac{2\pi}{\omega_0}$ because:

$$\psi(t) \xrightarrow{FT} \Psi(\omega) \Rightarrow \begin{cases} \psi\left(\frac{t}{a}\right) \xrightarrow{FT} a\,\Psi(a\omega) & \text{scaling} \\ \frac{1}{\sqrt{a}}\psi\left(\frac{t}{a}\right) \xrightarrow{FT} \sqrt{a}\,\Psi(a\omega) & \text{linearity} \end{cases}$$

Thus: $a\omega_0 = \omega$ and $\omega = 2\pi \Rightarrow a = \frac{2\pi}{\omega_0}$ hence the lowest freq. wavelet which can still analyze the signal is $a = a_{max}$

$a\omega_0 = \omega$ and $\omega = \pi \Rightarrow a = \frac{\pi}{\omega_0}$ hence the highest freq. wavelet which can still analyze the signal is $a = a_{min}$

Proof: démo sur page agrafée → idem ci-dessus

The frames of the WT and the STFT are mean to discretize these continuous-time transforms to represent the signal as set of transform coefficient and reconstruct the signal in a stable way:

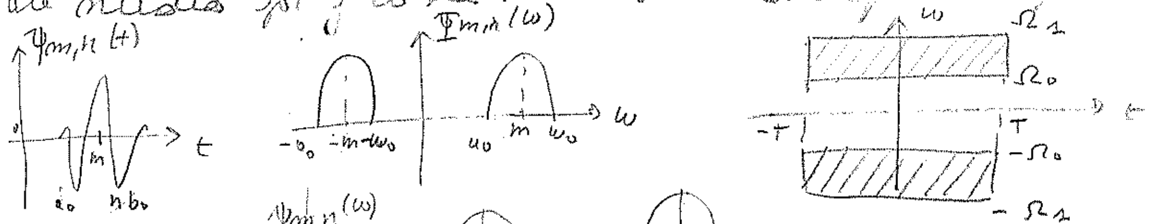
$g_{\omega,\tau} \xrightarrow{\text{discretize } \omega,\tau} g_{m,n}$ such that $f = \sum_m \sum_n \langle g_{m,n}, f \rangle \tilde{g}_{m,n}$

$\psi_{a,\tau} \xrightarrow{\text{discretize } a,\tau} \psi_{m,n}$ such that $f = \sum_m \sum_n \langle \psi_{m,n}, f \rangle \tilde{\psi}_{m,n}$

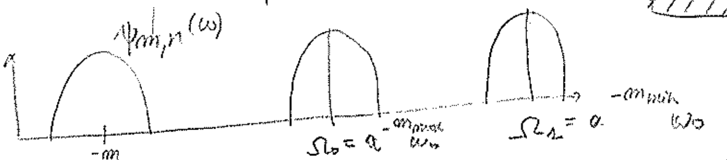
Demo & results done slides 56 – 58, only to be known for additional questions at exam → to print + add comments.

· CWT

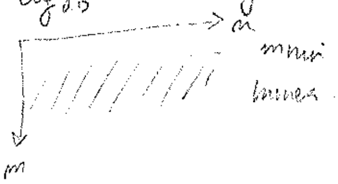
Assume that $|\psi|$ and $|\Psi|$ are symmetric and centered about $t=0$ and $\omega=\omega_0$ respectively. Then $\psi_{m,n}$ is centered around $t = a_0^m n b_0$ and $\Psi_{m,n}$ is centered around $f = \pm a_0^{-m}\omega_0$. $\langle \psi_{m,n}, f\rangle$ is the "information content" of $f(t)$ near $t = a_0^m n b_0$ and $\omega = \pm a_0^{-m}\omega_0$. Assume $f(t)$ is localized in time and frequency, then only the coefficients $\langle \psi_{m,n}, f\rangle$ for which $(t,\omega) = (a_0^m n b_0, a_0^{-m}\omega_0)$ lies within (or very close to) $[-T,T]\times([-\Omega_1,-\Omega_0]\cup[\Omega_0,\Omega_1])$ are needed for f to be reconstructed up to a good approximation



Proof:



$\log_{a_0}\Omega_0 = \log_{a_0} a^{-m_{max}} + \log_{a_0}\omega_0 = -m_{max} + \log_{a_0}\omega_0 \Rightarrow \begin{cases} m_{max} = \log_{a_0}\left(\frac{\omega_0}{\Omega_0}\right) \\ m_{min} = \log_{a_0}\left(\frac{\omega_0}{\Omega_1}\right) \end{cases}$



→ comme m limité → $a^{-m}\omega_0$ est limité → $\psi(\omega)$ est limité

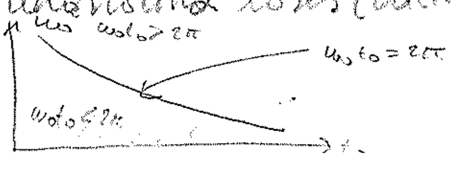
· STFT

If $g_{m,n}(t)$ constitutes a frame $L^2(\mathbb{R})$ with frame bounds A and B, then $A \le \frac{2\pi}{\omega_0 t_0}\|g\|^2 \le B$ (ψ is missing in this expression → no admissibility conditions as for bandlimited frames ???). Tight frames (for $\|g\|=1$ as g can always be normalized) are defined as $A = B = \frac{2\pi}{\omega_0 t_0}$. ω_0 and t_0 cannot be arbitrarily chosen:

→ no frames exists for $\omega_0 t_0 > 2\pi$??

→ good tight frames for $\omega_0 t_0 < 2\pi$ with good time-freq. localization properties but oversampling, with $A = B = \frac{2\pi}{\omega_0 t_0}$ the oversampling ratio. ??

→ orthonormal bases (critical sampling) for $A = B = 1 \Rightarrow \omega_0 t_0 = 2\pi$ bad localized in time-freq.



here is no way to construct STFT orthonormal basis with good time-frequency localization. This comes from the Balian-Low theorem:

If $g_{m,n}(t) = e^{j2\pi mt} w(t-n)$ $m,n \in \mathbb{Z}$ constitutes a frame for $L^2(\mathbb{R})$ then either $\int t^2 |w(t)|^2 dt = \infty$ either $\int \omega^2 |W(\omega)|^2 d\omega = \infty$ (settings $\omega_0 = 2\pi$ $t_0 = 1 \Rightarrow \omega_0 t_0 = cst$)

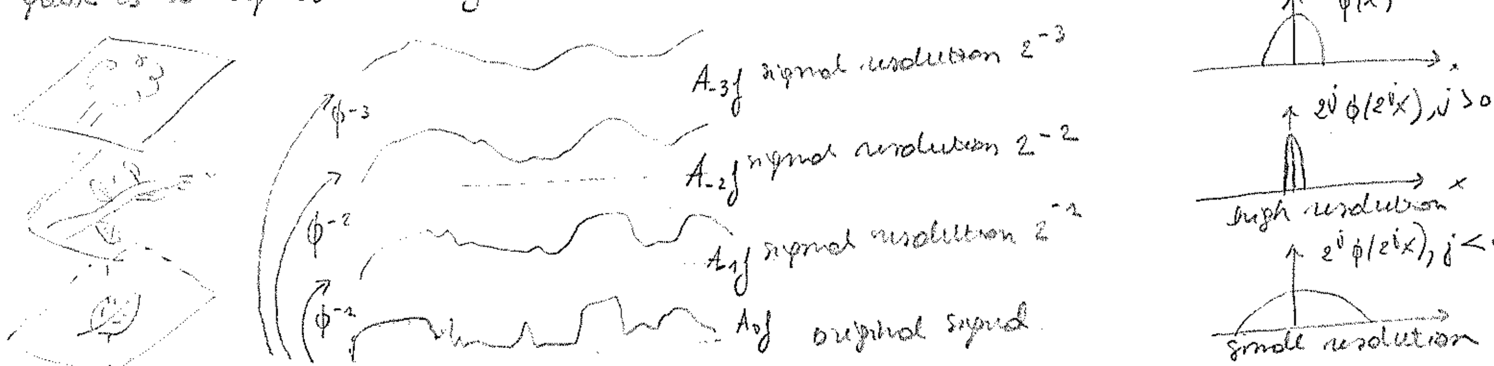
Ex: $g_{m,n}(t) = e^{jm\omega_0 t} w(t - nt_0)$ with gaussian window $w(t) = \pi^{-1/4} e^{-t^2/2}$

$\tilde{g}_{m,n}(t) = e^{jm\omega_0 t} \tilde{w}(t - nt_0)$ the dual frame and the window $\tilde{w}(t)$ can be calculated with an iterative procedure for different choices of $\omega_0 = t_0 = (2\pi\lambda)^{1/2}$. $\tilde{w}$ becomes less and less smooth as λ increases (oversampling decreases). For $\lambda < 1$, these frames have good time-freq localization. For $\lambda = 1$, $\tilde{w}$ is not even square-integrable anymore. Thus there is no STFT orthonormal basis with good time-localization properties (orthogonal bases are necessarily critically sampled). If modulation by cosines is used instead of modulation by complex exponentials, good orthonormal bases do exist $\rightarrow$ local cosine bases.

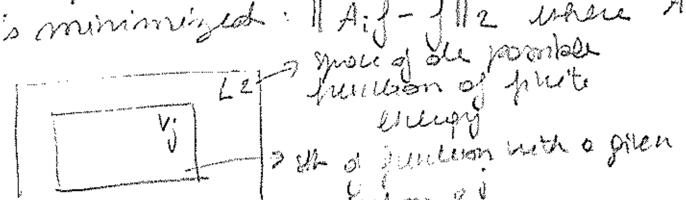
MULTIRESOLUTION REPRESENTATION

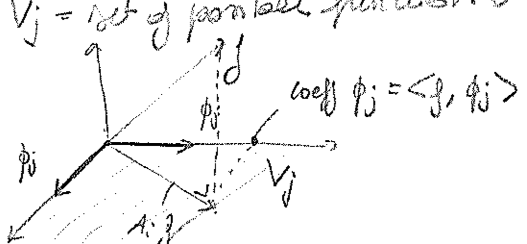
Scaling functions

The goal is to represent any arbitrary image as a set of recursive approximations.



The scaling functions $\phi_j(x)$, $j<0$ define the signal resolution and are all derived from one scaling function $\phi(x)$, forming a set of basis: $\phi_j(x) = 2^j \phi(2^j x)$, $j<0$.
The goal is to make an approximation of the signal such that the L^2 norm error is minimized: $\| A_j f - f \|_2$ where $A_j f \in V_j$ = set of possible functions at a given resolution 2^j





$A_j f(x) = 2^{-j} \sum_{n=-\infty}^{+\infty} \langle f(u), \phi_j(u - 2^{-j}n) \rangle . \phi_j(x - 2^{-j}n)$ $\rightarrow$ $A_j f$ is the projection of f onto the basis ϕ_j, $\langle f, g \rangle$ is the inner product:

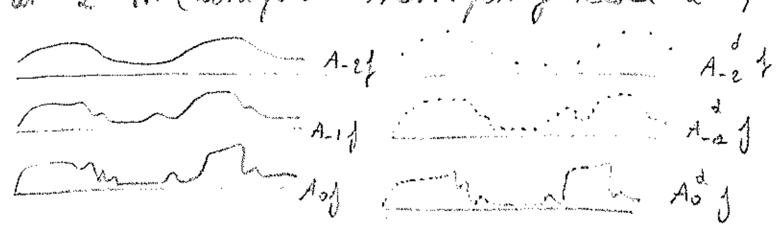
$s = \sum_{n=-\infty}^{+\infty} \text{coeff}(n)\, b(n)$ $\qquad \langle f, g \rangle = \int_{-\infty}^{+\infty} f(u), g(u)\, du$.

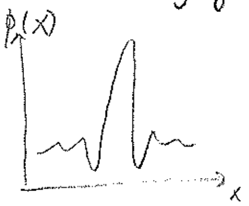
$\underline{s} = \underline{\text{coeff}}\ \underline{b}$

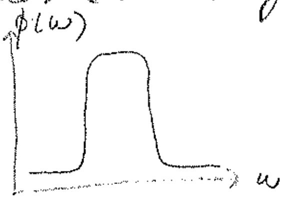
A scaling function is a low pass filter. The set of inner products $A_j^d f(n) = \langle f(u), \phi_j(u - 2^{-j}n) \rangle_{n \in \mathbb{Z}}$ are called the discrete approximation of $f(x)$ at the resolution 2^j. They can be rewritten as $A_j^d f(n) = \langle f(u), \phi_j(u - 2^{-j}n) \rangle = \int_{-\infty}^{+\infty} f(u)\, \phi_j(u - 2^{-j}n)\, du$

$= (f(u) * \phi_j(-u)) (2^{-j}n)_{n \in \mathbb{Z}}$ which shows that to obtain the discrete approximations
$1 \le m \le N$

of $A_j f$, f is convolved with a low pass filter and the result is evaluated at $2^{-j}n$ (uniform sampling rate 2^j) → $2^j N$ samples (downsampling). In practice $A_j^d f$ are used instead of A_j







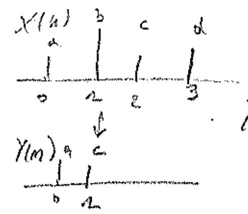
The discrete approximation allows to compute the information in discrete coefficients. The fast algorithm for calculating $A_j^d f$ is called the pyramidal algorithm and allows to calculate a given resolution from the previous one instead of starting every time from the initial signal. In this algorithm:

$A_j^d f$ is computed by convolving $A_{j+1}^d f$ with H and downsampling by 2 (H = low pass filter)

$$A_j^d f(n) = \sum_{k=-\infty}^{+\infty} h(\underbrace{2}_{\text{downsampling}}n-k)\, A_{j+1}^d f(k) \;,\; j<0 \quad \to \text{convolution}$$

$A_0^d f \to \boxed{H} \to (\downarrow 2) \to A_{-1}^d f \to \boxed{H} \to (\downarrow 2) \to A_{-2}^d f$

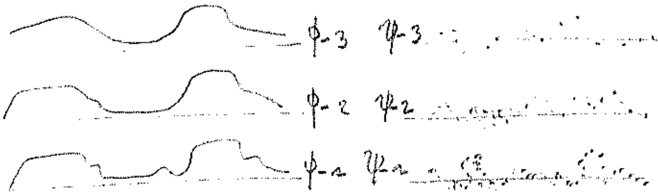
where $(\downarrow 2)$ means: retaining half of the results.

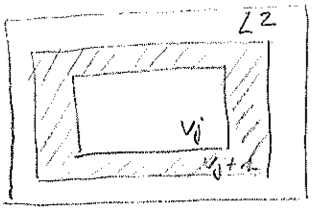


$Y(n) = X(2n)$

Wavelet function

To "come back" from a given resolution to the initial signal (multiresolution synthesis) it is necessary to store the lost details for each multiresolution analysis level. This is done through wavelet functions



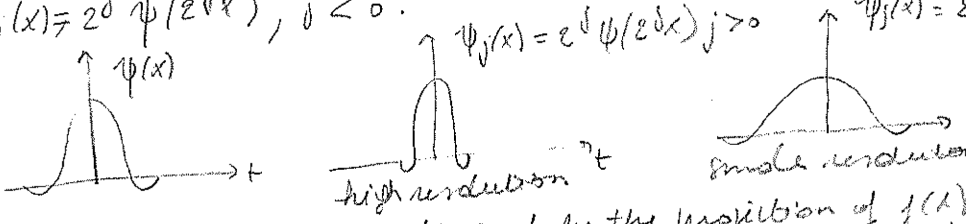


$\underset{\substack{\uparrow\\ \text{orthogonal complement of}\\ V_j \text{ and } V_{j+1}\\ \hookrightarrow \psi_j}}{O_j} \underset{}{\overset{\text{sum of vectorial spaces}}{\oplus}} \underset{\hookrightarrow \phi_j}{V_j} = \underset{\hookrightarrow \phi_{j+1}}{V_{j+1}}$

original signal

The wavelet functions $\psi_j(x)$, $j<0$ reconstruct the dropped information by applying g, and are derived from one wavelet function $\psi(x)$, forming a set of bases:

$\psi_j(x) = 2^j\, \psi(2^j x)$, $j<0$.



The discrete details are obtained by the projection of $f(x)$ on $\psi_j(x)$, $j<0$:

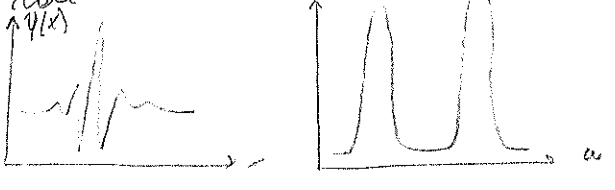
$$P_j f(x) = 2^{-j} \sum_{n=-\infty}^{+\infty} \langle f(u), \psi_j(u-2^{-j}n)\rangle\, \psi_j(x-2^{-j}n) \quad // \; s = \sum_{n=-\infty}^{+\infty} \text{coeff}(n)\, b(n) \Leftrightarrow s = \text{coeff}\; b$$

on inner products.

The wavelet forms a band pass filter and the set of inner products can be rewritten by def. of inner product as:

$D_j f(n) = \langle f(u), \psi_j(u-2^{-j}n)\rangle_{n\in\mathbb{Z}}$ which is called discrete details of $f(x)$ at resolution 2^j

$D_j f(n) = (f(u) * \psi_j(-u))(2^{-j}n)_{n\in\mathbb{Z}}$ and shows that it consists in a band pass filtering of $f(x)$ sampled uniformly at a rate 2^j

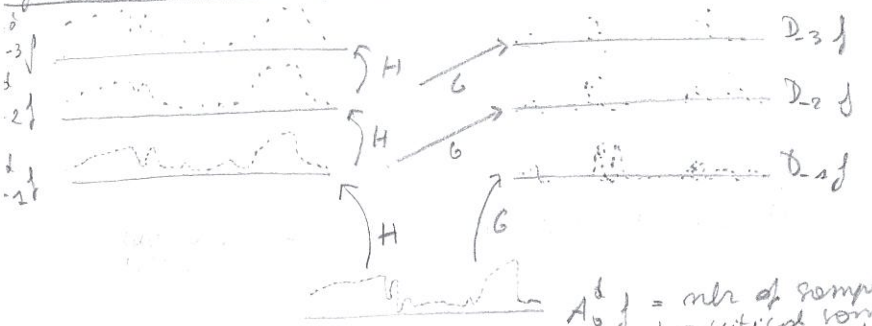


The fast algorithm to compute the $D_j f$ is also called the pyramidal algorithm and allows to define $D_j f$ from the corresponding $A^d_{j+1} f$. $D_j f$ is computed by convolving $A^d_{j+1} f$ with G followed by a downsampling by 2 (G = band pass filter)

$$D_j f(n) = \sum_{k=-\infty}^{+\infty} g(2n-k) \, A^d_{j+1} f(k), \; j \le 0$$

$$A^d_0 f \longrightarrow \boxed{G} \longrightarrow \boxed{\downarrow 2} \longrightarrow D_{-1} f$$

Dyadic Wavelet Representation



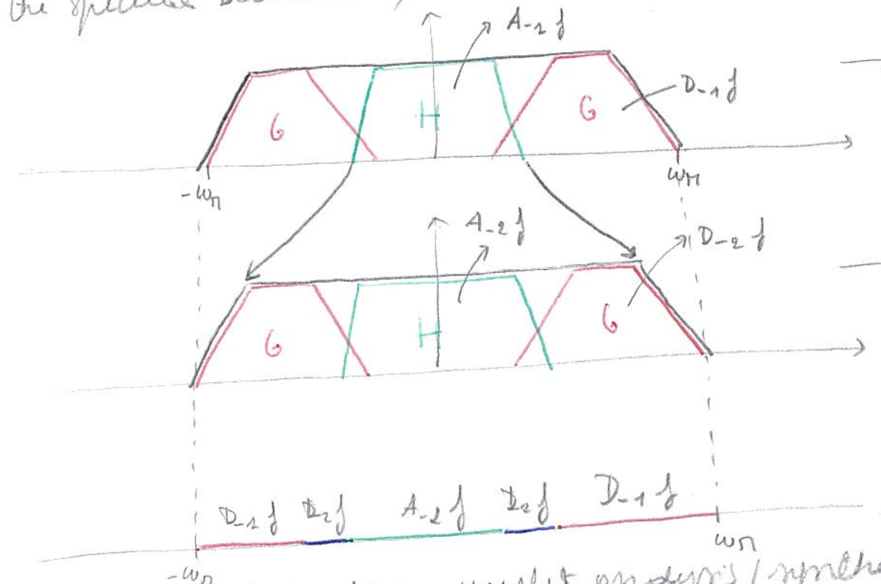
• from $A^d_0 f \longrightarrow A^d_{-3} f, D_{-3} f, D_{-2} f, D_{-1} f$
↳ analysis

• from $A^d_{-3} f, D_{-3} f, D_{-2} f, D_{-1} f \longrightarrow A^d_0 f$
↳ synthesis

$A^d_0 f$ = nbr of sample in $A^d_{-1} f$ $(= \frac{N}{2})$ + nbr of sample in $D_{-1} f$ $(= \frac{N}{2})$
↳ critical sampling

To fully reconstruct a signal, it is necessary to keep only the coarse representation of it ($A^d_{-3} f$) and all the lost details $D_j f$

In the spectral domain, this decomposition gives:



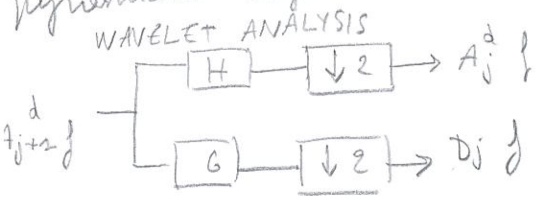
→ original spectrum: overlapping between H and G as non overlapping does not exist in reality (corresponds to a sinc)

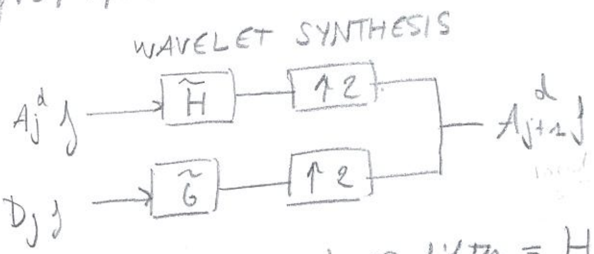
→ frequency band doubled as downsampling operation shrinks the time thus spread the frequency domain

PROOF: $F_y = \frac{F_x}{2} \rightarrow \omega_x = \frac{2\pi F}{F_x}$

$\omega_y = \frac{2\pi F}{F_y} = 2 \cdot \frac{2\pi F}{F_x} = 2\,\omega_x$

The algorithms to perform wavelet analysis / synthesis are a combination of the 2 pyramidal algorithms:





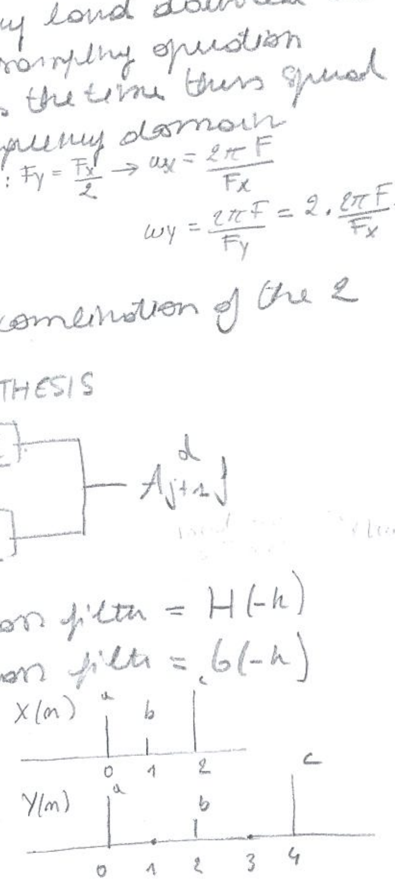
H - analysis low pass filter
G - analysis band pass filter

$\phi(2\omega) = H(\omega)\,\phi(\omega)$

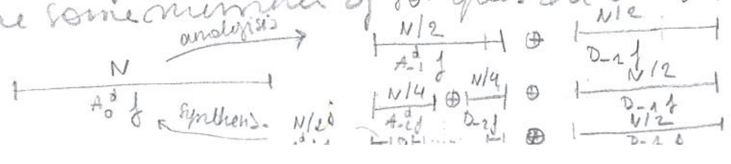
$\psi(2\omega) = G(\omega)\,\phi(\omega)$

$\tilde{H}$ - synthesis low pass filter = H(-n)
$\tilde{G}$ - synthesis band pass filter = G(-n)

upsampling means



The equality $\tilde{H}(n) = H(-n)$ holds only for orthonormal transform (critical sampling). This means that there are the same number of samples in the transformed domain as in the spatial domain:



12

5.4. Applications

Embedded Zerotree Coding of Wavelet Coefficients (EZW)

Organize the coefficients with similar locations in corresponding subbands in trees growing exponentially across the scales

Apply SAQ.

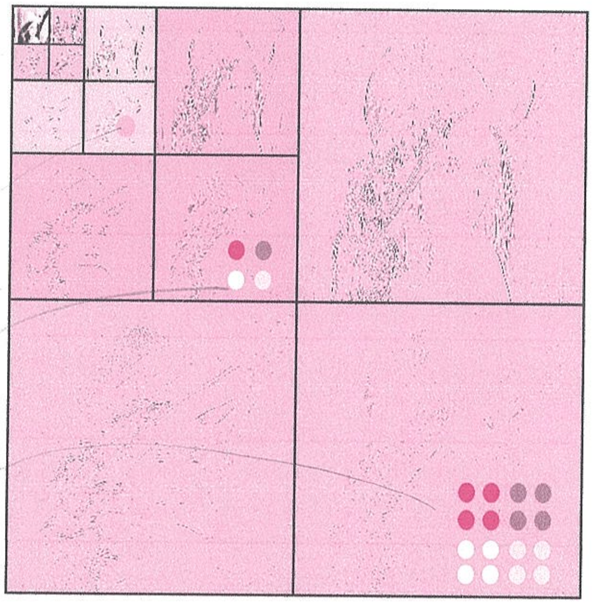
Zerotree hypothesis: if a coefficient is not significant with respect to a given threshold *T*, then all of its descendants are not significant either with respect to *T*.

For every *T* encode the resulting significance map.

if zero here then

zero here than

zero here



5.4. Applications

Zerotree Representation

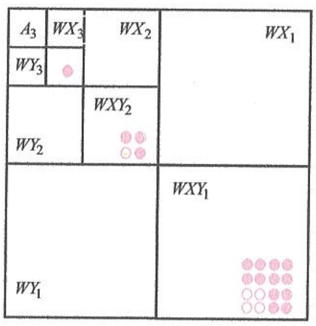
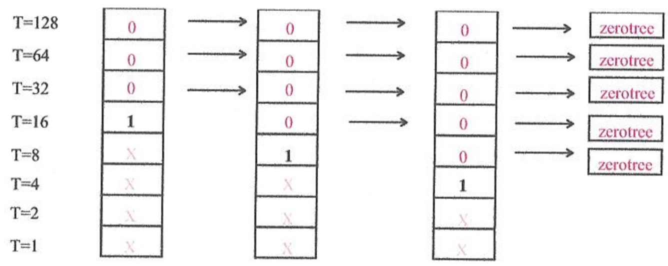


Illustration of the parent-child dependencies



Zerotree coding

pue reprezinta ultima coloana ???

- Model: the *zerotree hypothesis* is satisfied with a high probability
- The EZW coder exploits inter-band correlations.

5.4. Applications

Encoding Algorithm

Dominant Pass

- The subbands are scanned in Z-order.
- Raster scanning within the subbands.

$\Rightarrow$ *dominant list*

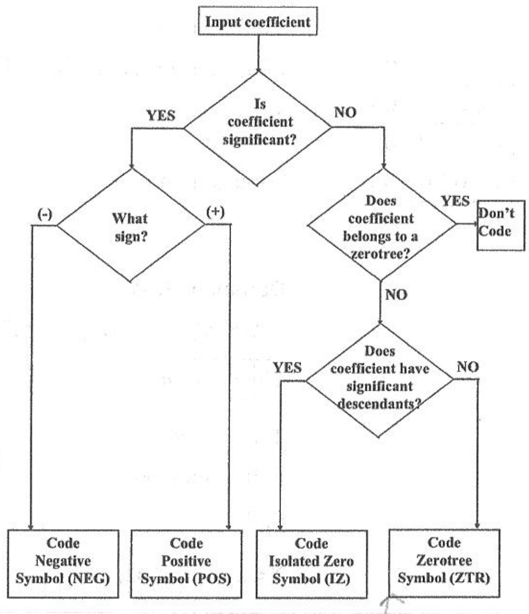
Subordinate Pass.

- Applied for the coefficients that have been coded previously as significant.
- Current threshold 2^T
- For every coefficient coded as POS or NEG send to the decoder the information contained in the bit-plane *T*.

$\Rightarrow$ *subordinate list*

- Entropy coding of both lists.

Dominant Pass



comes from this

5.4. Applications

Encoding Algorithm (example)

10	8	1	2
2	4	-1	-2
3	4	0	0
1	2	1	1

T = 8 → values of pixel $\geq$ T=8

1	1	0	0
0	0	0	0
0	0	0	0
0	0	0	0

Dominant Pass

POS, POS, ZTR, ZTR, ZTR, ZTR, ZTR, ZTR

10	8

Subordinate Pass

-

→ significant values replaced with 0

0	0	1	2
2	4	-1	-2
3	4	0	0
1	2	1	1

→ restart procedure with lower treshold

T = 4

0	0	1	2
2	4	-1	-2
3	4	0	0
1	2	1	1

0	0	0	0
0	1	0	0
0	1	0	0
0	0	0	0

Dominant Pass → nothing encoded for 2 first values.

-, ZTR, IZ, POS, ZTR, POS, ZTR, ZTR, ZTR, ZTR, ZTR, ZTR

10	8	4	4

T=8 T=4

Subordinate Pass

0, 0

	10	8
8	1	1
4	0	0
2	1	0
1	0	0

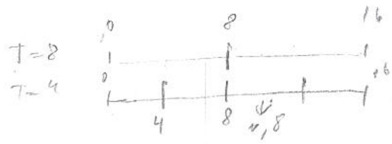
→ encoding at treshold 8

→ encoding at tresholds 4

→ superdondescos

T = 2, ... T = 1, ... → next steps

- halves the quantizer cells
- if magnitude $\geq$ upper half of old cell → 1
  < → 0



→ 10 & 8 encodés comme 1

→ 10 & 8 encodés comme 0 car pas plus grand que la moitié de l'ancienne intervalle

5.4. Applications

Decoding Algorithm (example)

→ from POS and T=8, we know that the nb to be reconstructed is between 8 and 16 (16 is the next threshold value (2^4)) and is positive. The reconstructed chosen value is in the middle of 8 to 16 → 12

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

Dominant Pass

POS, POS, ZTR, ZTR, ZTR, ZTR, ZTR, ZTR

10	8

Subordinate Pass

-

→ at next step, we refine, we know we have a

T = 8

12	12	0	0
0	0	0	0
0	0	0	0
0	0	0	0

T=8 1 so

T=4 0

we know we are in the lower half of previous value.

Dominant Pass

-, ZTR, IZ, POS| ZTR, POS, ZTR, ZTR, ZTR, ZTR, ZTR, ZTR

10	8	4	4

T=8 T=4

Subordinate Pass

0, 0

	10	8
8	1	1
4	**0**	**0**
2	1	0
1	0	0

T = 4

10	10	0	0
0	6	0	0
0	6	0	0
0	0	0	0

8 10 12 16

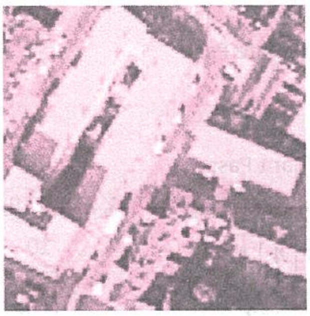
we know we have 10, bc btw 8 < 12 (middle of lower half of previous value)

T = 2, ... T =1, ...

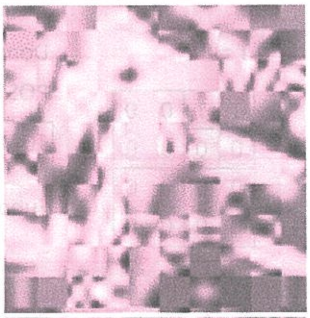
5.4. Applications

Example

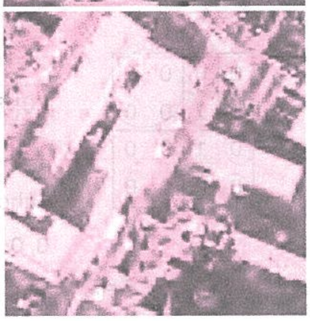
Original image



Zoomed area



Zoomed area in the image compressed with DCT (JPEG)

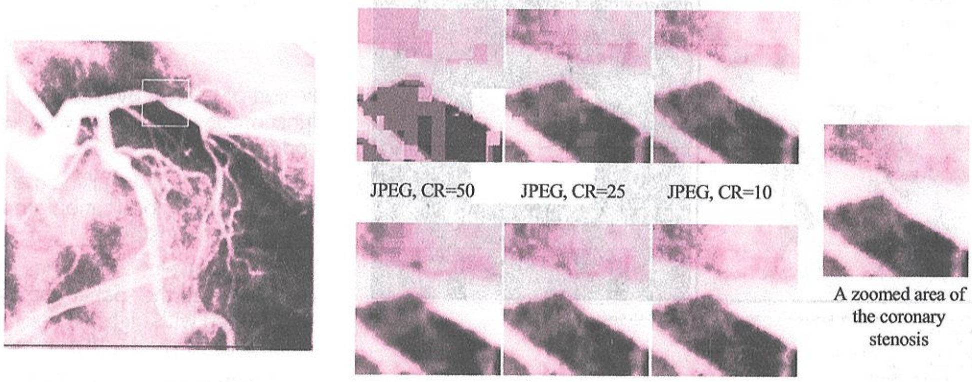


Zoomed area in the image compressed with wavelets

12 C

5.4. Applications

Quality Scalability

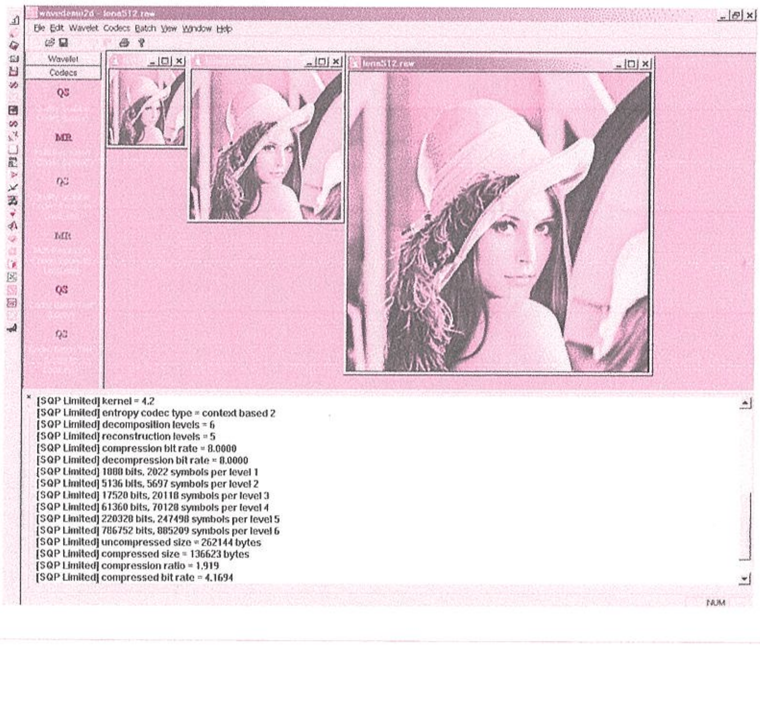


A zoomed area of the coronary stenosis

The image on the left is compressed at 50:1, 25:1, and 10:1 with JPEG and SQP. The zoomed area of the coronary stenosis indicates by the white rectangle is progressively refined up to the lossless version depicted on the right.

Note: the wavelet codec progressively refines the decoded image as more information is received. This illustrates the quality scalability property of wavelet-based image coding.

5.4. Applications

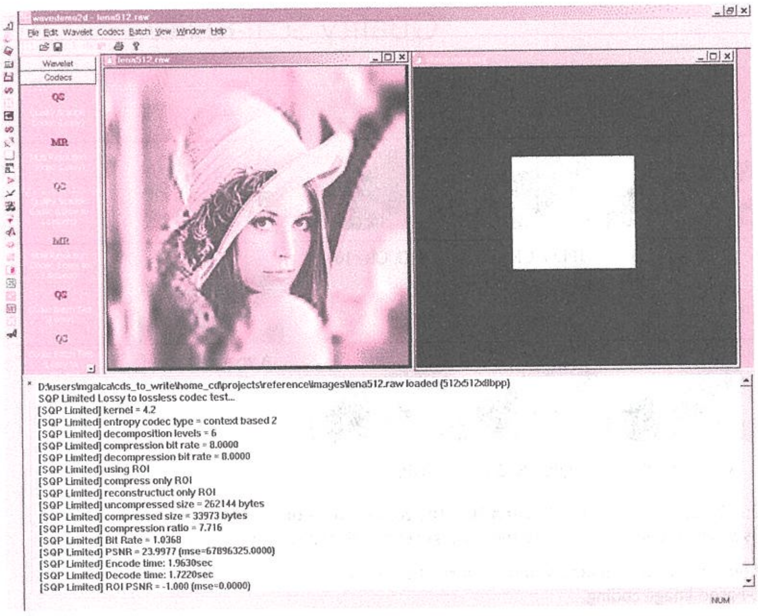


Resolution Scalability

Resolution Scalability provided by wavelet-based coding

- this is obtained by progressively transmitting the wavelet subbands from the lowest to the highest resolution level

5.4. Applications



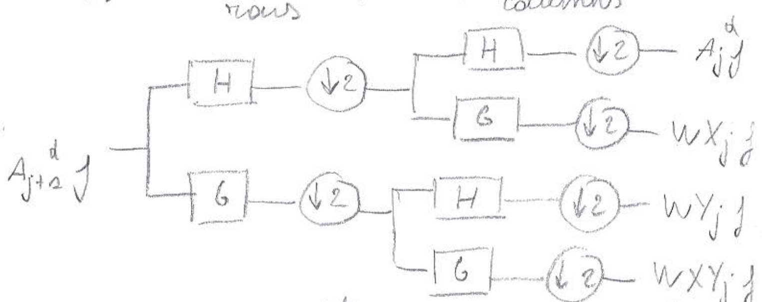
Region-of-Interest (RoI) coding

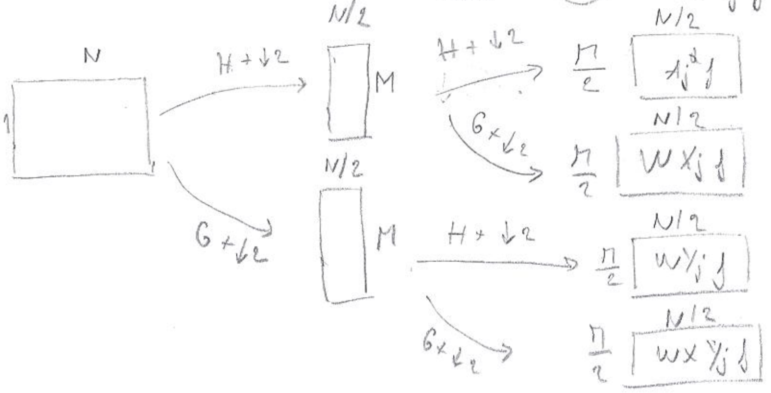
Region-of-Interest (RoI) coding functionality provided by wavelet-based coding systems

- this is obtained by making use of the time-frequency localization properties of the wavelet transform
- Only the wavelet coefficients that correspond to the spatial-domain ROI are transmitted with high accuracy (lossless reconstruction of the ROI in this example)
- A low quality version of the background is being transmitted as well
- This figure illustrates a rectangular ROI delineated by the user

on 2D, as the transform is considered to be separable, it is possible to build the 2D algorithm from the combination of 1D wavelets.

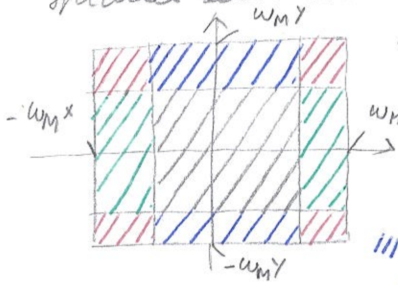
2D WAVELET ANALYSIS





Example slide 87 (à voir): the grey squares are the lost (?) details. There are 4 levels of resolutions (4x des carrés composés à l'intérieur de 4 carrés). The most noisy square corresponds to the edges of the image. The wavelet compacts energy in some few coefficients of high amplitude.

Spectral domain:



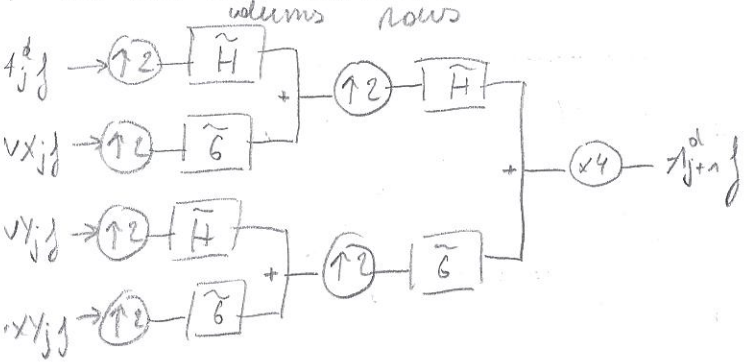
/// low pass filtering on rows & colums

/// band pass filtering on columns and low pass filtering on rows

/// band pass filtering on rows and low pass filtering on columns

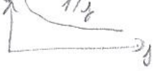
/// band pass filtering on rows & columns

2D WAVELET SYNTHESIS



Applications

Embedded Zerotree Coding of Wavelet Coefficients (EZW)

This technique deals with the efficient coding of the significance maps. These maps are sets of binary maps indicating the position of significant coefficients. They indicate the values above a treshold fixed by the embedded quantizer and their location. To do that in a fast way, zerotree hypothesis is used. It states that if there are zeros in a certain zone of the image details at a given resolution, there is a high likelihood that the corresponding coefficient in the next resolution level will be zeros too, forming a zerotree. This hypothesis comes from the 1/f tendency  stating that there is a higher chance to have important coefficient a low freq  than at high frequency

The input picture is first analyzed through wavelet then quantized with SAQ.

↳ important slide 90/98 pour examen.

- **Quality Scalability**

  cf slides 96

- **Resolution Scalability**

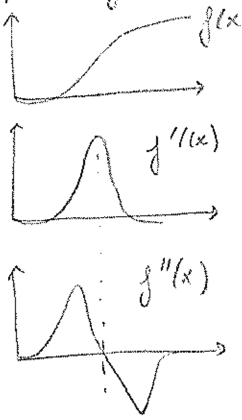
  cf slide 97

- **Region-of-Interest Scalability**

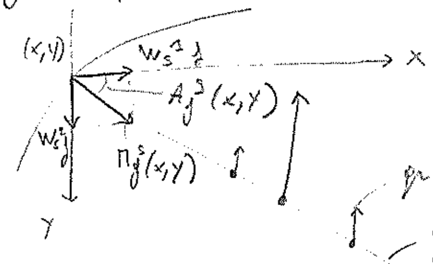
  cf slide 98

- **Multi-scale edge detection via CWT**

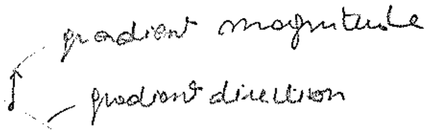
An edge is defined as an inflexion point on the function describing the image (i.e. f?). To find the inflexion point, the wavelet uses $f'(x)$ as it has a maximum at this point. In 2D, edges are thus defined as a local maximum of the gradient (2D $f'(x)$) magnitude in the gradient direction



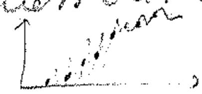
1D



2D



In every point (x,y), there are 2 wavelet information: the magnitude and the angle
$M_j^s(x,y)$ $A_j^s(x,y)$

In order to perfectly detect the edges, it is important to first suppress the noise of the image because otherwise, edges are found everywhere 

To delete it, the image is convolved with a smoothing filter: $\theta_s(x) = \frac{1}{s}\theta\left(\frac{x}{s}\right)$

The corresponding wavelet for edge detection is $\psi(x) = \frac{d}{dx}\left(\theta(x)\right)$, $\psi_s(x) = \frac{1}{s}\psi\left(\frac{x}{s}\right)$

In 2D, the smoothing function is: $\theta(x,y) = \theta(x).\theta(y)$

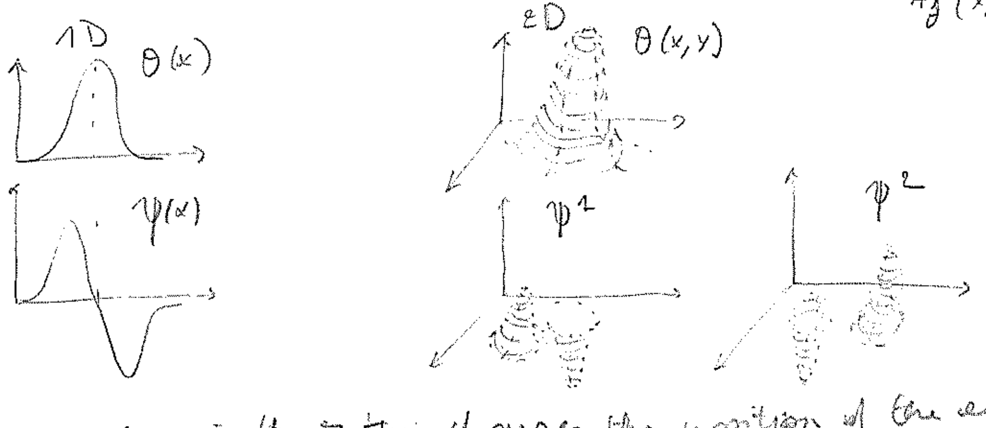
the wavelet functions are: $\psi^1(x,y) = \frac{d}{dx}\theta(x,y)$ $\psi^2(x,y) = \frac{d}{dy}\theta(x,y)$

the 2D wavelet transforms are: $W_s^1 f(x,y) = (f * \psi_s^1)(x,y) = s\frac{d}{dx}(f * \theta_s)(x,y)$

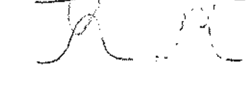
$$W_s^2 f(x,y) = (f * \psi_s^2)(x,y) = s\frac{d}{dy}(f * \theta_s)(x,y)$$

The modulus and angle of $\vec{\nabla}(f * \theta_s)$ at (x,y) is: $M_j^s(x,y) = \left(W_s^1 f(x,y)\right)^2 + \left(W_s^2 f(x,y)\right)^2$

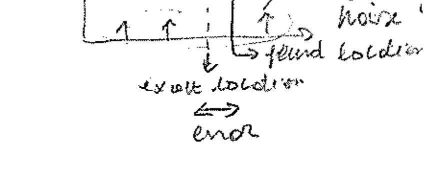
$$A_j^s(x,y) = \arg\left(W_s^1 f(x,y) + j W_s^2 f(x,y)\right)$$



The problem is that this changes the position of the edges. Bigger the support of the smoothing filter, better the deletion of the noise but worst is the error on edge location. Indeed if the signal is & the filter is

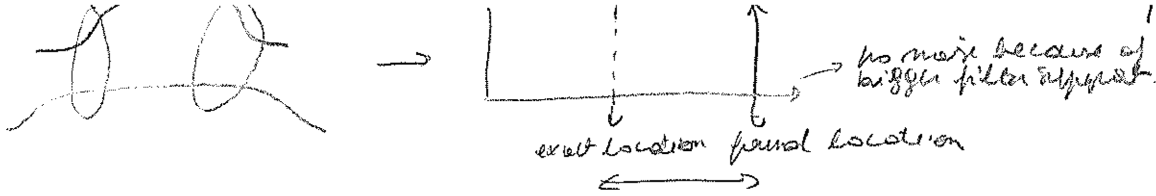


→



But if the signal is

& the filter is



The error is bigger if higher support because neighbouring edges overlap in convolution. The solution is to use multi-scale edge detection to refine the edge position. This technique uses different gradient orientation for edge detection and several scale for the smoothing filter support (cf slide 107 - 109)

WAVELET BASES & FILTER BANKS

Perfect reconstruction filter banks implement series expansions of discrete-time signals. There are 3 types of tools for analyzing filter banks: time domain analysis, modulation-domain analysis, polyphase-domain analysis.

Two channel Filter Banks - Modulation domain analysis

If the system is continuous $\longrightarrow$ Laplace transform to write CWT
If the system is discrete $\longrightarrow$ Z transform to write CWT

The discrete domain will be expressed in this case:

- convolution of discrete signals: $\mathcal{Z}\{x[n]*y[n]\} = X(z).Y(z)$

PROOF: $X(z).Y(z) = \left(\sum_{n=0}^{\infty} x[n] z^{-n}\right)\left(\sum_{m=0}^{\infty} y[m] z^{-m}\right) = \sum_{n=0}^{\infty}\sum_{m=0}^{\infty} x[n] y[m] z^{-n-m}$

introducing $n+m=p$,

$$= \sum_{p=0}^{\infty}\sum_{m=0}^{\infty} x[p-m]\, y[m]\, z^{-p} = \sum_{p=0}^{\infty}\left(\sum_{m=0}^{\infty} x[p-m]\, y[m]\right) z^{-p} \quad (*)$$

let $s[p] = \sum_{m=0}^{\infty} x[p-m]\, y[m] \triangleq x[p] * y[p]$

from $(*)$, we deduce $X(z).Y(z) = S(z)$

hence $\mathcal{Z}\{x[n]*y[n]\} = X(z).Y(z)$

- downsampling of discrete signals: $y[n] = x[2n] \rightarrow \mathcal{Z}(y[n]) = Y(z) = \frac{1}{2}\left[X(z^{1/2}) + X(-z^{1/2})\right]$

PROOF: let $X_e(z) = \sum_{n=0}^{\infty} x[2n] z^{-n}$ and $X_o(z) = \sum_{n=0}^{\infty} x[2n+1] z^{-n}$

thus $Y(z) = \sum_{n=0}^{\infty} y[n] z^{-n} = \sum_{n=0}^{\infty} x[2n] z^{-n} = X_e(z)$

also $X(z) = \sum_{n=0}^{\infty} x[n] z^{-n} = \sum_{m=0}^{\infty} x[2m] z^{-2m} + \sum_{m=0}^{\infty} x[2m+1] z^{-2m-1}$

$$= X_e(z^2) + z^{-1} X_o(z^2) \quad (*)$$

replacing $z \rightarrow -z$

$$X(-z) = X_e(z^2) - z^{-1} X_o(z^2) \quad (**)$$

from (*) and (**) it results: $X_e(z^2) = \frac{1}{2}[X(z) + X(-z)] \Rightarrow X_e(z) = \frac{1}{2}[X(z^{1/2}) + X(-z^{1/2})]$

$$X_o(z^2) = \frac{z}{2}[X(z) - X(-z)] \Rightarrow X_o(z) = \frac{z^{1/2}}{2}\left[X(z^{1/2}) - X(-z^{-1/2})\right]$$

$X_e(z)$, $X_o(z)$ represents the Z transforms of the even and odd phases of the input signal $x[n]$

upsampling discrete signals: $y[2n] = x[n]$, $y[2n+1] = 0$ thus $\mathcal{Z}(y[n]) = Y(z) = X(z^2)$

PROOF. $Y(z) = \sum_{n=0}^{\infty} y[n] z^{-n} = \sum_{n=0}^{\infty} y[2n] z^{-2n} + \sum_{n=0}^{\infty} y[2n+1] z^{-2n-1} = \sum_{n=0}^{\infty} x[n] z^{-2n} + 0 = X(z^2)$

from these results we obtain that:

ANALYSIS: $Y_H(z) = \frac{1}{2}\left[H(z^{1/2}) X(z^{1/2}) + H(-z^{1/2}) . X(-z^{1/2})\right]$ $\Rightarrow \begin{pmatrix} Y_H(z) \\ Y_G(z) \end{pmatrix} = \frac{1}{2} \underbrace{\begin{pmatrix} H(z^{1/2}) & H(-z^{1/2}) \\ G(z^{1/2}) & G(-z^{1/2}) \end{pmatrix}}_{H_m(z^{1/2})} \underbrace{\begin{pmatrix} X(z^{1/2}) \\ X(-z^{1/2}) \end{pmatrix}}_{X_m(z^{1/2})}$

$Y_G(z) = \frac{1}{2}\left[G(z^{1/2}) . X(z^{1/2}) + G(-z^{1/2}) . X(-z^{1/2})\right]$ $\Leftrightarrow Y(z) = \frac{1}{2} H_m(z^{1/2}) X_m(z^{1/2})$

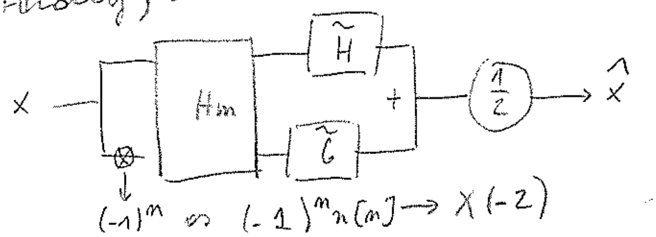
SYNTHESIS: $\hat{Y}_H(z) = \tilde{H}(z) . Y_H(z^2) = \frac{1}{2}\tilde{H}(z) . [H(z) . X(z) + H(-z) . X(-z)]$

$\hat{Y}_G(z) = \tilde{G}(z) . Y_G(z^2) = \frac{1}{2}\tilde{G}(z) . [G(z) . X(z) + G(-z) . X(-z)]$

$$\hat{X}(z) = \hat{Y}_H(z) + \hat{Y}_G(z) = \frac{1}{2}\begin{pmatrix} \tilde{H}(z) & \tilde{G}(z) \end{pmatrix} . \underbrace{\begin{pmatrix} H(z) & H(-z) \\ G(z) & G(-z) \end{pmatrix}}_{H_m(z)} \underbrace{\begin{pmatrix} X(z) \\ X(-z) \end{pmatrix}}_{X_m(z)} \qquad (I)$$

H_m is the analysis modulation matrix.

finally, the modulation-domain analysis of the two-channel filter bank gives



$(-1)^n$ or $(-1)^n x[n] \to X(-z)$ → Ajouter la décomposition pg 31 pour ... résultat ...

the perfect reconstruction condition is expressed as:

$\hat{X}(z) = X(z) \Rightarrow \begin{cases} \tilde{H}(z) . H(z) + \tilde{G}(z) . G(z) = 2 \; (a) \\ \tilde{H}(z) . H(-z) + \tilde{G}(z) . G(-z) = 0 \end{cases}$ from (I) $\xrightarrow[\text{matrix notation}]{}$ $\begin{pmatrix} \tilde{H}(z) & \tilde{G}(z) \end{pmatrix} . H_m(z) = \begin{pmatrix} 2 & 0 \end{pmatrix}$

By transposing this notation and multiplying by $(H_m^T(z))^{-1}$ we obtain.

(det vient de la formule $(H_m^T(z))^{-1} = \frac{(\quad)}{\det(H_m^T(z))}$)

$$\begin{pmatrix} \tilde{H}(z) \\ \tilde{G}(z) \end{pmatrix} = \frac{2}{\det(H_m(z))} \begin{pmatrix} G(-z) \\ -H(-z) \end{pmatrix}$$

defining $P(z) = \tilde{H}(z) H(z) = \frac{2}{\det(H_m(z))} H(z) . G(-z)$ → en trouvant $\tilde{H}(z)$ et $\tilde{G}(z)$ grâce à l'équation ci-dessus et en ... (z) ...

$\tilde{G}(z) . G(z) = \frac{-2}{\det(H_m(z))} H(-z) G(z) = P(-z)$ because $\det(H_m(z)) = -\det(H_m(-z))$

the perfect reconstruction condition is expressed as

$P(z) + P(-z) = 2$ ← équation de (a) ci-dessus et ... de $P(z)$ et $P(-z)$

↳ orthonormal wavelet expansions of discrete-time signals can be implemented using perfect reconstruction filter banks

HW: proof that $H(z) = \sum_{n=0}^{\infty} h[n] z^{-n} \to H(z) = \frac{\sqrt{2}}{2}(1 + z^{-1})$ with $h[n] = \frac{\sqrt{2}}{2}$

$\hookrightarrow \sum_{n=0}^{\infty} z^{-n} =$

4.3.1. Discretization of CWT

Basis function: $\psi_{a,b}(t)=\frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right)$

- Discretization of the scale parameter

$$a=a_0^{\,m},m\in Z,a_0\neq 1$$

- Discretization of the time shift

First idea: $For\ m=0, take\ b=nb_0$

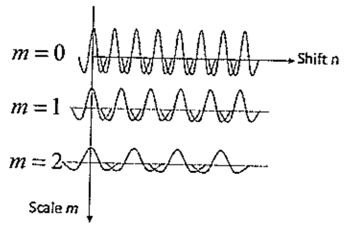
- Choose b_0 such that $\psi_{1,b}=\psi(t-nb_0)$ covers the whole time axis
- Moreover: choose b_0 such that for any m, $\psi_{a_0^{\,m},b}(t)$ covers the whole time axis.

- Note:

$$\Delta_t\left(\psi_{a_0^{\,m},0}(t)\right)=a_0^{\,m}\Delta_t(\psi_{1,0}(t))$$

- Result: $a=a_0^{\,m}\ and\ b=nb_0a_0^{\,m}$, $m,n\in Z,\ \ a_0>1,b_0>0$
- Discretized family of wavelets:

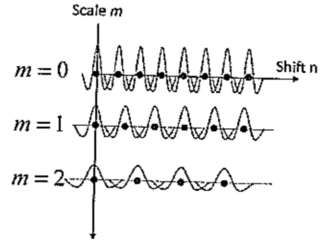
$$\psi_{m,n}(t)=a_0^{\,-m/2}\psi\left(a_0^{\,-m}t-nb_0\right)$$



The functions $\psi_{m,n}(t)$ for $a_0=2^{-1/2}$ and $b_0=1$

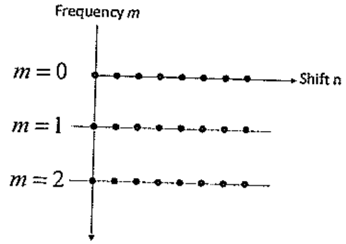
4.3.1. Discretization of CWT

Sampling grid for the CWT



The sampling grid for CWT
$a_0=2^{-1/2}$ and $b_0=1$

Sampling grid for the STFT



The sampling grid for STFT $t_0=1$

4.3.2. Discretization of STFT

Discretization of the frequency parameter

$$\omega=m\omega_0,\ \omega_0>0,m\in Z,\ \omega_0\ fixed$$

Basis function: $g_{\omega,\tau}(t)=e^{j\omega t}w(t-\tau)$

Discretization of the time shift

$$\tau=nt_0,\ t_0>0,\ n\in Z,t_0\ fixed$$

Result:

$$g_{\omega,\tau}(t)\rightarrow g_{m,n}(t)=e^{jm\omega_0 t}w(t-nt_0)$$

Note:

$$\Delta_t(g_{m,0}(t))=\Delta_t(g_{1,0}(t))$$
$$\Delta_\omega(g_{m,0}(t))=\Delta_\omega(g_{1,0}(t))$$

Different sampling grid for the discretization of the STFT if compared with the CWT

Let's come back to the problem:

Given $\langle g_{m,n},f\rangle$, *in which conditions* $f=\sum\sum\langle g_{m,n},f\rangle\tilde{g}_{m,n}$

Given $\langle \psi_{m,n},f\rangle$, *in which conditions* $f=\sum_m\sum_n\langle \psi_{m,n},f\rangle\tilde{\psi}_{m,n}$

4.3.3. Reconstruction in Frames

What is necessary to have a stable reconstruction?

1. $f(t)\xrightarrow{operator}\langle\psi_{m,n}(t),f(t)\rangle$

This operator should be bounded: $f(t)\in L^2(R)\Rightarrow\ \sum_{m,n}|\langle\psi_{m,n},f\rangle|^2$ *is finite*

It can be shown that $\exists B\in R^+$, such that $\sum_{m,n}|\langle\psi_{m,n},f\rangle|^2\leq B\|f\|^2$

2. Intuitively, if $\|f\|^2$ is small $\Rightarrow$ $\sum_{m,n}|\langle\psi_{m,n},f\rangle|^2$ is small too $\Rightarrow$

$\Rightarrow$ If $\sum_{m,n}|\langle\psi_{m,n},f\rangle|^2<1\Rightarrow\ \exists\alpha\in R$, such that $\|f\|^2\leq\alpha$

Define $\tilde{f}=\dfrac{f}{\sqrt{\sum_{m,n}|\langle\psi_{m,n},f\rangle|^2}}$ for an arbitrary f in $L^2(R)$

It can be shown that $\sum_{m,n}|\langle\psi_{m,n},\tilde{f}\rangle|^2\leq 1\Rightarrow\ \exists\alpha'\in R$, such that $\|\tilde{f}\|^2\leq\alpha'$

This implies: $\frac{1}{\alpha'}\|f\|^2\leq\sum_{m,n}|\langle\psi_{m,n},f\rangle|^2\Leftrightarrow A\cdot\|f\|^2\leq\sum_{m,n}|\langle\psi_{m,n},f\rangle|^2$

4.3.3. Reconstruction in Frames

- Reconstruction is possible if the family $(\psi_{m,n})_{m,n\in Z}$ *constitutes a frame.*

$$\exists A,B,\ 0<A\le B<\infty,\quad A\cdot\|f\|^2 \le \sum_{m,n}\left|\langle\psi_{m,n},f\rangle\right|^2 \le B\cdot\|f\|^2$$

- Note: take $f=f_1-f_2$. First inequality means that the distance $\|f_1-f_2\|$ cannot be arbitrarily large if is small: *stability* requirement. $\sum_{m,n}\left|\langle\psi_{m,n},f_1\rangle-\langle\psi_{m,n},f_2\rangle\right|^2$
- When $A=B$, the frame is *tight.*

Proposition

If $A=B=1$ and $\forall m,n\in Z,\ \|\psi_{m,n}\|=1$, the family $(\psi_{m,n})_{m,n\in Z}$ is an *orthonormal basis.*

∘ A = B gives the "redundancy ratio", or the oversampling ratio.

∘ A = B = 1 ⇒ critical sampling ⇔ orthonormal basis.

4.3.3. Reconstruction in Frames

A. **Reconstruction in tight frames**

$$\sum_{m,n}\left|\langle\psi_{m,n},f\rangle\right|^2 = A\|f\|^2 \Leftrightarrow \sum_{m,n}\langle f,\psi_{m,n}\rangle\cdot\langle\psi_{m,n},f\rangle = A\langle f,f\rangle$$

$$\sum_{m,n}\langle f,\psi_{m,n}\rangle\cdot\langle\psi_{m,n},f\rangle = \sum_{m,n}\langle f,\psi_{m,n}\rangle\cdot\int\psi_{m,n}^*(t)f(t)dt =$$

$$=\int\left(\sum_{m,n}\langle f,\psi_{m,n}\rangle^*\cdot\psi_{m,n}(t)\right)^*\cdot f(t)dt = \left\langle\sum_{m,n}\langle f,\psi_{m,n}\rangle^*\cdot\psi_{m,n},f\right\rangle = A\langle f,f\rangle$$

$$\Rightarrow f=\frac{1}{A}\sum_{m,n}\langle f,\psi_{m,n}\rangle^*\cdot\psi_{m,n}=\frac{1}{A}\sum_{m,n}\langle\psi_{m,n},f\rangle\cdot\psi_{m,n}$$

A frame, even a tight frame is not an orthonormal basis. It is a set of *non-independent* vectors.

Example:

∘ Consider R^2 and a redundant set of vectors:

$$\varphi_0=[1,0]^T,\ \varphi_1=\left[-1/2,\sqrt{3}/2\right]^T,\ \varphi_2=\left[-1/2,-\sqrt{3}/2\right]^T$$

4.3.3. Reconstruction in Frames

Example:

∘ Create $M=[\varphi_0,\varphi_1,\varphi_2]$. One can verify that: $MM^T=\frac{3}{2}I\Rightarrow\ \forall x\in R^2,$
$x=\frac{2}{3}\sum_{i=0}^{2}\langle\varphi_i,x\rangle\varphi_i$

∘ Note that $\|\varphi_i\|_{i=0,1,2}=1$ and 3/2 is the redundancy factor. The vectors are linearly dependent:

B. Reconstruction in frames that are not tight

Proposition $\exists a,b,\ such\ that:\ \varphi_0=a\cdot\varphi_1+b\cdot\varphi_2\Rightarrow\ a=b=-1$

Given a family of functions $(\gamma_j)_{j\in J}$ that constitutes a *non-tight frame* in a Hilbert space H, there exists a family of functions $(\tilde{\gamma}_j)_{j\in J}$ called a *dual frame*, satisfying:

$$B^{-1}\|f\|^2\le\sum_{j\in J}\left|\langle\tilde{\gamma}_j,f\rangle\right|^2\le A^{-1}\|f\|^2,\ and$$

$$f=\sum_{j\in J}\langle\gamma_j,f\rangle\tilde{\gamma}_j=\sum_{j\in J}\langle\tilde{\gamma}_j,f\rangle\gamma_j$$

4.3.4. Frames of the CWT

Proposition

If $\psi_{m,n}(t)=a_0^{-m/2}\psi\left(a_0^{-m}t-nb_0\right),\ m,n\in Z$ constitutes a frame in $L^2(R)$ with frame bounds *A*, *B*, then:

Admissibility condition: $\frac{b_0\ln a_0}{2\pi}A\le\int_0^\infty\frac{|\Psi(\omega)|^2}{\omega}d\omega\le\frac{b_0\ln a_0}{2\pi}B$ and $\frac{b_0\ln a_0}{2\pi}A\le\int_{-\infty}^0\frac{|\Psi(\omega)|^2}{|\omega|}d\omega\le\frac{b_0\ln a_0}{2\pi}B$

The wavelets form a frame ⇒ admissibility condition is automatically satisfied.

Frame bounds for tight frames: $C_\psi=\int_{-\infty}^{\infty}\frac{|\Psi(\omega)|^2}{|\omega|}d\omega<\infty$

For orthonormal bases (e.g. the dyadic case), the wavelet should satisfy:

$$A=B=\frac{2\pi}{b_0\ln a_0}\int_0^\infty\frac{|\Psi(\omega)|^2}{\omega}d\omega=\frac{2\pi}{b_0\ln a_0}\int_{-\infty}^0\frac{|\Psi(\omega)|^2}{|\omega|}d\omega$$

$$A=B=1,\ a_0=2,\ b_0=1,\quad\int_0^\infty\frac{|\Psi(\omega)|^2}{\omega}d\omega=\int_{-\infty}^0\frac{|\Psi(\omega)|^2}{|\omega|}d\omega=\frac{\ln 2}{2\pi}$$

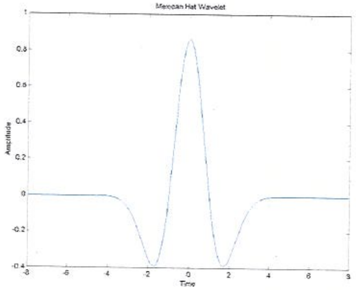
CW3

4.3.4. Frames of the CWT

Example - Mexican Hat wavelet frames

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

b_0	A	B	B/A
0.25	13.091	14.183	1.083
0.50	6.546	7.092	1.083
0.75	4.364	4.728	1.083
1.00	3.223	3.596	1.116
1.25	2.001	3.454	1.726
1.50	0.325	4.221	12.986



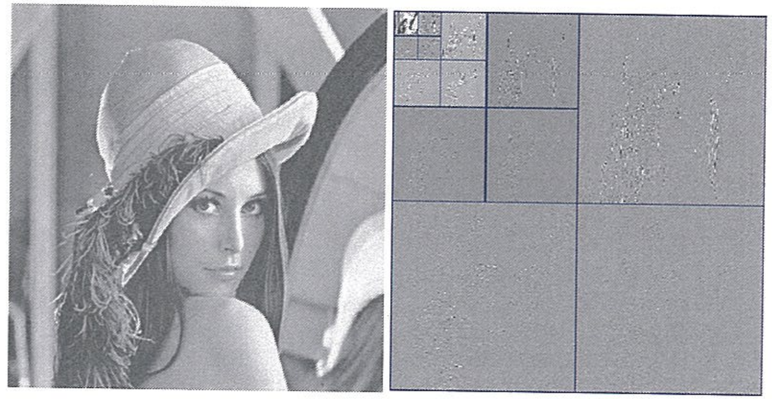
For some values of b_0 the frame is almost tight.

When the frame is almost tight, the frame bounds are inversely proportional to b_0; when b_0 is halved (twice as many points on the grid), the frame bounds should double - redundancy increases by two.

For b_0 higher than *1.50* the set $(\psi_{m,n})_{m,n\in Z}$ is not a frame anymore: $A<0$

5.3. The 2D Wavelet Representation

Example 2-D Wavelet Analysis

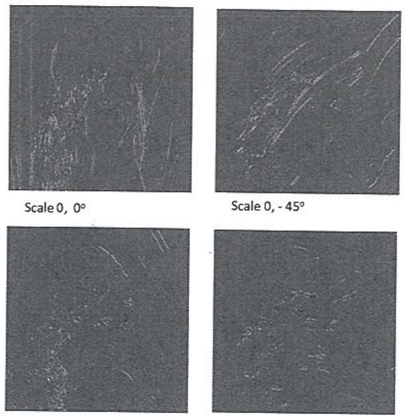


5.4 Applications

Intermediary Results

Different scales,

Different gradient orientations



Scale 0, 0°

Scale 0, - 45°

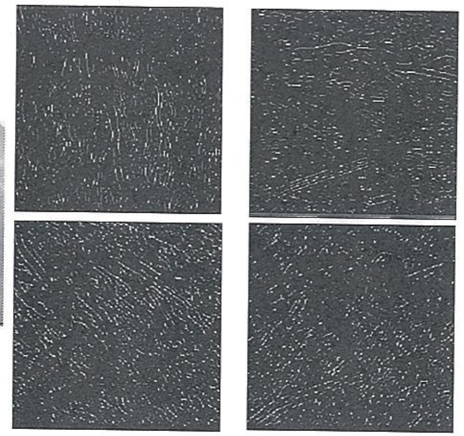
Scale 0, + 45°

Scale 0, 90°

5.4 Applications

Intermediary Results

Coronary Angiogram



Choosing a wavelet

Most applications (analysis, compression, noise-removal) exploit the ability of wavelet bases to efficiently approximate particular classes of functions with few non-zero coefficients. A function f has few non zero coefficients if most of the fine-scale wavelet coefficients are small (close to zero). Hence, the design of ψ must be optimized to produce a maximum number of coefficients $\langle f, \psi_{j,n} \rangle$ that are close to zero. This depends mostly on the regularity of f, the number of vanishing moments of ψ and the size of its support.

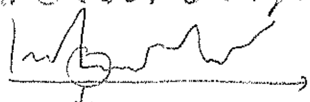
order of vanishing moments

The number of vanishing moments is defined as $\int_{-\infty}^{+\infty} \psi(x) x^p \, dx = 0 \quad p \in P$

If this order is low → lots of polynomials order of the signal kept → loose information
high → polynomials of high order (≈ large filter) of the signal kept → lot of information

The polynomial is the function defined to fit the best the signal:

$$f(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_k x^k + \text{noise}$$



discontinuities must be modelled.

It is thus important to have a high order of vanishing moment but then we loose locality.

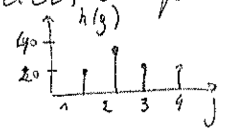
CHAPTER 4: IMAGE ENHANCEMENT

Image Enhancement is the art of playing with a broad set of available image processing tools, tuning their parameters and combining them in a skillful way. The goal is to improve the appearance of images to human viewers and to enhance the performance of subsequent image processing tools. Different methods exist such as contrast and/or dynamic range modification (dynamic range is the range spanned by the pixel gray values), noise reduction and/or edge enhancement or pseudo-coloring.

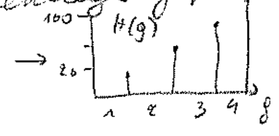
HISTOGRAM

The histogram $h(g)$ reflects the occupation level of the grey values in the image. The cumulative histogram reflects the percentage of pixels with a gray value $i \leq g$

Here: $i \leq g \rightarrow H(g) = \sum_{i=0}^{g} h(i)$



→

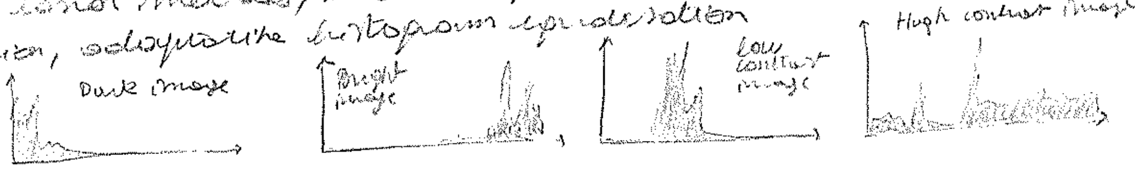


Example slide 3 à voir!

CONTRAST AND DYNAMIC RANGE MODIFICATION

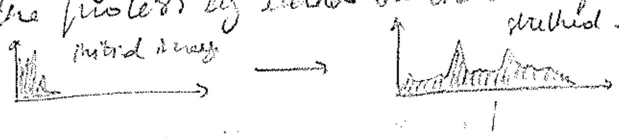
The quality of an image can be enhanced by modifying its histogram, i.e. applying a specific transformation in order to give to the histogram a desired shape. The following histogram based methods will be explained: histogram stretching, histogram equalization, adaptive histogram equalization

Histogram examples:



Histogram stretching

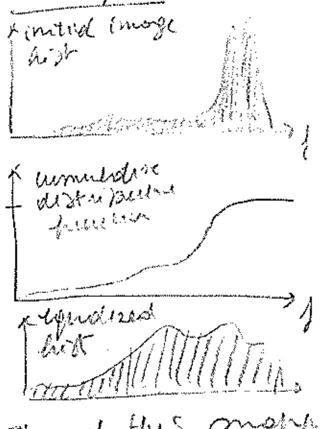
Stretching is the process by which the usable dynamic range of the image intensities is increased:



Histogram equalization

The processed histogram is aimed to be constant. It can be proven that this result can be obtained by using the cumulative distribution function of the input image as the transformation function

Example 1

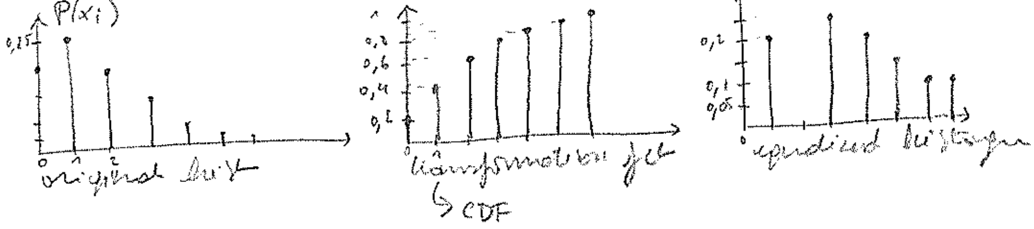


Example 2:

Probabilities	CDF	Actual CDF value	Mapped value
$P(0) = 0,2$	$CDF(0) = 0,2$	0,125 (0)	1 ; $P'(1) = 0,2 = P(0)$
$P(1) = 0,25$	$CDF(1) = 0,45$	0,25 (1)	3 ; $P'(3) = 0,25 = P(1)$
$P(2) = 0,2$	$CDF(2) = 0,65$	0,375 (2)	4 ; $P'(4) = 0,2 = P(2)$
$P(3) = 0,15$	$CDF(3) = 0,8$	0,5 (3)	5 ; $P'(5) = 0,15 = P(3)$
$P(4) = 0,1$	$CDF(4) = 0,9$	0,625 (4)	6 ; $P'(6) = 0,1 = P(4)$
$P(5) = 0,06$	$CDF(5) = 0,96$	0,750 (5)	7
$P(6) = 0,03$	$CDF(6) = 0,99$	0,875 (6)	7
$P(7) = 0,01$	$CDF(7) = 1$	1 (7)	7 ; $P'(7) = 0,1$

chance of seeing $P'(7)$
$= P(5) + P(6) + P(7)$
$= 0,06 + 0,03 + 0,01$
$= 0,1$

Through this mapping, we draw:



CH4

Histogram of an image

$n_C=7$

$n_R=5$

1	3	2	4	3	2	1
2	2	3	4	1	3	2
3	4	4	4	1	1	2
3	2	1	4	1	3	2
2	2	2	3	3	4	1

Number pixels = N = $n_R * n_C$ = 35

Every pixel can only take 4 values

Pixel value: g	No pixels: N(g)	N(g) /N
1	8	23%
2	11	31%
3	9	26%
4	7	20%
	35	

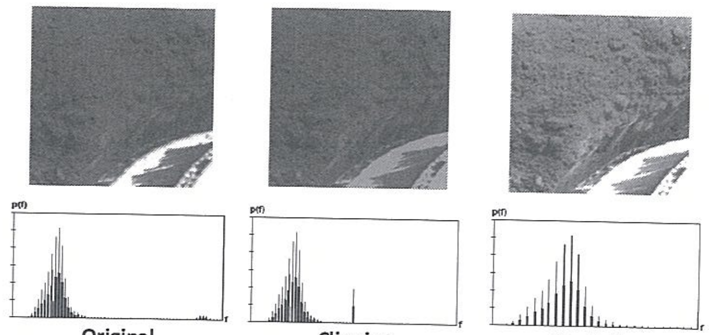
The histogram shows the percentage of pixels having a certain grey value: h(g) = N(g)/N

Is an estimate for the probability for grey value g

Alternative Histogram Modification techniques (1/2)

Recipe for clipping before stretching:

1. Clip the pixel gray values to a range [0,128]
2. Stretch the clipped image

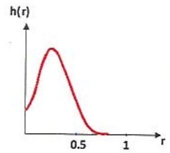


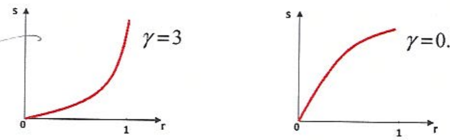
Alternative Histogram Modification techniques: γ-correction (2/2)

Mapping function: $s = r^{\gamma}$

with r being normalised between 0 and 1

EXAMPLE

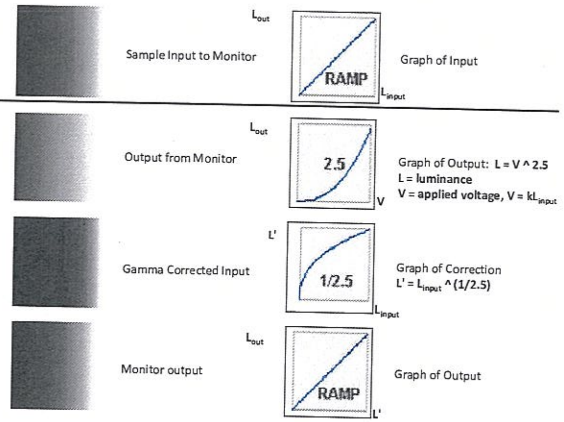




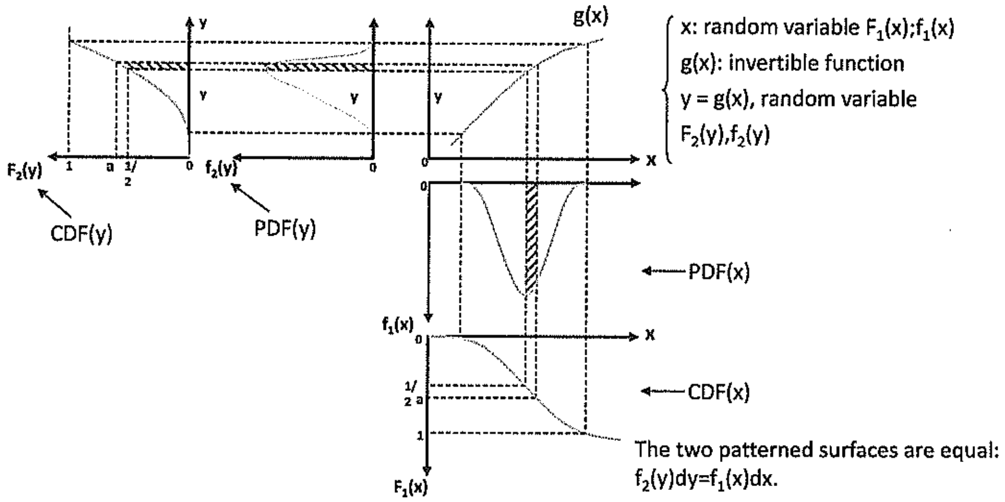
enhanced toward dark shades

→ enhanced toward light shades

Gamma Correction for CRT-Monitors



The CDF is the transform to be used in histogram equalisation



$\forall y$ for which g^{-1} exists:

$$F_2(y)=P(\mathbf{y}\le y)=P\left[\mathbf{x}\le g^{-1}(y)\right]=F_1\left[g^{-1}(y)\right]\ (*)$$

For the probability distribution functions, we derive:

$$f_1(x)=\frac{dF_1(x)}{dx}\ \text{ and }\ f_2(y)=\frac{dF_2(y)}{dy},$$

$$f_2(y)=\frac{dF_2(y)}{dy}=\frac{dF_1\left[g^{-1}(y)\right]}{dg^{-1}(y)}\frac{dg^{-1}(y)}{dy}=f_1\left[g^{-1}(y)\right]\frac{dg^{-1}(y)}{dy}=f_1(x)\frac{dx}{dy}$$

If we wish $f_2(y)$ to be uniform, what should we use for g( )?

$$\begin{cases} 0\le y\le 1\rightarrow f_2(y)=1 & \Rightarrow dy=f_1(x)dx\Leftrightarrow y=\int_0^x f_1(t)dt \\ \text{otherwise}\quad f_2(y)=0 & \Leftrightarrow y=g(x)=F_1(x)=cdf(x) \end{cases}$$

Notice also that: $F_2(y)=F_1\left[g^{-1}(y)\right]=F_1\left[F_1^{-1}(y)\right]=y$

Conclusion: $y=g(x)=F_1(x)$ maps $f_1(x)$ towards a uniform distribution $f_2(y)$

$x=g^{-1}(y)=F_1^{-1}(y)$ maps a uniform distribution $f_2(y)$ towards $f_1(x)$

Slide 15 à être revu par le professeur mais à aller voir

Adaptive histogram equalization

The problem of histogram equalization is that it is often difficult to distinguish between low contrast details and noise. The process of adaptive histogram equalization is:

- calculate an histogram and equalize over a window around the pixel of interest
- assign an a priori defined characteristic value (e.g. median) of the equalized histogram to the pixel of interest.

Other techniques also exists such as clipping + stretching (see slide 18) or γ-correction (see slide 19-20)

NOISE REDUCTION & EDGE ENHANCEMENT

Linear filtering

- continuous convolution of f with a filter g: $f(x,y) * g(x,y) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(\alpha,\beta)\, g(x-\alpha, y-\beta)\, d\alpha\, d\beta$

$= \int_{+\infty}^{-\infty}\int_{+\infty}^{-\infty} f(x-s, y-t)\, g(s,t)\,(-ds)(-dt)$ with $\begin{cases} s = x-\alpha \\ t = y-\beta \end{cases}$ $\begin{cases} ds = -d\alpha \\ dt = -d\beta \end{cases}$

$= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x-s, y-t)\, g(s,t)\, ds\, dt$

by comparison with the definition of correlation: $f(x,y) \circ g(x,y) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(\alpha,\beta)\, g(x+\alpha, y+\beta)\, d\alpha\, d\beta$

we see that the convolution can be resumed to a correlation with a filter rotated at 180°. As correlation in the frequency domain can be written as $F^*(u,v)\, G(u,v)$ a simple product, we deduce that the most efficient computation is done in the domain

N.B: if f = M non zero samples, g = N non zero samples → convolution has M+N−1 non zero samples as

f: 0 … M−1 } → M+N−1 non zeros

g: 0 N−1

- discrete: convolution: $f(x,y) * g(x,y) = \sum_{\alpha=-M}^{+M}\sum_{\beta=-M}^{+M} g(x-\alpha, y-\beta)\, f(\alpha,\beta)$

$= \sum_{\alpha'=M}^{-M}\sum_{\beta'=M}^{-M} g(x+\alpha', y+\beta')\, f(-\alpha', -\beta')$

$= \sum_{\alpha'=-M}^{M}\sum_{\beta'=-M}^{M} g(x+\alpha', y+\beta')\, f_{rot180}(\alpha',\beta')$

correlation: $f(x,y) \circ g(x,y) = \sum_{\alpha=-M}^{+M}\sum_{\beta=-M}^{+M} g(x+\alpha, y+\beta)\, f(\alpha,\beta)$

↳ $(2M+1) \times (2M+1)$ mask

↳ convolution of 2 3x3 mask → 5x5 mask

→ lin. slide 26 - 27 - 28

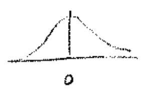
- composed convolution: $filt_1(x,y) * (filt_2(x,y) * im(x,y)) = (filt_1(x,y) * filt_2(x,y)) * im(x,y)$

$filt_2(x,y) * filt_1(x,y) * im(x,y)$

- $FILT_1(u,v)\,(FILT_2(u,v) \cdot Im(u,v)) = (FILT_1(u,v) \cdot FILT_2(u,v))\, Im$

Lowpass filtering

It allows to reduce the noise variance at the expense of distorting the signal by reducing the high frequency components while preserving the low frequency components. The larger the mask is, the less we detect edges and the worse is their exact location evaluated. Typical masks are:

PSF 

1/9

1	1	1
1	1	1
1	1	1

1/10

1	1	1
1	2	1
1	1	1

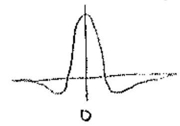
1/16

1	2	1
2	4	2
1	2	1

EXAM: explain

Unsharp masking

It accentuates the high frequency components of the image. The sum of the coefficients is one. This characteristic has the effect of preserving the average intensity of the original image in the filtered image. The edges are better highlighted but the background noise has been increased because it has typically significantly high frequency components.



0	-1	0
-1	5	-1
0	-1	0

1	-2	1
-2	5	-2
1	-2	1

1/2

-1	-2	-1
-2	19	-2
-1	-2	-1

Unsharp masking is closely related to highpass filtering. It is the general process of adding a highpass filtered version of an image to a fraction of the original image:

unsharp $= (\alpha - 1) *$ original $+$ highpass, with α amplification factor,

such that Σ highpass components $= 0$ and Σ unsharp components $= 1$

If $\alpha = 1 \rightarrow$ standard highpass result, if $\alpha > 1$, part of the original image is added to the highpass result, which partially restores the low frequency components lost in the highpass filtering operation. As a result, the image after unsharp masking looks more like the original image, with a relative degree of edge enhancement that depends on the value of α.

A highpass filter

-1	-1	-1
-1	8	-1
-1	-1	-1

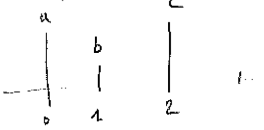
an unsharp mask with $W = 9*\alpha - 1$ and $\alpha > 1$

1/9

-1	-1	-1
-1	W	-1
-1	-1	-1

Median filtering

The median filter is a non-linear filter. It replaces the value of the pixel located at the center of the moving mask by the median of the gray values of the pixel inside the mask. The goal is to reduce impulse or salt-and-pepper noise, preserves edges * (adapt window size to width of the spikes and their density ??),



1°) sort: $b < a < c$

2°) replace center value (1) by median value a: $b \rightarrow a$

* while reducing random noise and preserve corners (adapt window shape ??)

Example: stroke 3×3, the 3 points median filter doesn't work because when the filter is center on , the median value resulting is always □

↳ solution: increase filter size.

Median filter is characterized by smoothening on the edge.

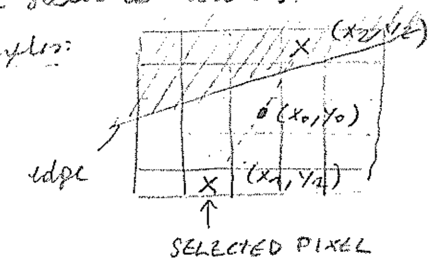
Symmetric nearest neighbour

This type of filter can detect an edge and select the pixel that we suppose to be filtered such that it doesn't filter the edge out. The goal is to flatten the interior of uniform regions while at the same time enhancing blurred edges. It does this by selecting the pixels as such, supposing that the mask is centered at pixel (x_0, y_0):

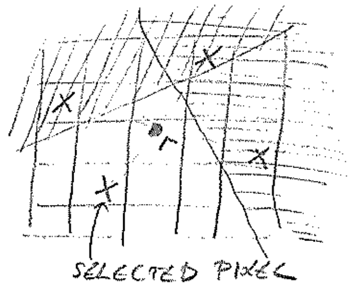
- if $|f(x_0,y_0) - f(x_1,y_1)| < |f(x_0,y_0) - f(x_2,y_2)|$ → select $f(x_1,y_1)$
- if $|f(x_0,y_0) - f(x_1,y_1)| > |f(x_0,y_0) - f(x_2,y_2)|$ → select $f(x_2,y_2)$
- if $|f(x_0,y_0) - f(x_1,y_1)| = |f(x_0,y_0) - f(x_2,y_2)|$ → select $f(x_0,y_0)$

and the value of the pixel at position (x_0,y_0) is replaced by the mean or median of the selected values.

Examples:



or filter preserving edges intersecting at 90° (looks at 4 positions at same time)



PSEUDO COLOURING

The gray scale image is transformed to a colour image (e.g. by mapping each gray level or a range of levels onto a different colour, which is called pseudo colour)
The advantages are that the coloured image enables easier identification of pixels having different gray values. Also, colours allow quick identification of all regions in the image having the same pixel values.
The disadvantages are usually destroys the spatial pattern of image (false contours might be created, very similar gray values can get completely different colours)