COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

General Image Transforms and Applications

Lecture 6, March 2nd, 2009

Lexing Xie

EE4830 Digital Image Processing
http://www.ee.columbia.edu/~xlx/ee4830/

thanks to G&W website, Min Wu, Jelena Kovacevic and Martin Vetterli for slide materials

announcements

- HW#2 due today
- HW#3 out

- Midterm next week, class time+location
  - Monday March 9th (4:10-6:40, Mudd 1127)
  - “Open-book”
    - YES: text book(s), class notes, calculator
    - NO: computer/cellphone/matlab/internet
  - 5 analytical problems
  - Coverage: lecture 1-6
    - intro, representation, color, enhancement, transforms and filtering (until DFT and DCT)
  - Additional instructor office hours
    - 2-4pm Monday March 9th, Mudd 1312

- Grading breakdown
  - HW-Midterm-Final: 30%-30%-40%

outline

- Recap of DFT and DCT
- Unitary transforms
- KLT
- Other unitary transforms
- Multi-resolution and wavelets
- Applications

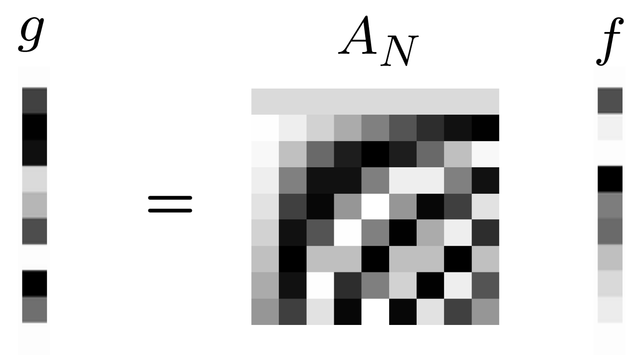
- Readings for today and last week: G&W Chap 4, 7, Jain 5.1-5.11

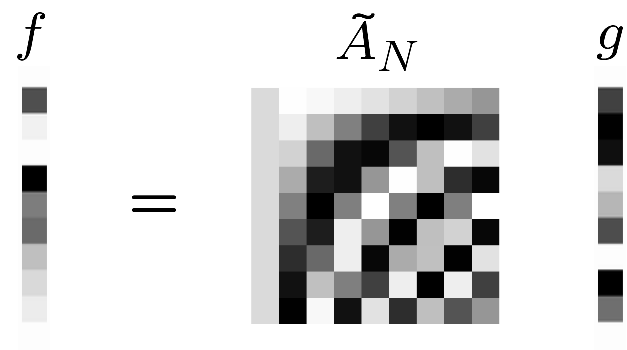
recap: transform as basis expansion

$$g(u) = \sum_{n=0}^{N-1} f(n)\ a_N^{un}$$

$$f(n) = \sum_{u=0}^{N-1} g(u)\ \tilde{a}_N^{un}$$

inverse transform





$$\tilde{a}_N^{un} = a_N^{*un}$$

DFT: $a_N^{un} = e^{-j2\pi\frac{un}{N}}, \quad \tilde{A}_N = A_N^{*T}$

DCT:

$$a_N^{0n} = \sqrt{\frac{1}{N}} \qquad u = 0$$

$$a_N^{un} = \sqrt{\frac{2}{N}} cos\frac{\pi(2n+1)u}{2N} \qquad u = 1, \ldots, N-1$$

$$\tilde{a}_N^{un} = a_N^{un}$$

$$\tilde{A}_N = A_N^T$$

recap: DFT and DCT basis

1D-DCT

$$a_N^{0n} = \sqrt{\frac{1}{N}} \qquad u = 0$$

$$a_N^{un} = \sqrt{\frac{2}{N}} cos \frac{\pi(2n+1)u}{2N}$$

$$u = 1, 2, \ldots, N-1$$

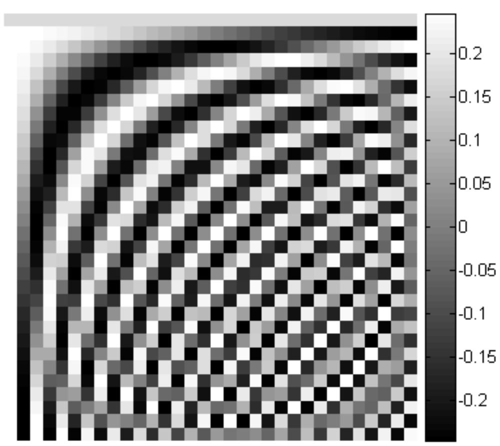
1D-DFT

$$a_N^{un} = e^{-j2\pi\frac{un}{N}}$$

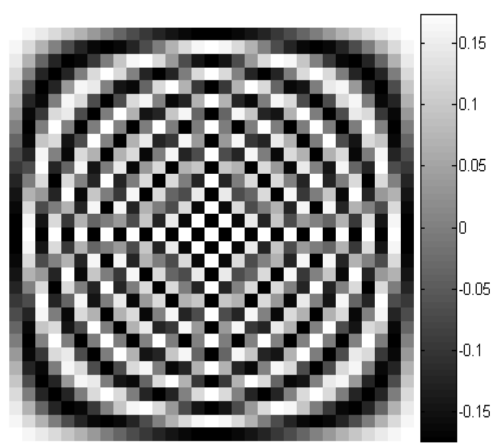
$$= cos(2\pi\frac{un}{N}) - jsin(2\pi\frac{un}{N})$$

N=32

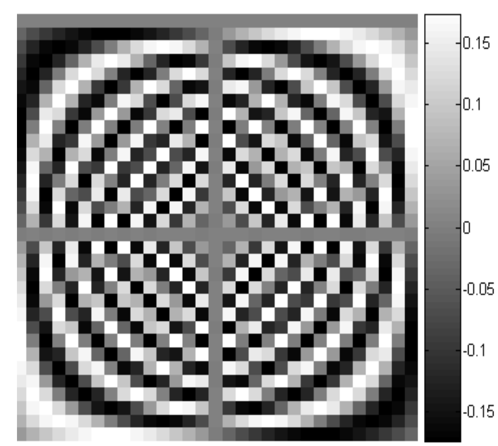
A



real(A)



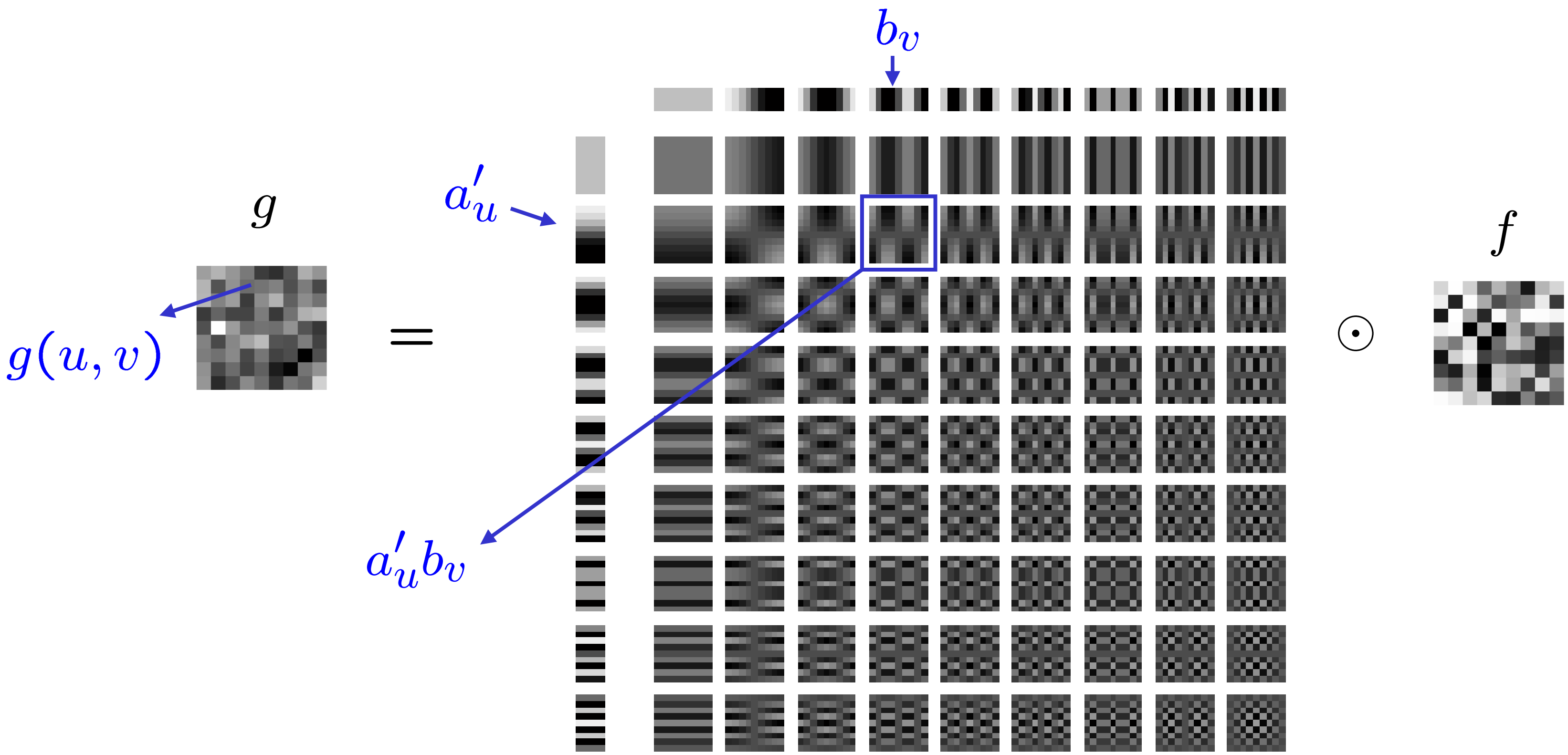
imag(A)



recap: 2-D transforms

$$g(u,v) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(m,n)a_{uv}(m,n), \quad f(m,n) = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1} g(u,v)\tilde{a}_{uv}(m,n)$$

for DFT, $a(u,v,\ m,n) = e^{-j2\pi(\frac{um}{N}+\frac{vn}{N})} = e^{-j2\pi\frac{um}{N}} \cdot e^{-j2\pi\frac{vn}{N}}$



A transform is *separable*,
when $a_{uv}(m,n) = a_u(m)b_v(n)$.

2D-DFT and 2D-DCT are separable transforms.

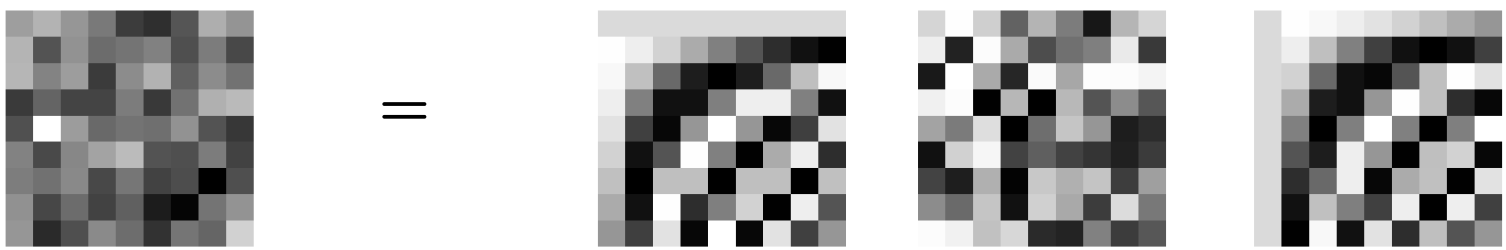
separable 2-D transforms

when $a = b,\ M = N$

$$g(u,v) = \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} a_N^{um}\ f(m,n)\ a_N^{vn}$$

$$f(m,n) = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1} \tilde{a}_N^{um}\ g(u,v)\ \tilde{a}_N^{vn}$$

$g \qquad = \qquad A_N \qquad f \qquad A'_N$



$$g = A_N f A'_N \implies g' = A_N (A_N f)'$$

Symmetric 2D separable transforms can be expressed with the notations of its corresponding 1D transform. → We only need to discuss 1D transforms

Exercise

- How do we decompose this picture?

DCT2 → ?

■ =-1

□ =1

$$\begin{bmatrix} -1 & -1 \\ -1 & 1 \end{bmatrix}$$

■ =0

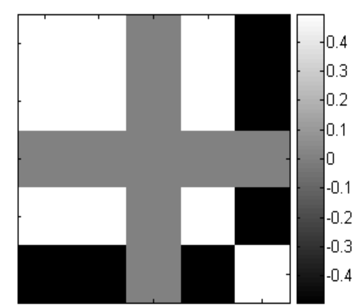
□ =1

$$\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \xrightarrow{\text{DCT2}} \begin{bmatrix} ? & ? \\ ? & ? \end{bmatrix}$$

What if black=0, does the transform coefficients look similar?

$$A_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Four basis images for 2D DCT



two properties of DFT and DCT

$$g(u) = \sum_{n=0}^{N-1} f(n)\; a_N^{un} \qquad \tilde{A}_N = A_N^{*T}$$

$$f(n) = \sum_{u=0}^{N-1} g(u)\; \tilde{a}_N^{un}$$

- Orthonormal (Eq 5.5 in Jain)
  : no two basis represent the same information in the image

$$\sum_n a_N^{un} a_N^{*vn} = \delta(u-v)$$

- Completeness (Eq 5.6 in Jain)
  : all information in the image are represented in the set of basis functions

$$\sum_u a_N^{um} a_N^{*un} = \delta(m-n)$$

⇨

for $Q < N$, let $f_Q(n) = \sum_{u=0}^{Q-1} \hat{g}(u) a_N^{*un}$

$\sigma_Q^2 = \sum_{n=1}^{N-1} [f(n) - f_Q(n)]^2$ minimized when $\hat{g}(u) = g(u)$

$f - f_Q = 0$, iff. $Q = N$

Unitary Transforms

A linear transform:

$$\mathcal{R}^N \rightarrow \mathcal{R}^N \qquad g = A_N\ f, \quad f = A_N^{*T}\ g$$

The Hermitian of matrix A is: $A^H = A^{*T}$

This transform is called "unitary" when A is a unitary matrix, "orthogonal" when A is unitary and real.

$$A^{-1} = A^H, \ AA^H = A^* A^T = I$$

- Two properties implied by construction
  - Orthonormality

$$\sum_n a_N^{un} a_N^{*vn} = \delta(u - v)$$

  - Completeness

$$\sum_u a_N^{um} a_N^{*un} = \delta(m - n)$$

Exercise

- Are these transform matrixes unitary/orthogonal?

$$\begin{bmatrix} 2 & 3 \\ 1 & 2 \end{bmatrix} \qquad \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \qquad \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$$

$$\begin{bmatrix} \sqrt{2} & j \\ -j & \sqrt{2} \end{bmatrix} \qquad \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & j \\ j & 1 \end{bmatrix}$$

- Unitary/orthogonal checklist:
  - determinant equals 1, $|A|=1$
  - unit row/column vector
  - orthogonal row/column vectors, $AA^H=I$

properties of 1-D unitary transform

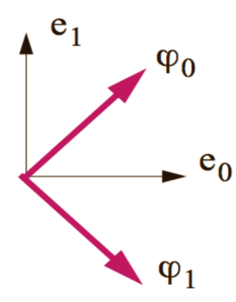
- energy conservation $\|g\|^2 = \|f\|^2$

$$\|g\|^2 = \|Af\|^2 = (Af)^{*T}(Af) = f^{*T}A^{*T}Af = f^{*T}f = \|f\|^2$$

- rotation invariance
  - the angles between vectors are preserved

$$cos\theta = \frac{f_1 \cdot f_2}{\|f_1\|\|f_2\|} \qquad g_1 \cdot g_2 = g_1^{*T} g_2 = (Af_1)^{*T} A f_2 = f_1 \cdot f_2$$

  - unitary transform: rotate a vector in R^n, i.e., rotate the basis coordinates



observations about unitary transform

- Energy Compaction
  - Many common unitary transforms tend to pack a large fraction of signal energy into just a few transform coefficients
- De-correlation
  - Highly correlated input elements → quite uncorrelated output coefficients
  - Use the covariance matrix to measure correlation

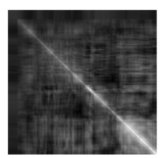
$$R_g = cov(g) = E\{(g - E\{g\})(g - E\{g\})^{*T}\}$$

let $\hat{g} = g - E\{g\}$, then $R_{mn} = E\{\hat{g}_m \hat{g}_n\}$

f: columns of image pixels

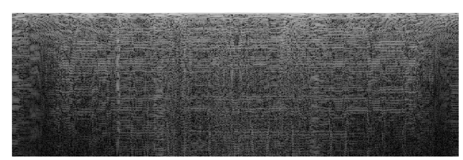
$g = DCT(f)$

$f_1, f_2, \ \ldots, \ f_{600}$ $\quad cov(f)$ $\quad g_1, g_2, \ \ldots, \ g_{600}$ $\quad cov(g)$

linear display scale: g



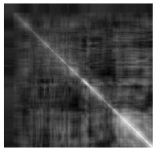



display scale: log(1+abs(g))

one question and two more observations

- is there a transform with
  - best energy compaction
  - maximum de-correlation
  - is also unitary… ?
- transforms so far are data-independent
  - transform basis/filters do not depend on the signal being processed
  - "optimal" should be defined in a statistical sense so that the transform works well with many images
    - "optimal" for each signal is ill-defined
  - signal statistics should play an important role

review: correlation after a linear transform



- x is a zero-mean random vector in $\mathcal{R}^N$

  $E[x] = 0$

  - the covariance (autocorrelation) matrix of x

    $R_x = cov(x) = E[xx^H]$

    - $R_x(i,j)$ encodes the correlation between x_i and x_j
    - R_x is a diagonal matrix iff. all N random variables in x are uncorrelated

- apply a linear transform: $y = Ax$
  - What is the correlation matrix for y ?

$$R_y = cov(y) = E[yy^H] = E[Ax(Ax)^H]$$

$$= E[Axx^H A^H] = AE[xx^H]A^H = AR_xA^H$$

transform with maximum energy compaction

$$y = A'x$$

$$y(u) = a'_u x$$

$$A' = \begin{bmatrix} a'_0 \\ a'_1 \\ \cdots \\ a'_{N-1} \end{bmatrix}$$

$$a'_u a^*_u = 1$$
$$a'_u a^*_v = 0 \quad \forall u \neq v$$

$$\|x\|^2 = E[x^H x] = \sum_u R_x(u,u)$$
$$\|y\|^2 = E[y^H y] = \|x\|^2$$
$$\|y_Q\|^2 = \sum_{u=0}^{Q-1} y^2(u)$$

$$\text{max. } E[y_Q^H y_Q]$$
$$\text{s.t. } y(u) = a'_u x, \quad a'_u a^*_u = 1, \; a'_u a^*_v = 0 \quad \forall u \neq v$$

proof. maximum energy compaction

max. $E[y_Q^H y_Q] = E[(A_Q x)^H A_Q x]$

$$A_Q = \begin{pmatrix} a_0' \\ \cdots \\ a_{Q-1}' \\ \cdots \\ 0 \end{pmatrix}$$

$$= E[x^H \begin{pmatrix} a_0^* & \ldots & a_{Q-1}^* & \ldots 0 \end{pmatrix} \begin{pmatrix} a_0' \\ \cdots \\ a_{Q-1}' \\ \cdots \\ 0 \end{pmatrix} x]$$

$$= E[x^H \sum_{u=0}^{Q-1} a_u^* a_u' \; x]$$

matrix identity →

$$= \sum_{u=0}^{Q-1} a_u' R_x a_u^*$$

constraints:

$y(u) = a_u' x$

$a_u' a_u^* = 1$

$a_u' a_v^* = 0$

$a_u' a_u^* = 1$ →

let $L = \sum_{u=0}^{Q-1} a_u' R_x a_u^* - 2\sum_{u=0}^{Q-1} \lambda_u (1 - a_u' a_u^*)$

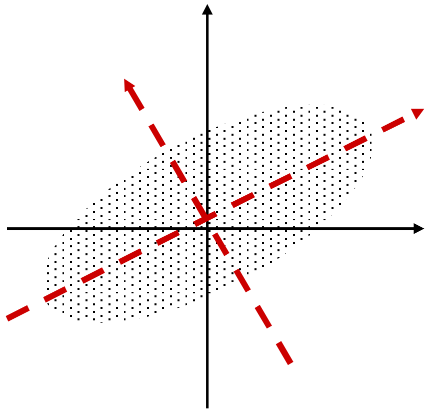
$$\frac{\partial L}{\partial a_u^*} = 2R_x a_u^* - 2\lambda_u a_u^* = 0$$

⇨ a^*_u are the eigen vectors of R_x

$$R_x a_u^* = \lambda_u a_u^*$$

Karhunen-Loève Transform (KLT)

- a unitary transform with the basis vectors in A being the “orthonormalized” eigenvectors of R_x



$$y = A^T x,\ x = Ay,$$

$$\text{with } A \in \mathcal{R}^{N \times N},\ A = [a_0, \ldots, a_{N-1}]$$

$$R_x a_u = \lambda_u a_u,\ u = 0, \ldots, N-1$$

  - assume real input, write A^T instead of A^H
  - denote the inverse transform matrix as A, $AA^T=I$
  - R_x is symmetric for real input, Hermitian for complex input i.e. $R_x^T=R_x$, $R_x^H = R_x$
  - R_x nonnegative definite, i.e. has real non-negative eigen values

- Attributions
  - Kari Karhunen 1947, Michel Loève 1948
  - a.k.a Hotelling transform (Harold Hotelling, discrete formulation 1933)
  - a.k.a. Principle Component Analysis (PCA, estimate R_x from samples)

Properties of K-L Transform

- Decorrelation by construction

$$R_y = E[yy^T] = AR_xA^T \quad = \begin{pmatrix} \lambda_0 & & & \\ & \lambda_1 & & \\ & & \cdots & \\ & & & \lambda_{N-1} \end{pmatrix}$$

  - note: other matrices (unitary or nonunitary) may also de-correlate the transformed sequence [Jain's example 5.5 and 5.7]

- Minimizing MSE under basis restriction
  - Basis restriction: Keep only a subset of m transform coefficients and then perform inverse transform (1≤ m ≤ N)

  → Keep the coefficients w.r.t. the eigenvectors of the first m largest eigenvalues

$$v = A^T u \quad w = I_m v \quad z = Bw$$

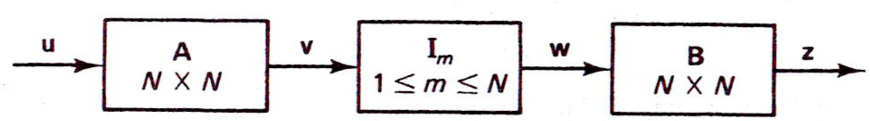


Figure 5 16 KL transform basis restriction

$$I_m = \begin{pmatrix} 1 & & & & \\ & 1 & & & \\ & & \cdots & & \\ & & & 0 & \\ & & & & \cdots \end{pmatrix}$$

discussions about KLT

- The good
  - Minimum MSE for a “shortened” version
  - De-correlating the transform coefficients
- The ugly
  - Data dependent
    - Need a good estimate of the second-order statistics
    - Increased computation complexity

data: $x_1, \ldots, x_M \in \mathcal{R}^N$

estimate R_x: $O(MN)$

linear transform: $O(MN)$

compute eig R_x: $\sim O(N^3)$

fast transform: $O(M \log N)$

Is there a data-independent transform with similar performance?

DFT is the optimal transform when …

- The signal x is periodic

$$x(m) = x(m+n),\ \forall m$$

- The autocorrelation matrix R_x is circulant

$$R_x = E[xx^H] = \begin{bmatrix} r_0 & r_1 & \cdots & r_{n-1} \\ r_{n-1} & r_0 & \cdots & r_{n-2} \\ \cdots & & \cdots & \\ r_1 & r_2 & \cdots & r_0 \end{bmatrix}$$

- The eigen vectors of R_x are Fourier basis

$$R_x W_n^u = \lambda_u W_n^u$$

energy compaction properties of DCT

- DCT is close to KLT when …
  - x is first-order stationary Markov $x_n = \rho x_{n-1} + z_n, \quad z_n \sim \mathcal{N}(0, \sigma_z^2), \ |\rho| < 1$

$$\rightarrow \quad E[x_n x_{n-1}] = \rho\sigma_x^2, \ E[x_n x_{n-2}] = \rho^2\sigma_x^2, \ \ldots \quad r(n) = \rho^{|n|}$$

$$\rightarrow \quad R_x = \begin{pmatrix} 1 & \rho & \rho^2 & \ldots & \\ \rho & 1 & \rho & & \\ \ldots & & \ldots & & \\ \rho^{n-1} & & & & 1 \end{pmatrix}$$

$$\beta^2 \triangleq \frac{\rho^2}{1+\rho^2}$$

$$\alpha \triangleq \frac{\rho}{1+\rho^2}$$

$$\rightarrow \quad \beta^2 R_x^{-1} = \begin{pmatrix} 1-\rho\alpha & -\alpha & & & \\ -\alpha & 1 & -\alpha & 0 & \\ \ldots & & \ldots & & \\ & 0 & & -\alpha & 1-\rho\alpha \end{pmatrix}$$

- R_x and $\beta^2 R_x^{-1}$ have the same eigen vectors
- $\beta^2 R_x^{-1} \sim Q_c$ when ρ is close to 1

- DCT basis vectors are eigenvectors of a symmetric tri-diagonal matrix Q_c

$$Q_c = \begin{pmatrix} 1-\alpha & -\alpha & 0 & \ldots & & \\ -\alpha & 1 & -\alpha & & & \\ \ldots & & \ldots & & & \\ 0 & & & -\alpha & 1-\alpha \end{pmatrix}$$

$$a_0 = const.$$

$$a_u \propto \left[1, \ cos\frac{\pi 3u}{2N}, \ \ldots, \ cos\frac{\pi u(2N-1)}{2N}\right]^T$$

$$\rightarrow \quad Q_c a_u = \lambda_u a_u$$

verify with trigonometric identity:
cos(a+b)+cos(a-b)=2cos(a)cos(b)

DCT energy compaction

- DCT is close to KLT for
  - highly-correlated first-order stationary Markov source

- DCT is a good replacement for KLT
  - Close to optimal for highly correlated data
  - Not depend on specific data
  - Fast algorithm available

DCT/KLT example for vectors

x: columns of image pixels ρ* = 0.8786

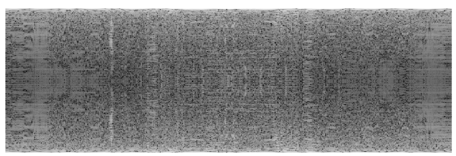
fraction of coefficient values in the diagonal

$x_1, x_2, \ \ldots, \ x_{600}$

R_x

0.0136

$abs(DFT_{1D}(x))$

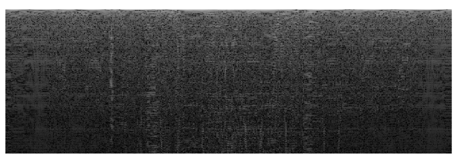


$R_{DFT(x)}$

0.1055

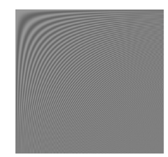
transform basis

$DCT_{1D}(x)$

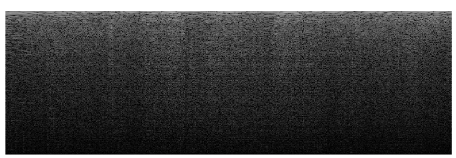


$R_{DCT(x)}$

0.1185

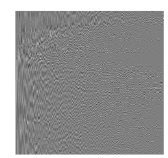


$KLT_{1D}(x)$



$R_{KLT(x)}$

1.0000



display scale: log(1+abs(g)), zero-mean

KL transform for images

- autocorrelation function 1D → 2D

$x(1:n)$ $\quad\quad$ $R_x(n_1, n_2)$

$x(1:m,\ 1:n)$ $\quad\quad$ $R_x(m_1, m_2, n_1, n_2)$

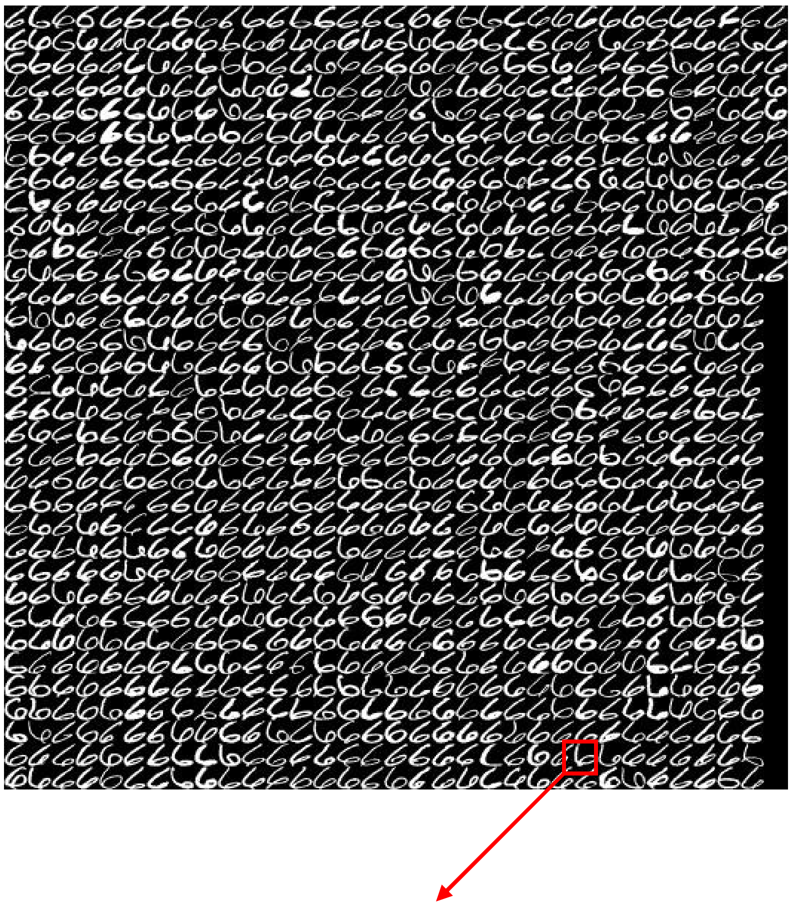
- KL basis images are the orthonormalized eigen-functions of *R*
- rewrite images into vector forms (N^2x1)
  - solve the eigen problem for N^2xN^2 matrix ~ O(N^6)

- or, make R_x "separable"

$$R_x(m_1, m_2, n_1, n_2) \rightarrow r(m_1, m_2) \cdot r(n_1, n_2)$$

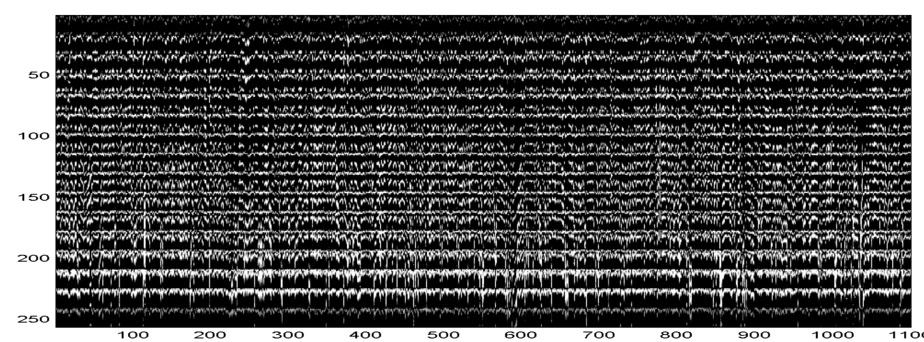
  - perform separate KLT on the rows and columns
  - transform complexity O(N^3)

KLT on hand-written digits …



1100 digits “6”
16x16 pixels

1100 vectors of size 256x1



The Desirables for Image Transforms

	DFT	DCT	KLT
Theory			
Inverse transform available	✓	✓	✓
Energy conservation (Parsevell)	✓	✓	✓
Good for compacting energy	?	?	✓
Orthonormal, complete basis	✓	✓	✓
(sort of) shift- and rotation invariant	✓	✓	?
Transform basis signal-independent			
Implementation			
Real-valued	x	✓	✓
Separable	✓	✓	x
Fast to compute w. butterfly-like structure	✓	✓	x
Same implementation for forward and inverse transform	✓	✓	x

Walsh-Hadamard Transform

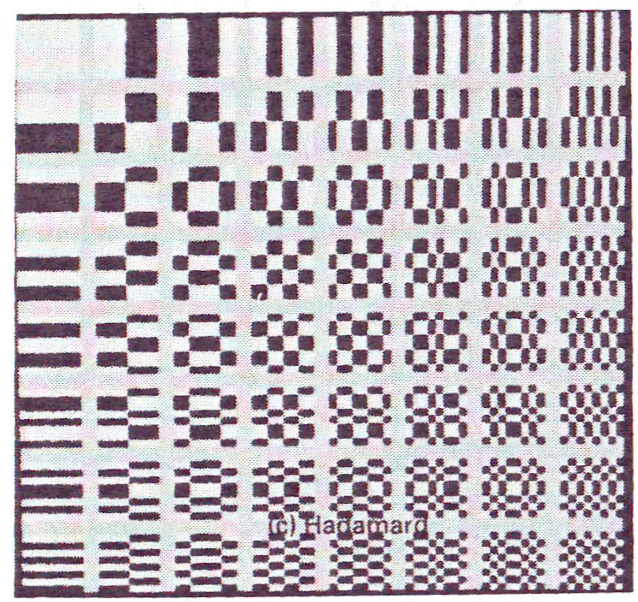
$$H_0 = +1$$

$$H_1 = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$H_m = \frac{1}{\sqrt{2}}\begin{pmatrix} H_{m-1} & H_{m-1} \\ H_{m-1} & -H_{m-1} \end{pmatrix},$$

$$(H_m)_{k,n} = \frac{1}{2^{m/2}}(-1)^{\sum_j k_j n_j}$$

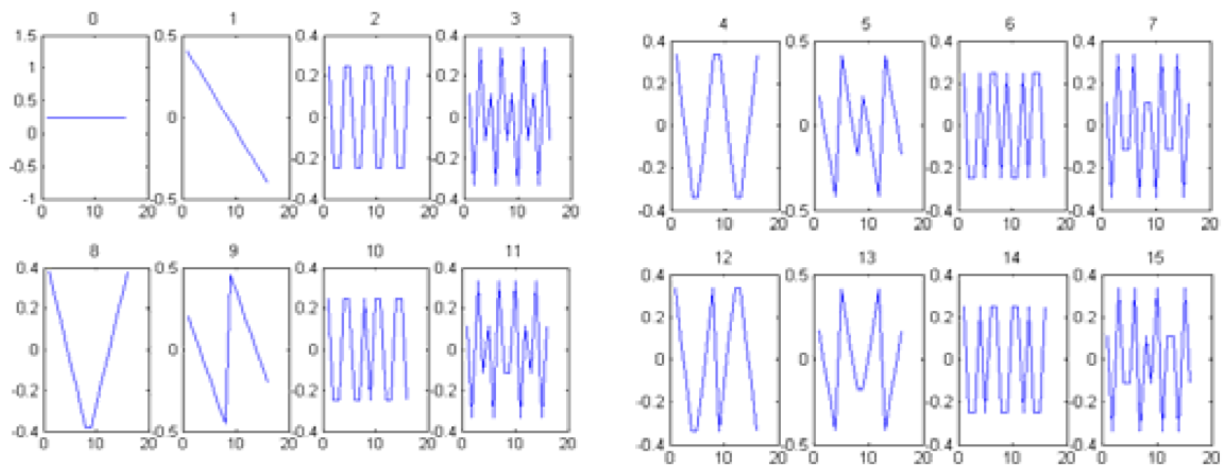
$$H_2 = \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}$$



$$H_3 = \frac{1}{2^{3/2}}\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{pmatrix}.$$

slant transform

$$\begin{bmatrix}
0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 \\
0.5401 & 0.3858 & 0.2315 & 0.0772 & -0.0772 & -0.2315 & -0.3858 & -0.5401 \\
0.3536 & -0.3536 & -0.3536 & 0.3536 & 0.3536 & -0.3536 & -0.3536 & 0.3536 \\
0.1581 & -0.4743 & 0.4743 & -0.1581 & 0.1581 & -0.4743 & 0.4743 & -0.1581 \\
0.4743 & 0.1581 & -0.1581 & -0.4743 & -0.4743 & -0.1581 & 0.1581 & 0.4743 \\
0.2415 & -0.0345 & -0.3105 & -0.5866 & 0.5866 & 0.3105 & 0.0345 & -0.2415 \\
0.3536 & -0.3536 & -0.3536 & 0.3536 & -0.3536 & 0.3536 & 0.3536 & -0.3536 \\
0.1581 & -0.4743 & 0.4743 & -0.1581 & -0.1581 & 0.4743 & -0.4743 & 0.1581
\end{bmatrix}$$



Nassiri et. al, “Texture Feature Extraction using Slant-Hadamard Transform”

energy compaction comparison

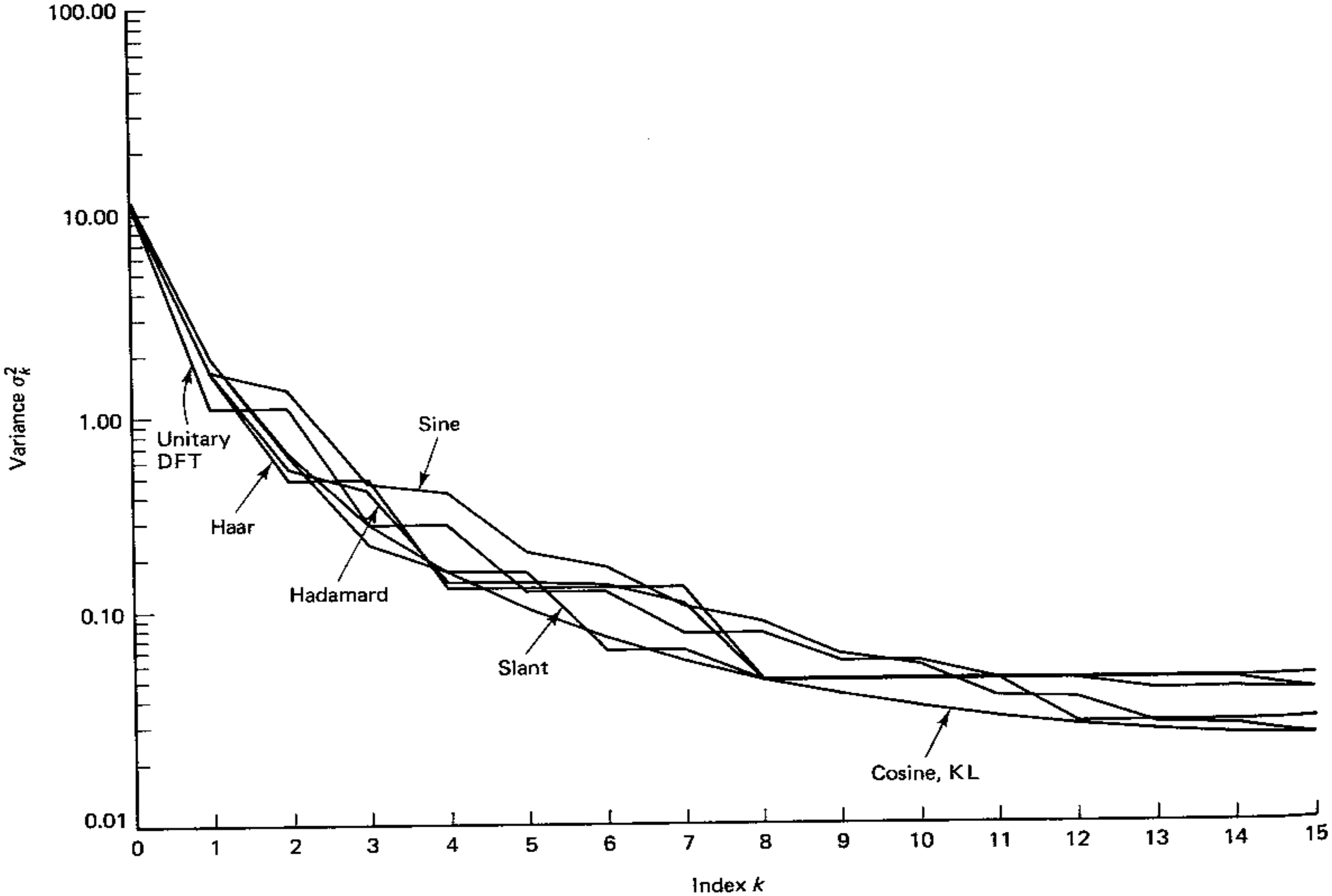
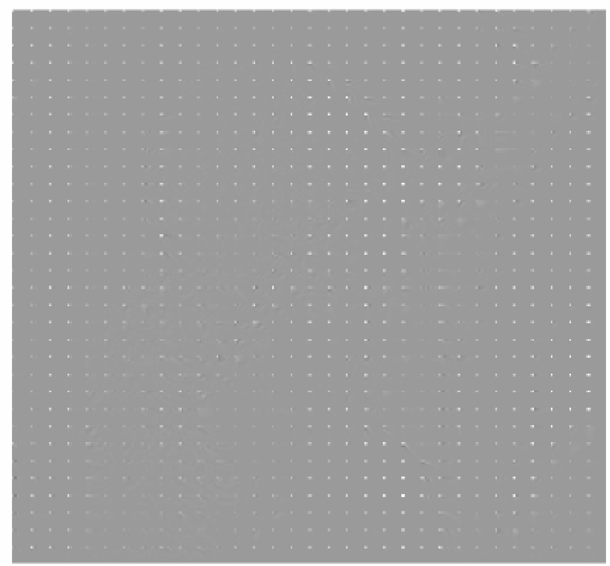


Figure 5.18 Distribution of variances of the transform coefficients (in decreasing order) of a stationary Markov sequence with $N = 16$, $\rho = 0.95$ (see Example 5.9).

implementation note: block transform

- similar to STFT (short-time Fourier transform)
  - partition a NxN image into mxn sub-images
  - save computation: O(N) instead of O(NlogN)
  - lose long-range correlation



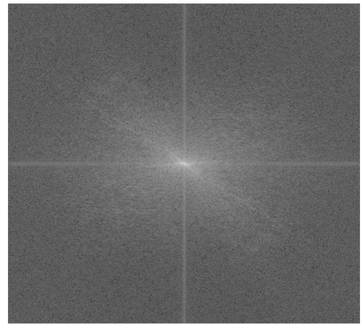
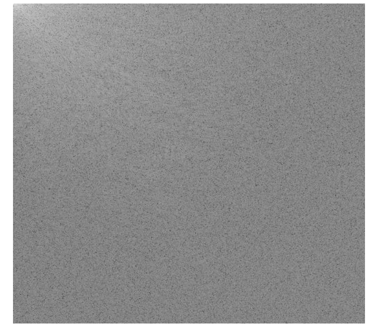
8x8 DCT coefficients

applications of transforms

- enhancement

- (non-universal) compression
- feature extraction and representation
- pattern recognition, e.g., eigen faces
- dimensionality reduction
  - analyze the principal ("dominating") components

Image Compression

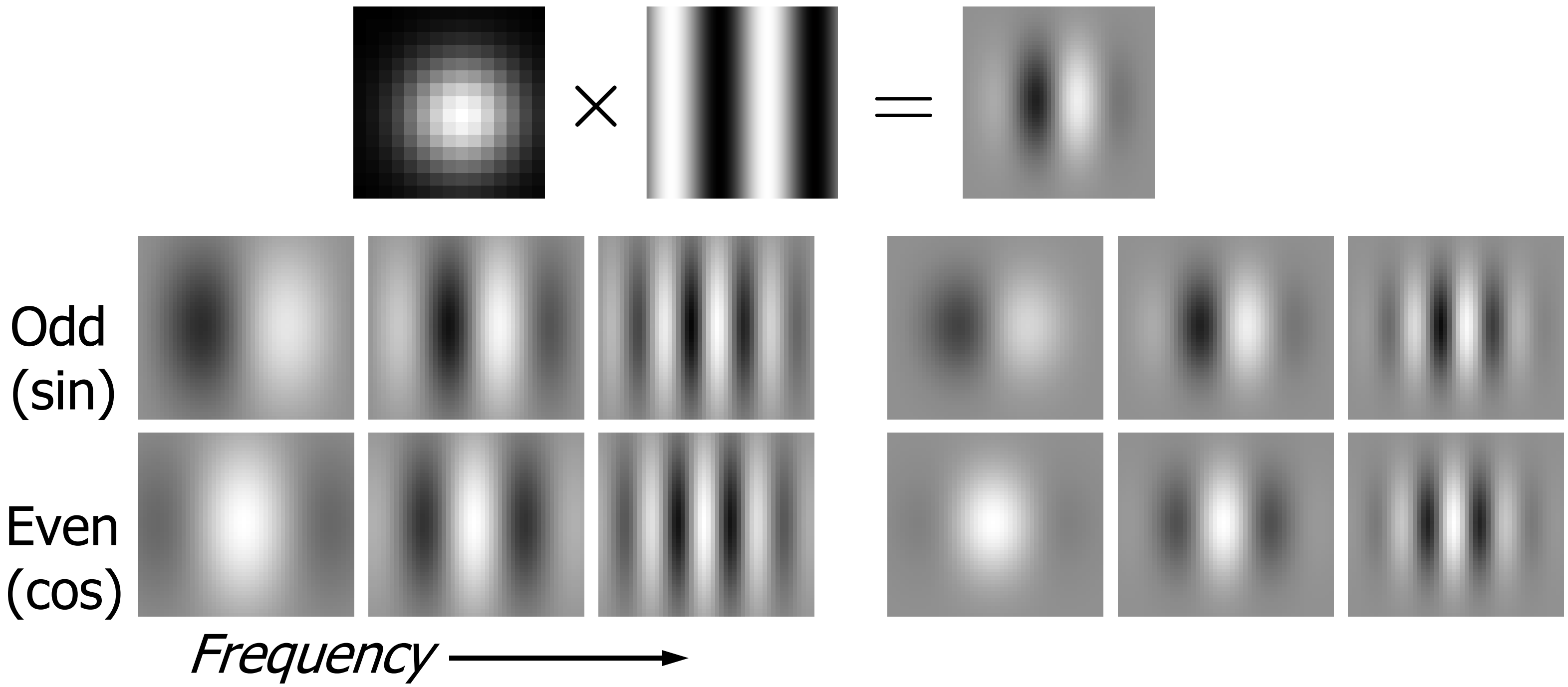
Measure compression quality with signal distortion:

$$\mathrm{SNR(dB)} = 10\log_{10}\left(\frac{P_{\mathrm{signal}}}{P_{\mathrm{noise}}}\right) = 20\log_{10}\left(\frac{A_{\mathrm{signal}}}{A_{\mathrm{noise}}}\right)$$

where P is average power and A is RMS amplitude.

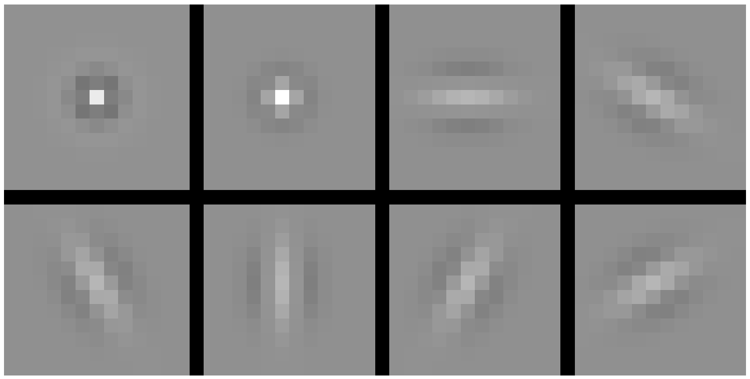
Gabor filters

- Gaussian windowed Fourier Transform
  - Make convolution kernels from product of Fourier basis images and Gaussians

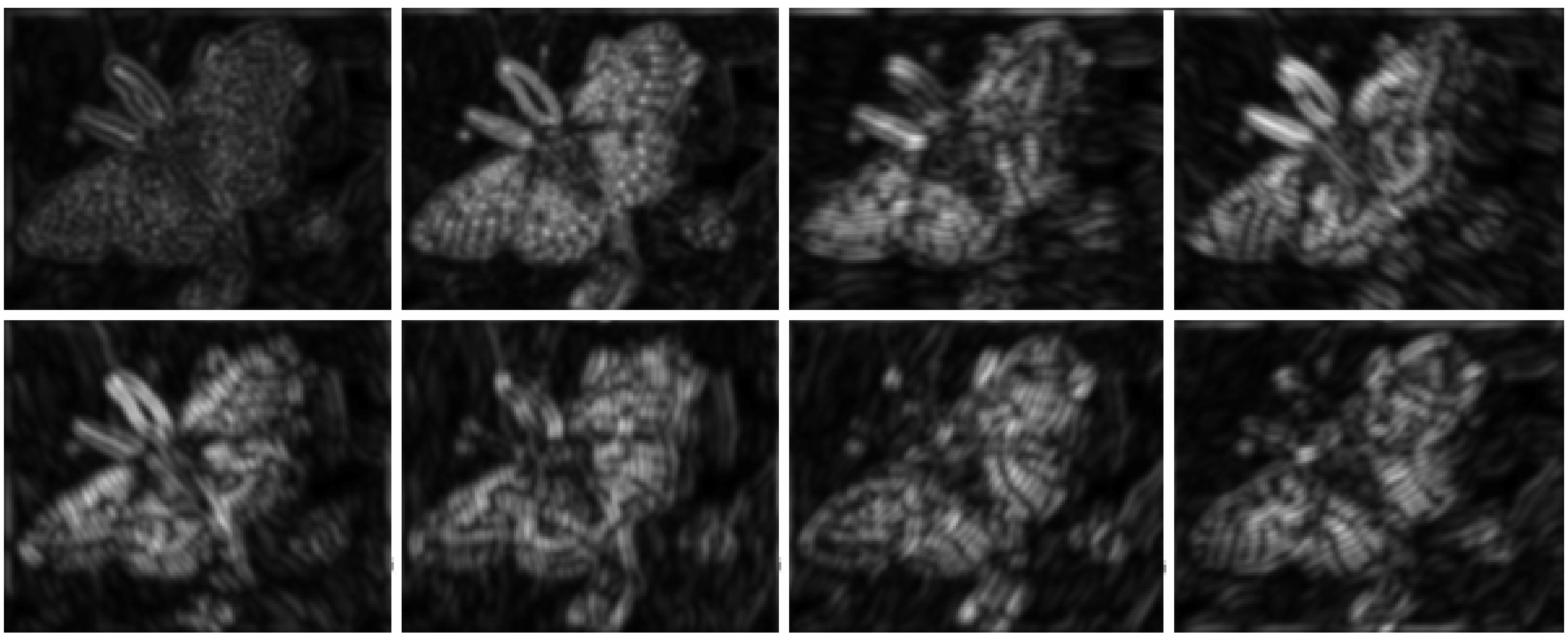


Example: Filter Responses

Filter bank



Input image

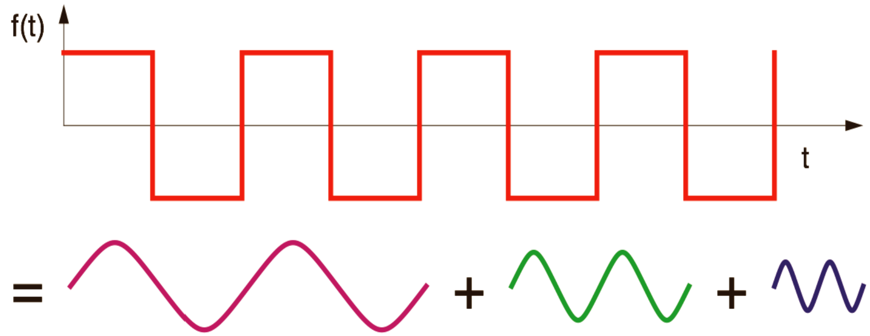


from Forsyth & Ponce

outline

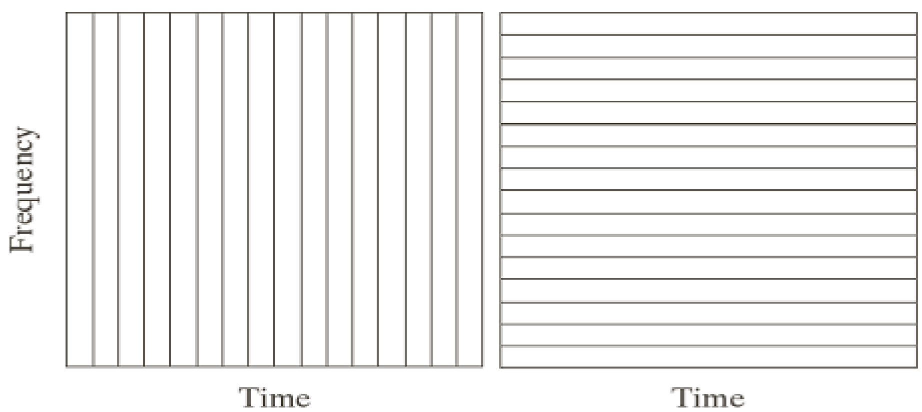
- Recap of DFT and DCT
- Unitary transforms
- KLT
- Other unitary transforms
- Multi-resolution and wavelets
- Applications

1807: Fourier upsets the French Academy....

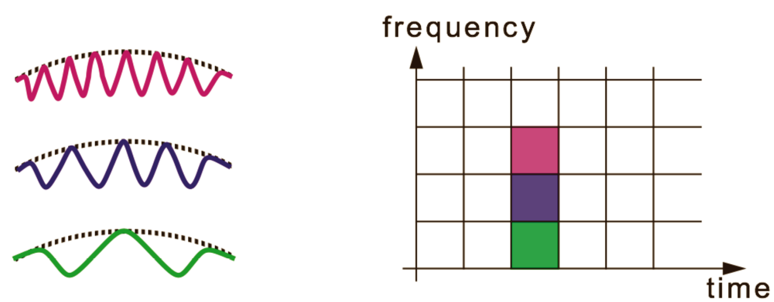


Fourier Series: Harmonic series, frequency changes, f_0, $2f_0$, $3f_0$, ...

sampling (dirac) FT

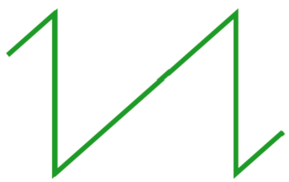


STFT



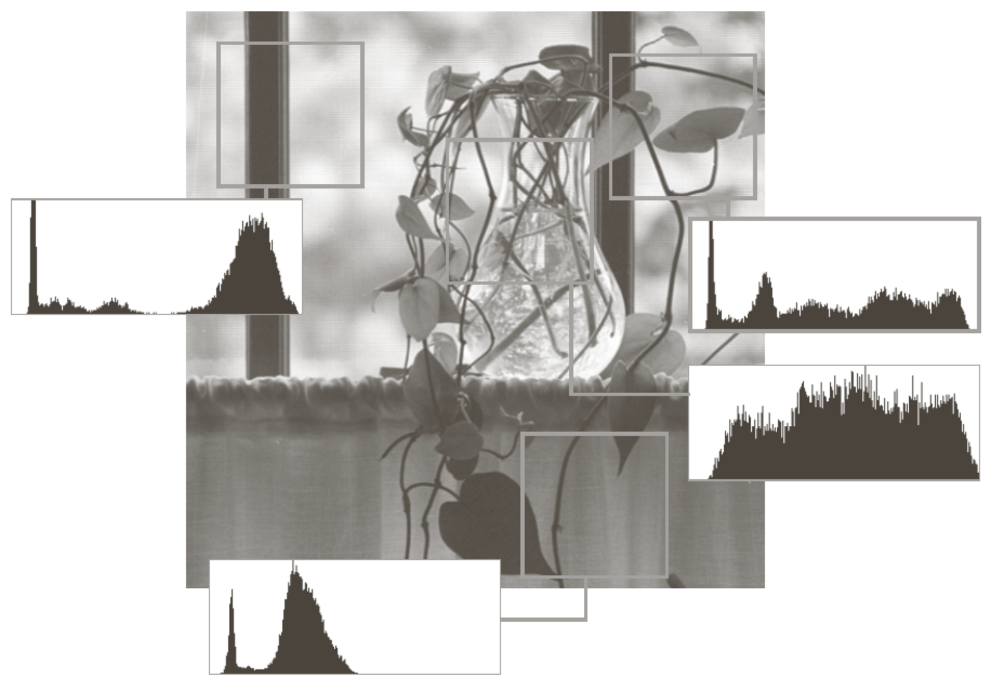
FT does not capture discontinuities well

But... 1898: Gibbs' paper

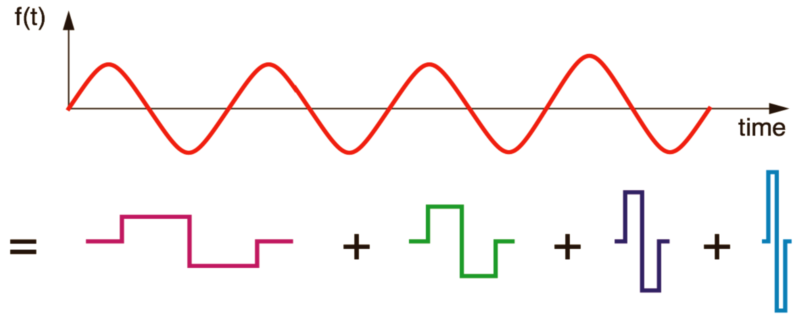


1899: Gibbs' correction

Orthogonality, convergence, complexity

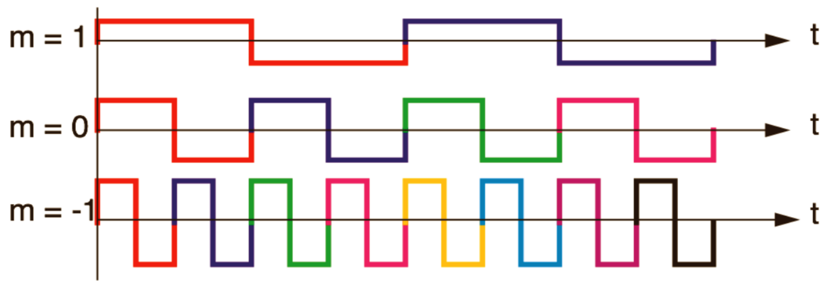


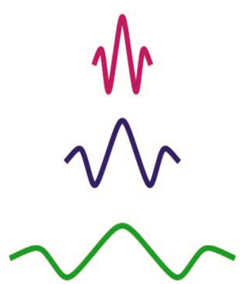
1910: Alfred Haar discovers the Haar wavelet
"dual" to the Fourier construction

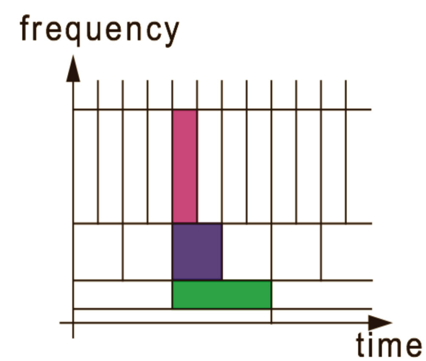


Haar series:

- Scale changes S_0, $2S_0$, $4S_0$, $8S_0$ ...
- orthogonality

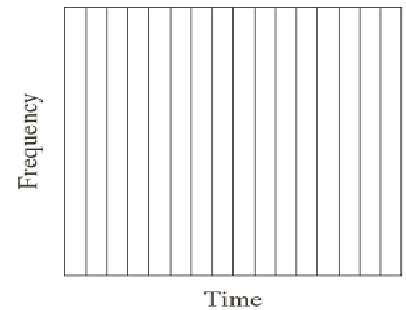


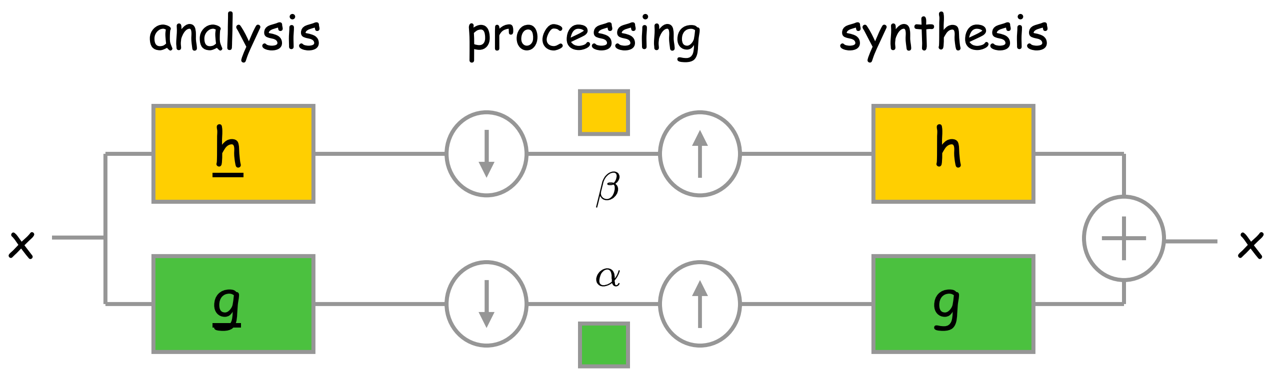


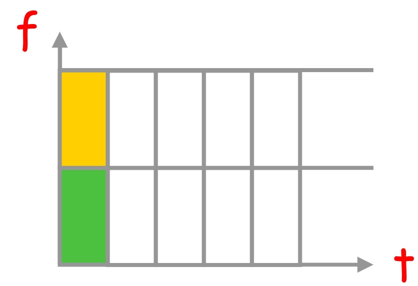


one step forward from dirac …

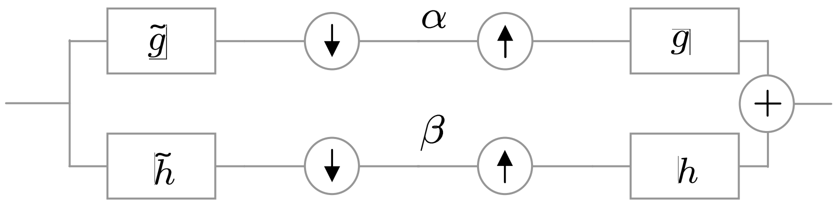
- Split the frequency in half means we can downsample by 2 to reconstruct upsample by 2.
- Filter to remove unwanted parts of the images and add
- Basic building block: Two-channel filter bank







orthogonal filter banks



1. Start from the reconstructed signal

$$x_{rec} = x_V + x_W = \sum_{k \in \mathbb{Z}} \alpha_k g_{n-2k} + \sum_{k \in \mathbb{Z}} \beta_k h_{n-2k}$$

$$= \begin{bmatrix} \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \\ \cdots & g_0 & h_0 & 0 & 0 & 0 & 0 & \cdots \\ \cdots & g_1 & h_1 & 0 & 0 & 0 & 0 & \cdots \\ \cdots & g_2 & h_2 & g_0 & h_0 & 0 & 0 & \cdots \\ \cdots & g_3 & h_3 & g_1 & h_1 & 0 & 0 & \cdots \\ \cdots & g_4 & h_4 & g_2 & h_2 & g_0 & h_0 & \cdots \\ \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} \vdots \\ \alpha_0 \\ \beta_0 \\ \alpha_1 \\ \beta_1 \\ \alpha_2 \\ \vdots \end{bmatrix} = \Phi X$$

- Read off the basis functions

$$\Phi = \{\varphi_k\}_{k \in \mathbb{Z}} = \{\varphi_{2k}, \varphi_{2k+1}\}_{k \in \mathbb{Z}} = \{g_{\cdot - 2k}, h_{\cdot - 2k}\}_{k \in \mathbb{Z}}$$

orthogonal filter banks

2. We want the expansion to be orthonormal $\Phi\Phi^T = I$
   - The output of the analysis bank is

$$X = \tilde{\Phi}^T x = \Phi^T$$

3. Then
   - The rows of Φ^T are the basis functions $\{g_{\cdot-2k}, h_{\cdot-2k}\}_{k\in\mathbb{Z}}$
   - The rows of Φ^T are the reversed versions of the filters

$$\begin{aligned} \alpha_k &= \langle g_{\cdot-2k}, x\rangle = (g_{-n} * x_n)_{2k} \quad \Leftrightarrow \quad \alpha = \Phi_g^T x, \\ \beta_k &= \langle h_{\cdot-2k}, x\rangle = (h_{-n} * x_n)_{2k} \quad \Leftrightarrow \quad \beta = \Phi_h^T x. \end{aligned}$$

   - The analysis filters are

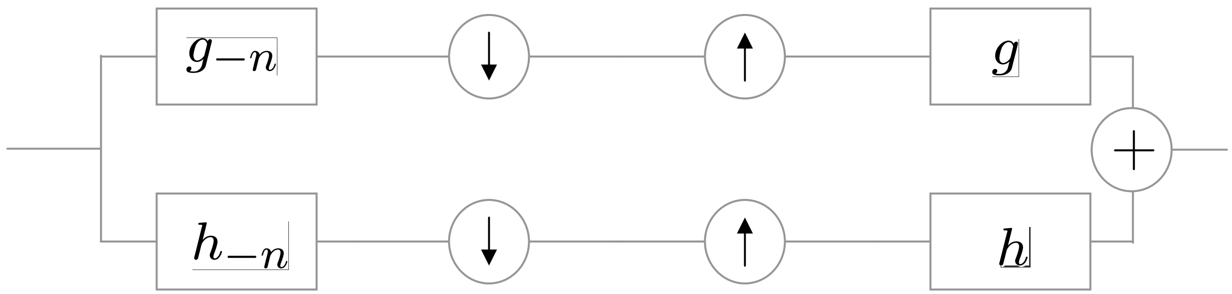
$$\tilde{g}_n = g_{-n}, \qquad \tilde{h}_n = h_{-n}$$

orthogonal filter banks

4. Since Φ is unitary, basis functions are orthonormal

$$\begin{aligned} \langle g_{\cdot-2k}, g \rangle &= \delta_k, \\ \langle h_{\cdot-2k}, h \rangle &= \delta_k, \\ \langle h_{\cdot-2k}, g \rangle &= 0. \end{aligned}$$

5. Final filter bank



orthogonal filter banks: Haar basis

Solve for the filter h explicitly.

$$g_n = \frac{1}{\sqrt{2}}(\delta_n + \delta_{n-1}).$$

Given that h_n must be of norm 1 and of same the length as g_n,

$$h_n = (\cos\alpha)\delta_n + (\sin\alpha)\delta_{n-1}.$$

Computing the inner product $\langle h_{\cdot-2k}, g\rangle = 0$:

$$\frac{1}{\sqrt{2}}(\cos\alpha + \sin\alpha) = 0.$$

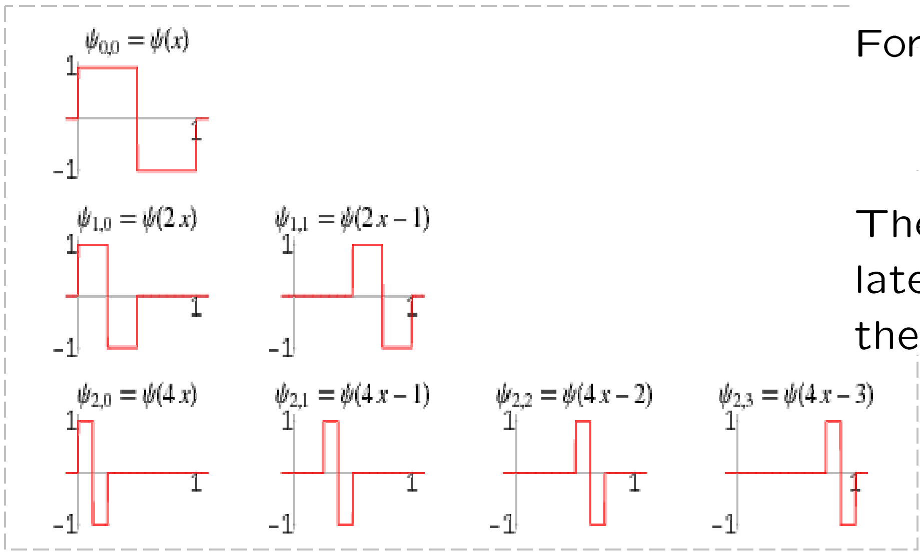
The solution to the above is:

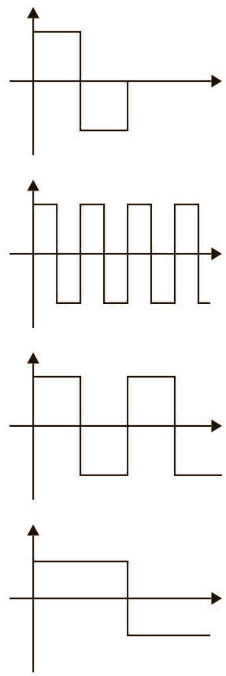
$$\sin\alpha = -\cos\alpha \qquad \Rightarrow \qquad \alpha = k\pi - \frac{\pi}{4}.$$

For $k = 0$, a solution to h_n is:

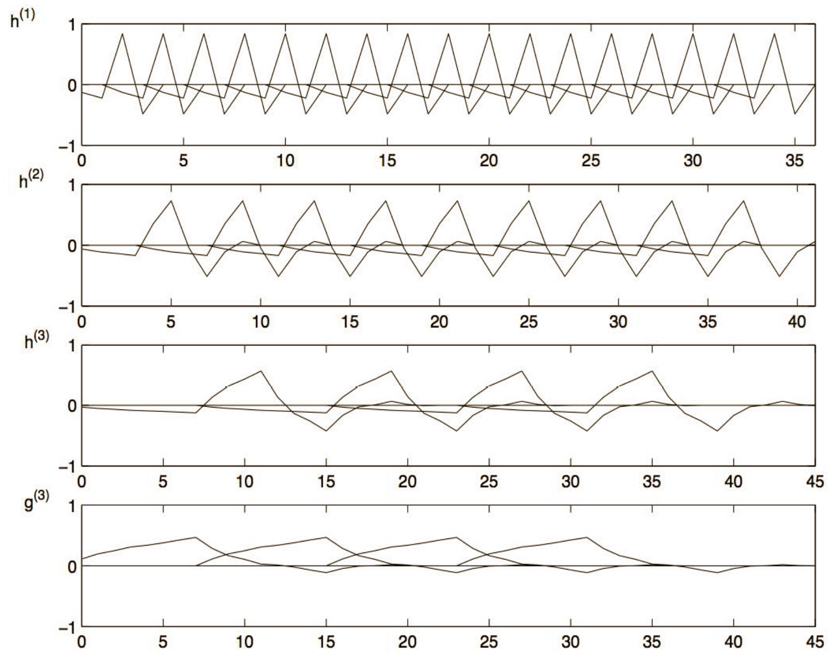
$$h_n = \frac{1}{\sqrt{2}}(\delta_n - \delta_{n-1}).$$

The above pair and their even translates translates constitute an ONB for $\ell^2(\mathbb{Z})$ and are called the *Haar filter pair*.



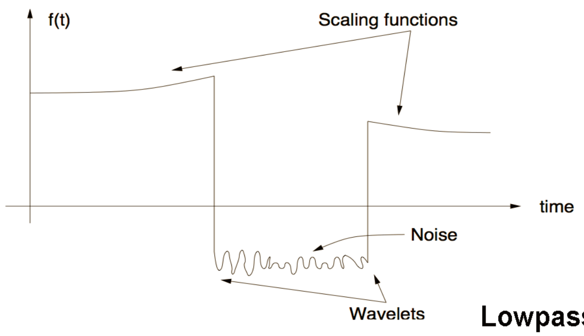


Haar



Daubechies, D_2

Goal: efficient representation of signals like



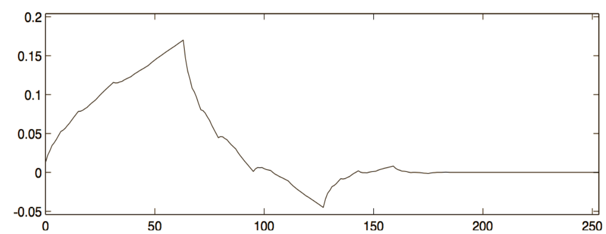
where:

- Wavelet act as singularity detectors
- Scaling functions catch smooth parts
- "Noise" is circularly symmetric

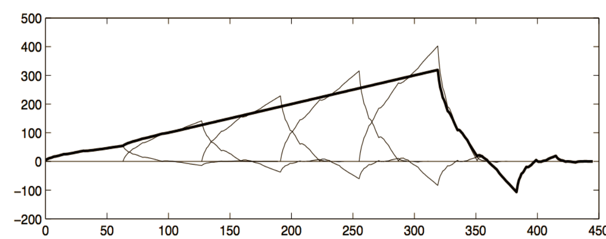
Note: Fourier gets all Gibbs-ed up!

Lowpass filters and scaling functions reproduce polynomials

- Iterate of Daubechies L=4 lowpass filter reproduces linear ramp



scaling function

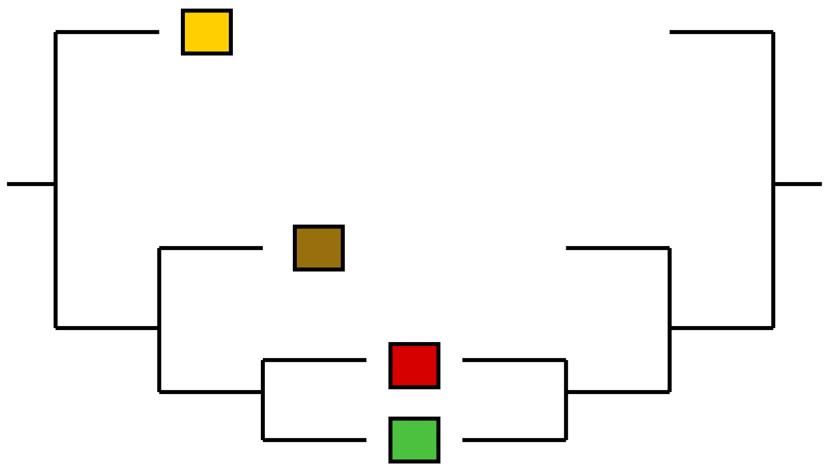


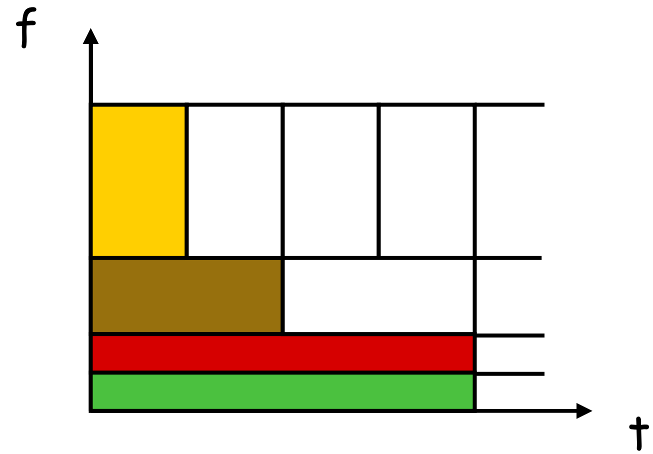
linear ramp

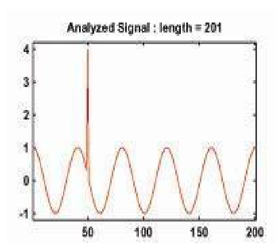
Scaling functions catch "trends" in signals

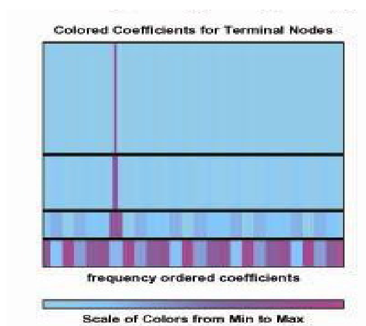
DWT

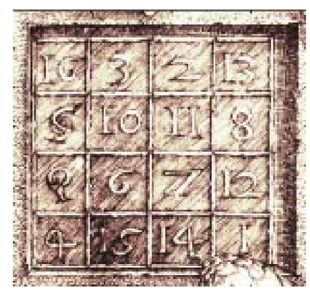
- Iterate only on the lowpass channel

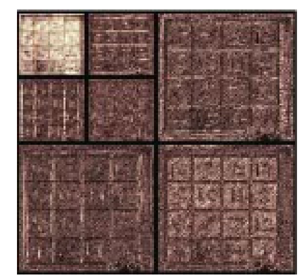






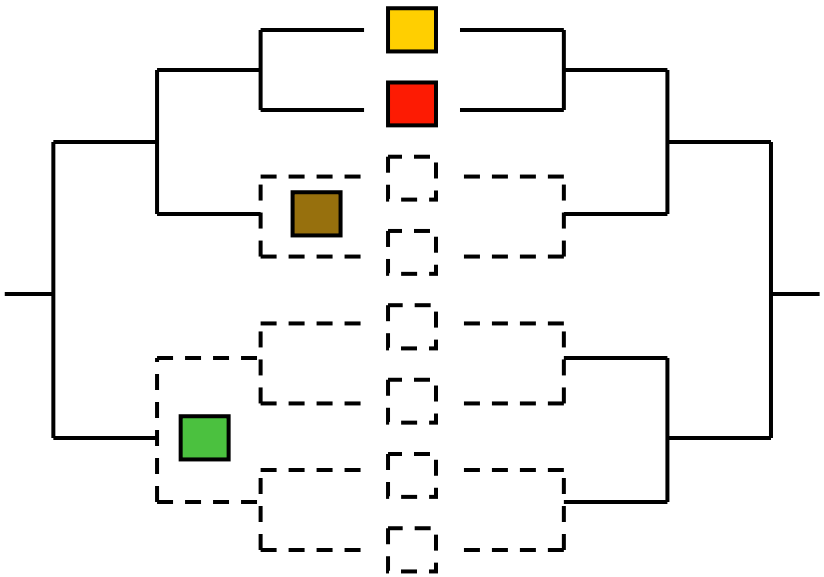


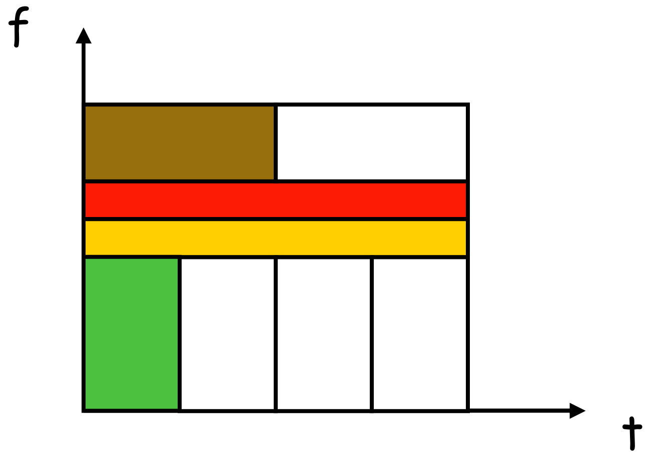




wavelet packet

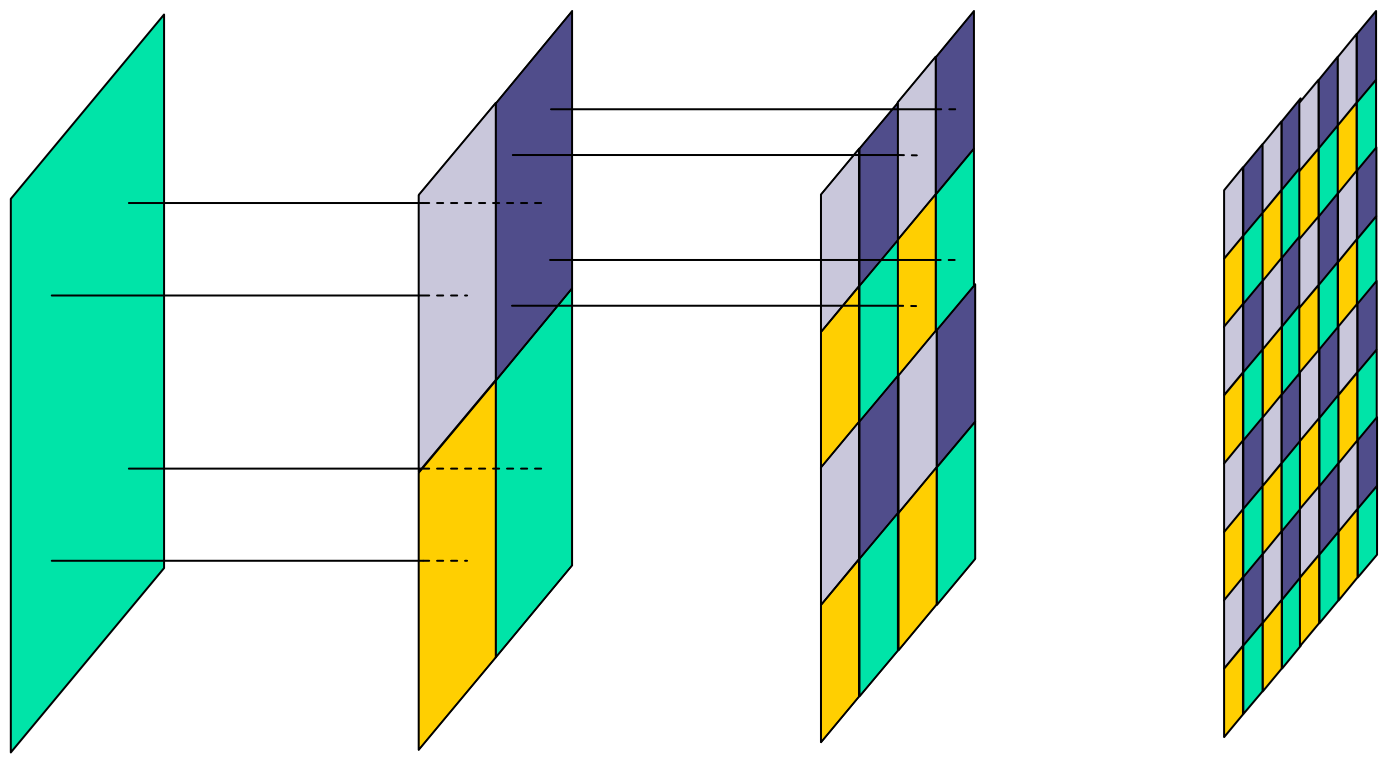
- Iterate on both the low pass and (selected) high-pass channels





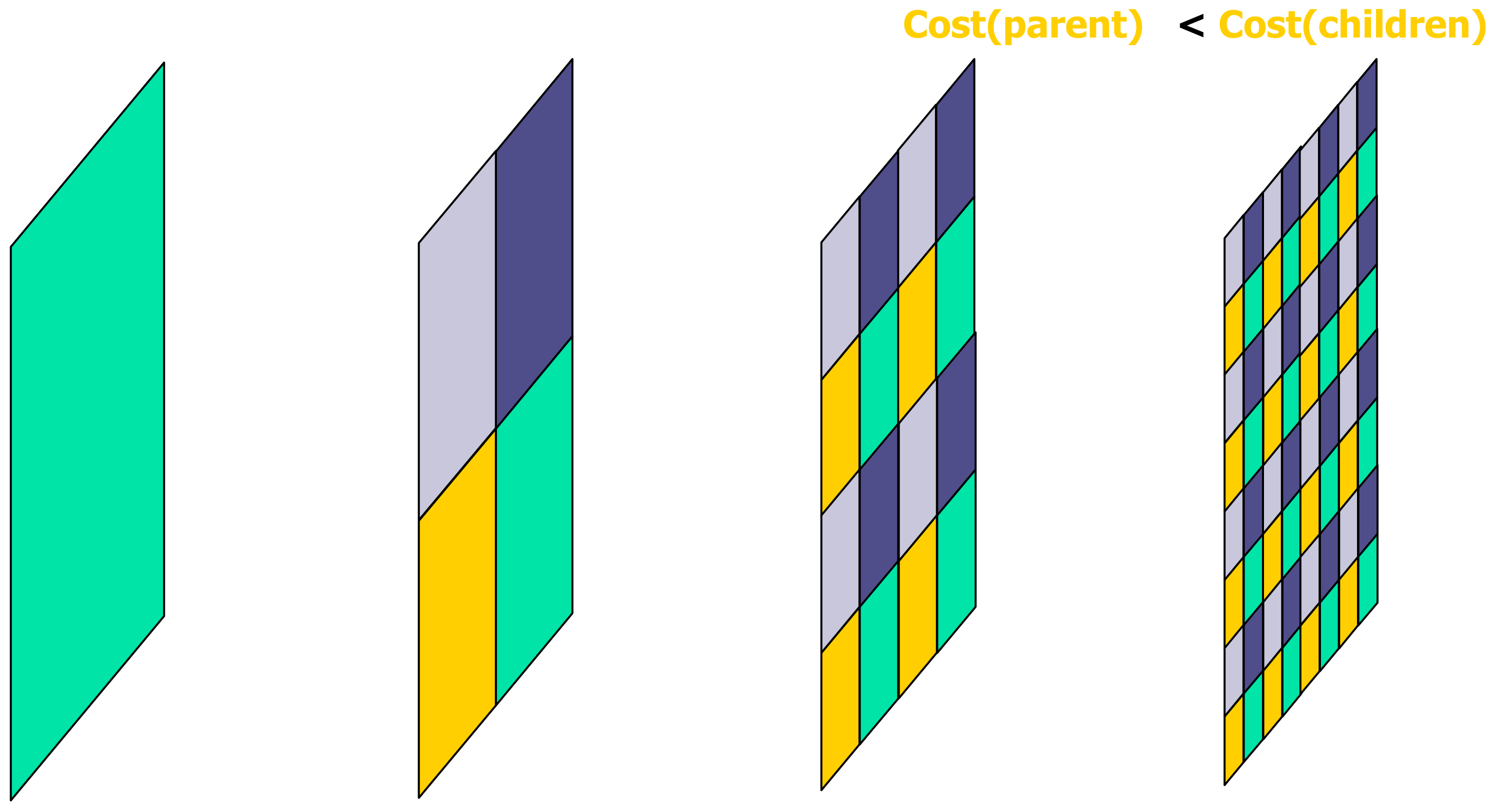
wavelet packet

- First stage: full decomposition



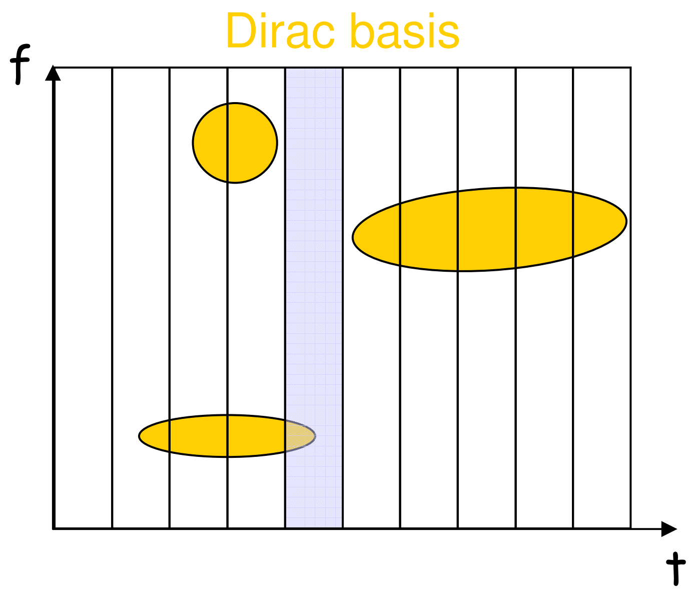
wavelet packet

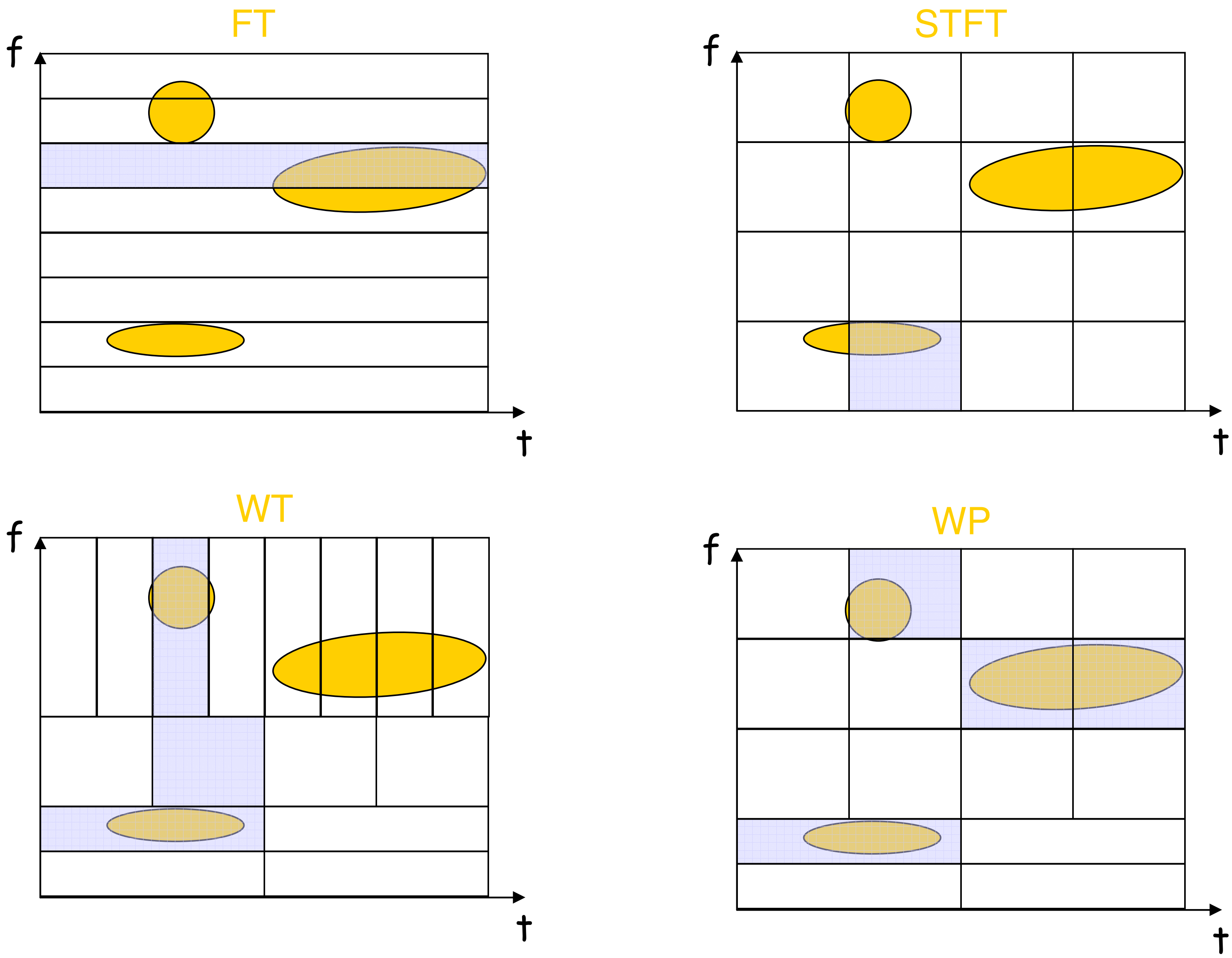
- Second stage: pruning



wavelet packet: why it works

- One of the grand challenges in signal analysis and processing is in understanding “blob”-like structures of the energy distribution in the time-frequency space, and designing a representation to reflect those.




FT
f
t
STFT
f
t
WT
f
t
WP
f
t

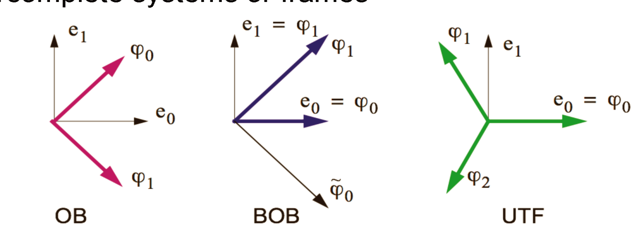
- are we solving x=x?
  - sort of: find matrices such that $x = Ix = \Phi\tilde{\Phi}^* x$
  - after finding those
    - Decomposition $X = \tilde{\Phi}^* x$
    - Reconstruction $x = \Phi X = \Phi\tilde{\Phi}^* x$

- in a nutshell
  - if Φ is square and nonsingular, Φ is a basis and $\tilde{\Phi}$ is its dual basis
  - if Φ is unitary, that is, $\Phi\ \Phi^* = I$, Φ is an orthonormal basis and $\tilde{\Phi} = \Phi$
  - if Φ is rectangular and full rank, Φ is a frame and $\tilde{\Phi}$ is its dual frame
  - if Φ is rectangular and $\Phi\ \Phi^* = I$, Φ is a tight frame and $\tilde{\Phi} = \Phi$

overview of multi-resolution techniques

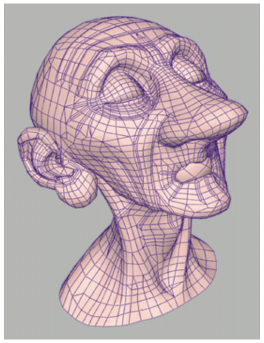
Property	Orthonormal Basis	Biorthogonal Basis	Tight Frame	General Frame
Expansion Set	$\Phi = \{\varphi_i\}_{i=1}^n$ $\varphi_i \in \mathbb{C}^n$	$\Phi = \{\varphi_i\}_{i=1}^n$ $\tilde{\Phi} = \{\tilde{\varphi}_i\}_{i=1}^n$ $\varphi_i \in \mathbb{C}^n, \tilde{\varphi}_i \in \mathbb{C}^n$	$\Phi = \{\varphi_i\}_{i=1}^m$ $\varphi_i \in \mathbb{C}^n,\ m \geq n$	$\Phi = \{\varphi_i\}_{i=1}^m$ $\tilde{\Phi} = \{\tilde{\varphi}_i\}_{i=1}^m$ $\varphi_i \in \mathbb{C}^n, \tilde{\varphi}_i \in \mathbb{C}^n,\ m \geq n$
Self-Dual	Yes	No	Yes	No
Linearly Independent	Yes	Yes	No	No
Orthogonality Relations	$\langle \varphi_i, \varphi_j \rangle = \delta_{i-j}$	$\langle \varphi_i, \tilde{\varphi}_j \rangle = \delta_{i-j}$	None	None
Expansion	$x = \sum_{i=1}^n \langle \varphi_i, x \rangle \varphi_i$	$x = \sum_{i=1}^n \langle \tilde{\varphi}_i, x \rangle \varphi_i$	$x = \sum_{i=1}^m \langle \varphi_i, x \rangle \varphi_i$	$x = \sum_{i=1}^m \langle \tilde{\varphi}_i, x \rangle \varphi_i$
Matrix Representation	Φ of size $n \times n$ Φ unitary $\Phi\Phi^T = \Phi^T\Phi = I$	Φ of size $n \times n$ Φ full rank $\Phi\tilde{\Phi}^T = I,\ \tilde{\Phi} = (\Phi^T)^{-1}$	Φ of size $n \times m$ rows of Φ orthogonal $\Phi\Phi^T = I$	Φ of size $n \times m$ Φ full rank $\Phi\tilde{\Phi}^T = I$
Norm Preservation	Yes $\|x\|^2 = \sum_{i=1}^n \|\langle x, \varphi_i \rangle\|^2$	No	Yes [illegible]	No
Successive Approximation	Yes $\hat{x}^{(k)} = \hat{x}^{(k-1)} + \langle x, \varphi_k \rangle \varphi_k$	No		
Redundant	No	No		

- orthonormal bases (e.g. Fourier series, wavelet series)
- biorthogonal bases
- overcomplete systems or frames

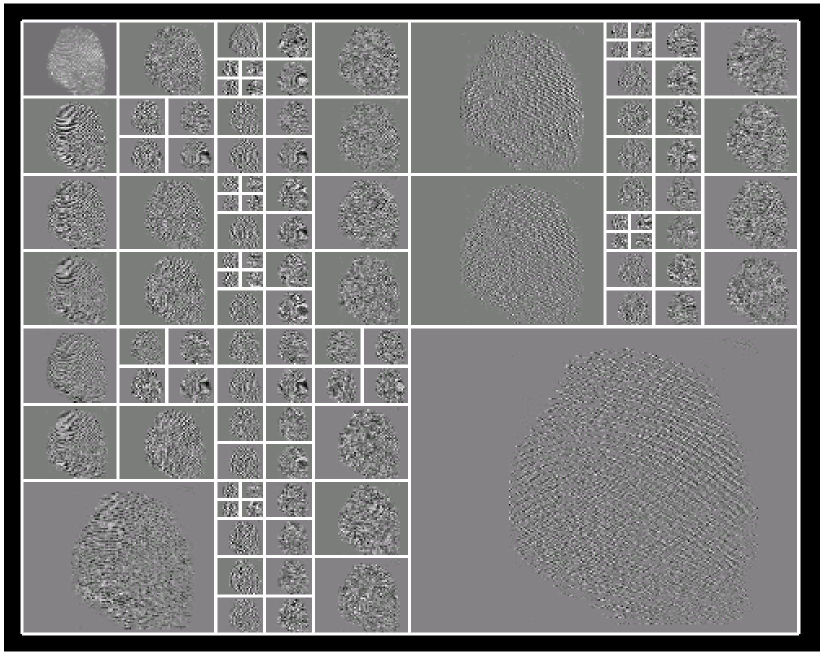


applications of wavelets

- enhancement and denoising
- compression and MR approximation
- fingerprint representation with wavelet packets
- bio-medical image classification
- subdivision surfaces “Geri’s Game”, “A Bug’s Life”, “Toy Story 2”

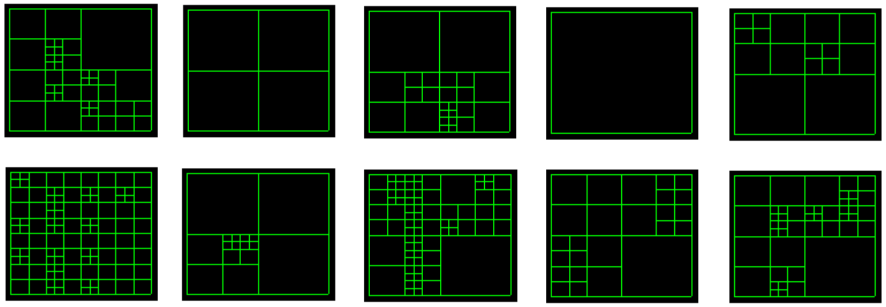


fingerprint feature extraction

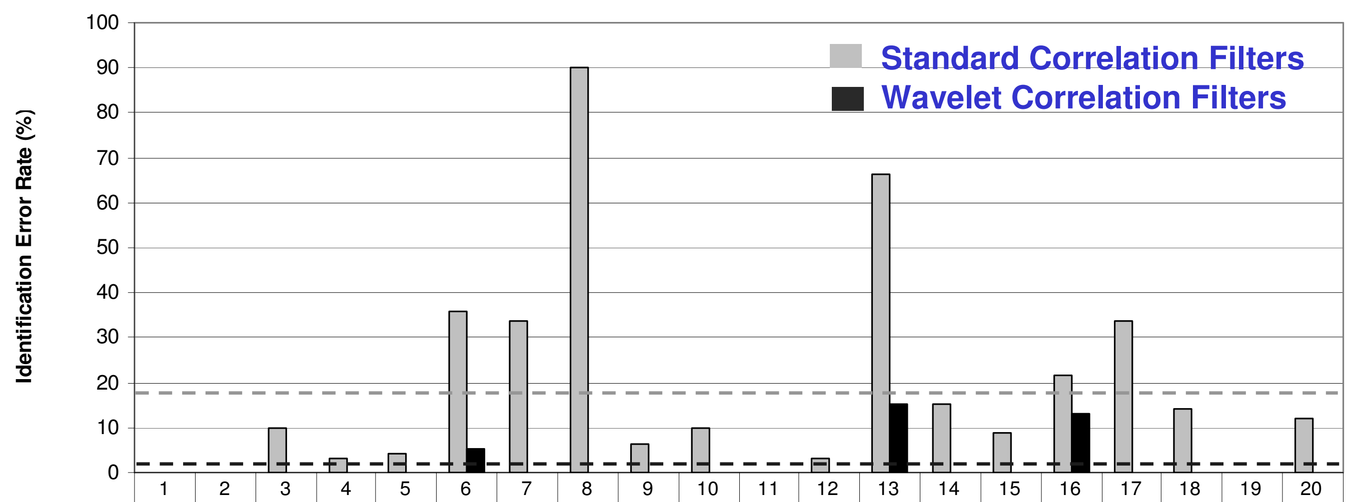


- MR system
  - Introduces adaptivity
  - Template matching performed on different space-frequency regions
  - Builds a different decomposition for each class

$$F(\text{parent}) > F(\text{child 1}) + F(\text{child 2}) + F(\text{child 3}) + F(\text{child 4}), \qquad F = \frac{1}{E}$$



fingerprint identification results



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
SCF	0	0	9.78	3.26	4.35	35.90	33.70	89.96	6.52	9.78	0	3.26	66.30	15.22	8.70	21.74	33.70	14.13	0	11.96
WDCF	0	0	0	0	0	5.43	0	0	0	0	0	0	15.22	0	0	13.04	0	0	0	0

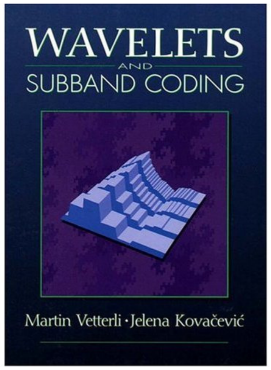
SCF Average IER = 18.41% **WDCF Average IER = 1.68%**

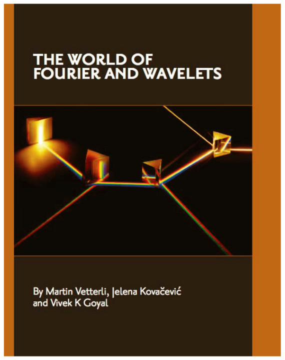
NIST 24 fingerprint database

- 10 people (5 male & 5 female), 2 fingers
- 20 classes, 100 images/class

references for multiresolution

- Light reading
  - "Wavelets: Seeing the Forest -- and the Trees", D. Mackenzie, Beyond Discovery, December 2001.
- Overviews
  - D.Donoho, M.Vetterli, R.DeVore and I.Daubechies, Data Compression and Harmonic Analysis, IEEE Tr. on IT, Oct.1998.
  - M. Vetterli, Wavelets, approximation and compression, IEEE Signal Processing Magazine, Sept. 2001
- Books
  - "Wavelets and Subband Coding", M. Vetterli and J. Kovacevic, Prentice Hall, 1995.
  - "A Wavelet Tour of Signal Processing", S. Mallat, Academic Press, 1999.
  - "Ten Lectures on Wavelets", I. Daubechies, SIAM, 1992.
  - "Wavelets and Filter Banks", G. Strang and T. Nguyen, Wells. Cambr. Press, 1996.





summary

- unitary transforms
  - theory revisited
  - the quest for optimal transform
  - example transforms
    DFT, DCT, KLT, Hadamard, Slant, Haar, …
- multi-resolution analysis and wavelets
- applications
  - compression
  - feature extraction and representation
  - image matching (digits, faces, fingerprints)

Timeline

Wavelets have had an unusual scientific history, marked by many independent discoveries and rediscoveries. The most rapid progress has come since the early 1980s, when a coherent mathematical theory of wavelets finally emerged.

1807
Jean Baptiste Joseph Fourier claims that any periodic function, or wave, can be expressed as an infinite sum of sine and cosine waves of various frequencies. Because of serious doubts over the correctness of his arguments, his paper is not published until 15 years later.

1909
Alfred Haar, a Hungarian mathematician, discovers a "basis" of functions that are now recognized as the first wavelets. They consist of a short positive pulse followed by a short negative pulse.

1930
John Littlewood and R.A.E.C. Paley, of Cambridge University, show that local information about a wave, such as the timing of a pulse of energy, can be retrieved by grouping the terms of its Fourier series into "octaves."

1946
Dennis (Denes) Gabor, a British-Hungarian physicist who invented holography, decomposes signals into "time-frequency packets" or "Gabor chirps."

1976
IBM physicists Claude Galand and Daniel Esteban discover subband coding, a way of encoding digital transmissions for the telephone.

1981
Petroleum engineer Jean Morlet of Elf-Aquitaine finds a way to decompose seismic signals into what he calls "wavelets of constant shape." He turns to quantum physicist Alex Grossmann for help in proving that the method works.

1984
Joint paper by Morlet and Grossmann brings the word "wavelet" into the mathematical lexicon for the first time.

10 yrs

1985
Yves Meyer of the University of Paris discovers the first smooth orthogonal wavelets.

1986
Stéphane Mallat, then at the University of Pennsylvania, shows that the Haar basis, the Littlewood-Paley octaves, the Gabor chirps, and the subband filters of Galand and Esteban are all related to wavelet-based algorithms.

1987
Ingrid Daubechies constructs the first smooth orthogonal wavelets with compact support. Her wavelets turn the theory into a practical tool that can be easily programmed and used by any scientist with a minimum of mathematical training.

1990
David Donoho and Iain Johnstone, at Stanford University, use wavelets to "denoise" images, making them even sharper than the originals.

1992
The FBI chooses a wavelet method developed by Tom Hopper of the FBI's Criminal Justice Information Services Division and Jonathan Bradley and Chris Brislawn from Los Alamos National Laboratory, to compress its enormous database of fingerprints.

1995
Pixar Studios releases the movie *Toy Story*, the first fully computer-animated cartoon. In the sequel, *Toy Story 2*, some shapes are rendered by subdivision surfaces, a technique mathematically related to wavelets.

1999
The International Standards Organization approves a new standard for digital picture compression, called **JPEG-2000**. The new standard uses wavelets to compress image files by **1:200** ratios with no visible loss in image quality. Web browsers are expected to support the new standard by **2001**.

1 yr