Image Processing

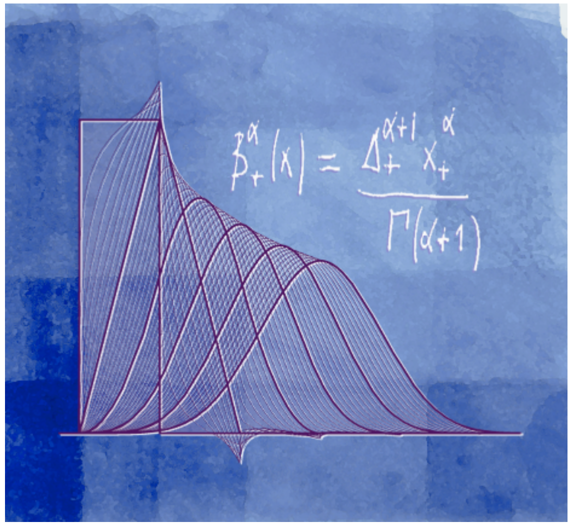
Chapter 8

Image transforms

Prof. Michael Unser, LIB

Prof. Dimitri Van De Ville, MIP

Dr. Daniel Sage, LIB



March-April 2025

CONTENT

Transform-domain processing

- Abstract Hilbert-space formulation

  Image space: $\mathcal{H} = \mathbb{R}^N$ or $\mathcal{H} = \ell_2(\mathbb{Z}^2)$

  $\boldsymbol{x}$: image vector

  $\boldsymbol{y}$: transformed image

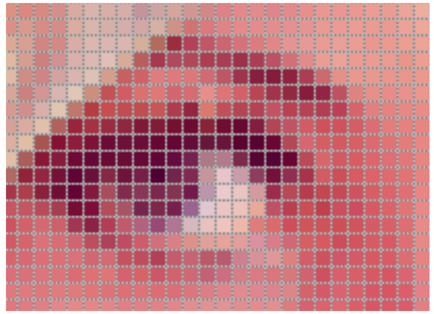
  - Invertible linear operator

    $$\boldsymbol{y} = \boldsymbol{A} \cdot \boldsymbol{x} \quad \Leftrightarrow \quad \boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{y}$$

  - General transform requirements
    - Fast transform (FFT-like algorithm)
    - Orthonormality

    $$\boldsymbol{A}^{-1} = \boldsymbol{A}^H \quad \Leftrightarrow \quad \|\boldsymbol{x}\| = \|\boldsymbol{y}\|$$

- Constructing the image vector



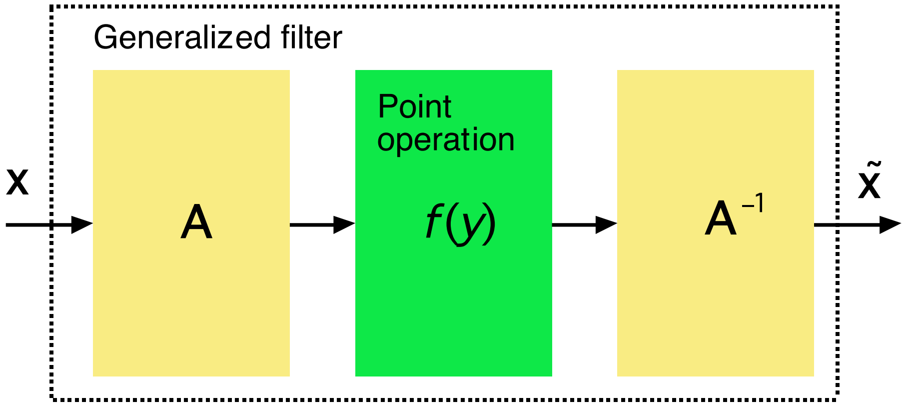
$\{f[k,l]\}$

$$\boldsymbol{x} = \begin{bmatrix} f[0,0] \\ f[1,0] \\ \vdots \\ f[K-1,0] \\ f[0,1] \\ \vdots \\ \vdots \\ \vdots \end{bmatrix} \begin{array}{l} \left.\begin{array}{l} \\ \\ \\ \\ \end{array}\right\} \text{1st row} \\ \left.\begin{array}{l} \\ \\ \\ \end{array}\right\} \text{2nd row} \\ \\ \\ \end{array}$$

Image transforms: applications

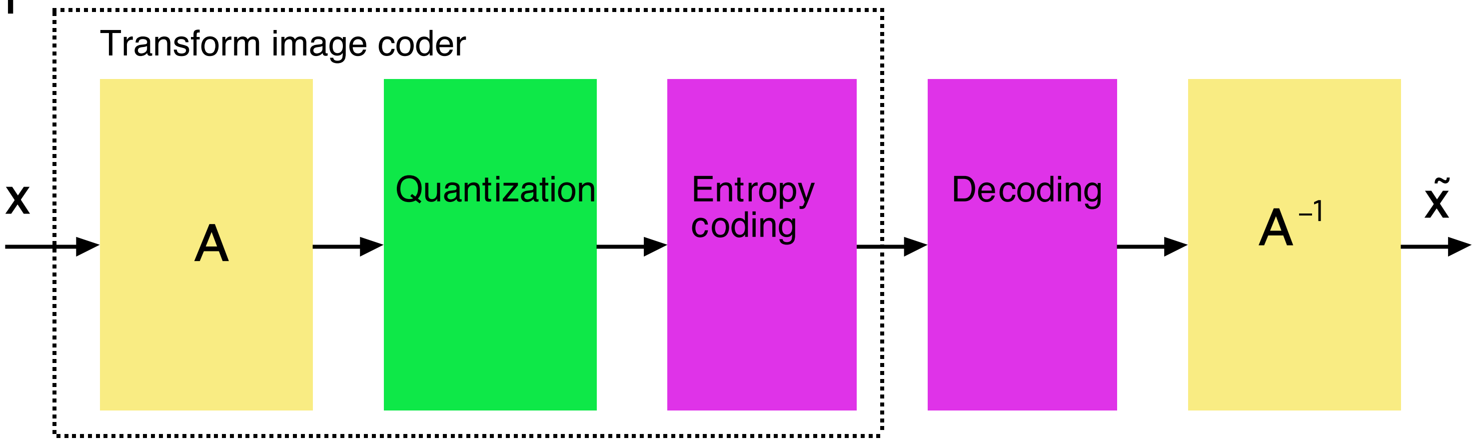
- **Data processing**

Examples
- image enhancement
- denoising
- Wiener filter



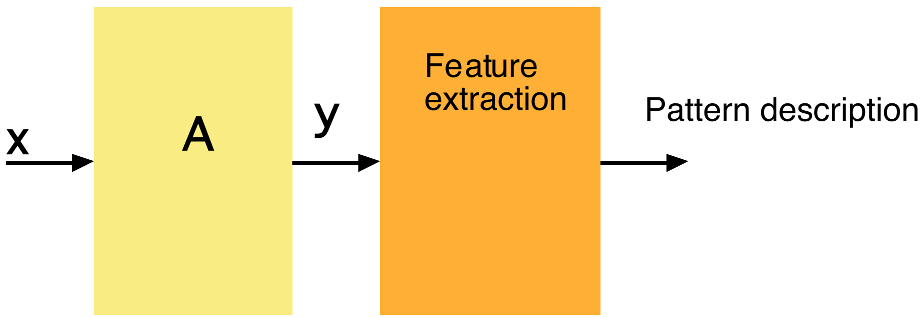
- **Data compression**

Examples of coders
- JPEG
- MPEG
- EZW



- **Data analysis**

Examples
- texture analysis
- feature detection

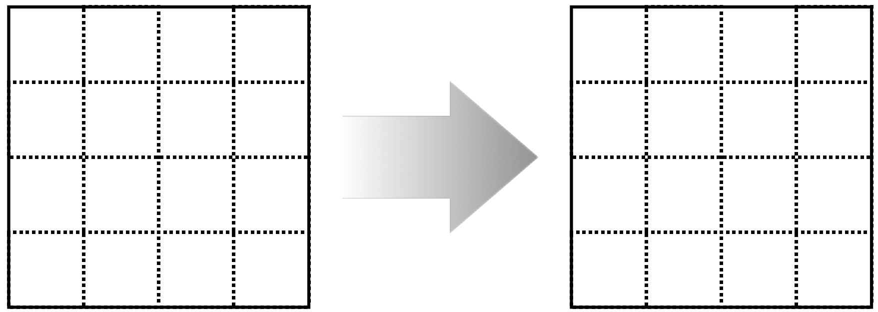


Transform types

Block-by-block versus global

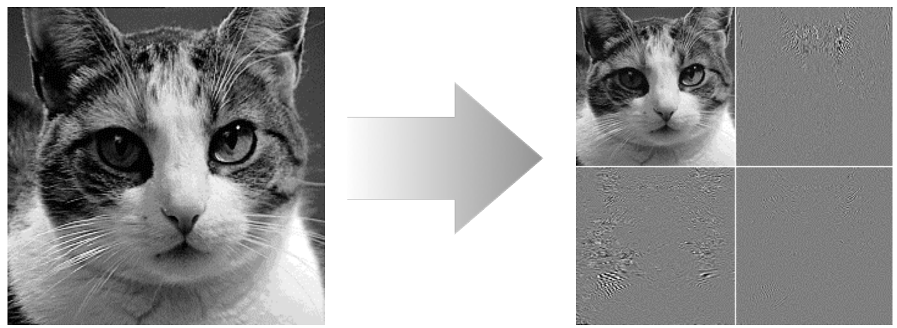
- Block transforms

  Example: 8 × 8 DCT



- Global transforms

  Example: wavelet transform



One-to-one versus redundant

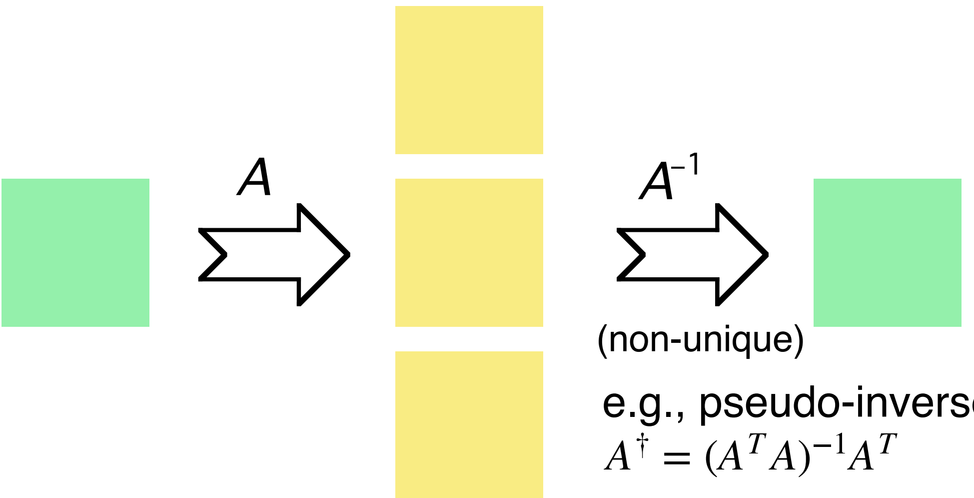
- Decomposition using basis functions

"basis" ⟷ square transform matrix

Preferred application: coding

- Decomposition using frames

"Frame" ⟷ rectangular transform matrix



e.g., pseudo-inverse:
$A^{\dagger} = (A^T A)^{-1} A^T$

Preferred applications
- generalized filtering
- data analysis

Review of linear algebra

$\mathcal{H} = \mathbb{C}^N$

- Signal/image vector: $\boldsymbol{x} = (x_1, \ldots, x_N) = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} \in \mathbb{C}^N$

  Representation in the canonical basis: $\boldsymbol{x} = \sum_{i=1}^{N} x_i \boldsymbol{e}_i$ with $[\boldsymbol{e}_i]_j = \delta_{i-j}$

  $\boldsymbol{e}_i = (0, 0, \ldots, 1, 0, \ldots, 0)$

- Inner-product in $\mathbb{C}^N$: $\langle \boldsymbol{u}, \boldsymbol{v} \rangle = \sum_{i=1}^{N} u_i^* v_i = \boldsymbol{u}^H \boldsymbol{v}$

- Hermitian transpose: $\boldsymbol{A}^H = \left(\boldsymbol{A}^T\right)^* = (\boldsymbol{A}^*)^T$, $\left[\boldsymbol{A}^H\right]_{ij} = [\boldsymbol{A}]_{ji}^*$

- Unitary matrices

  Definition: the $N \times N$ matrix $\boldsymbol{U}$ is unitary iff. $\boldsymbol{U}^{-1} = \boldsymbol{U}^H$

  Example: $\boldsymbol{U} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$

Basis vectors

The vectors $\boldsymbol{u}_1, \boldsymbol{u}_2, \dots, \boldsymbol{u}_N \in \mathbb{C}^N$ form a basis of $\mathbb{C}^N$ iff. they are linearly independent

$$a_1\boldsymbol{u}_1 + \cdots + a_N\boldsymbol{u}_N = \boldsymbol{0} \qquad \Leftrightarrow \qquad a_1 = \cdots = a_N = 0$$

$$\Updownarrow$$

The $N \times N$ matrix $\boldsymbol{U} = \begin{bmatrix} \boldsymbol{u}_1 & \boldsymbol{u}_2 & \cdots & \boldsymbol{u}_N \end{bmatrix}$ is invertible

$$\Updownarrow$$

$\boldsymbol{G}_u = \boldsymbol{U}^H\boldsymbol{U}$ is positive-definite

- Gram matrix: $\quad \boldsymbol{G}_u = \begin{bmatrix} \langle \boldsymbol{u}_1, \boldsymbol{u}_1 \rangle & \cdots & \langle \boldsymbol{u}_1, \boldsymbol{u}_N \rangle \\ \vdots & \ddots & \vdots \\ \langle \boldsymbol{u}_N, \boldsymbol{u}_1 \rangle & \cdots & \langle \boldsymbol{u}_N, \boldsymbol{u}_N \rangle \end{bmatrix} = \boldsymbol{U}^H\boldsymbol{U}$

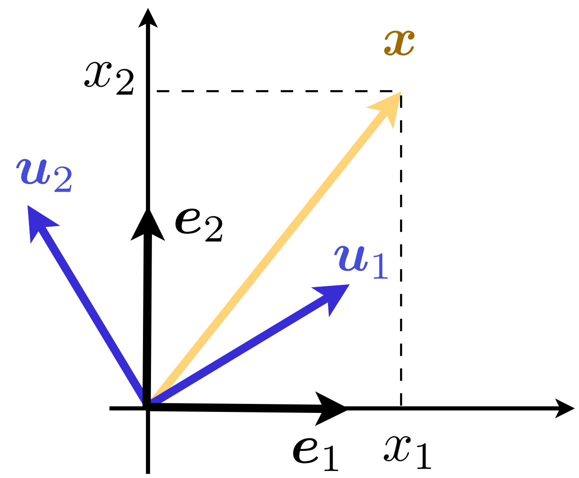
- Orthonormal basis $\quad \Leftrightarrow \quad \langle \boldsymbol{u}_i, \boldsymbol{u}_j \rangle = \delta_{i-j} \quad \Leftrightarrow \quad \boldsymbol{G}_u = \boldsymbol{U}^H\boldsymbol{U} = \boldsymbol{I}$

Transforms and inner product

- Linear transform as a change of basis

Consider the basis $\boldsymbol{U} = \begin{bmatrix} \boldsymbol{u}_1 & \boldsymbol{u}_2 & \cdots & \boldsymbol{u}_N \end{bmatrix}$

$$\boldsymbol{x} = \sum_{i=1}^{N} x_i \boldsymbol{e}_i = \sum_{i=1}^{N} y_i \boldsymbol{u}_i = \boldsymbol{U} \cdot \boldsymbol{y}$$



Question: how do we get the y_i's from the x_i's?

Answer: by linear transform

$$\boldsymbol{x} = \boldsymbol{U} \cdot \boldsymbol{y} \qquad \Rightarrow \qquad \boldsymbol{y} = \boldsymbol{U}^{-1} \cdot \boldsymbol{x}$$

Transform matrix: $\boldsymbol{A} = \boldsymbol{U}^{-1} = \begin{bmatrix} \tilde{\boldsymbol{u}}_1^H \\ \vdots \\ \tilde{\boldsymbol{u}}_N^H \end{bmatrix}$

- Identification of inner products

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} a_{11} & \cdots & a_{1N} \\ \\ a_{N1} & \cdots & a_{NN} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} \quad \Leftrightarrow \quad y_i = \langle \tilde{\boldsymbol{u}}_i, \boldsymbol{x} \rangle, \; (i = 1, \ldots, N)$$

Dual basis

- Linear transform: $\boldsymbol{y} = \boldsymbol{A} \cdot \boldsymbol{x}$ with $\boldsymbol{A} = \boldsymbol{U}^{-1} = \begin{bmatrix} \tilde{\boldsymbol{u}}_1^H \\ \vdots \\ \tilde{\boldsymbol{u}}_N^H \end{bmatrix}$ ($N \times N$ matrix)

- Dual basis: $\tilde{\boldsymbol{U}} = \begin{bmatrix} \tilde{\boldsymbol{u}}_1 & \tilde{\boldsymbol{u}}_2 & \cdots & \tilde{\boldsymbol{u}}_N \end{bmatrix} = \left(\boldsymbol{U}^{-1}\right)^H$ $(= \boldsymbol{A}^H)$ (unique)

  Analysis formula: $\boldsymbol{y} = \tilde{\boldsymbol{U}}^H \boldsymbol{x} \quad \Leftrightarrow \quad y_i = \langle \tilde{\boldsymbol{u}}_i, \boldsymbol{x} \rangle, \ (i = 1, \ldots, N)$

  Synthesis formula: $\boldsymbol{x} = \sum_{i=1}^{N} y_i \boldsymbol{u}_i = \boldsymbol{U} \cdot \boldsymbol{y} \quad \Leftrightarrow \quad \boxed{\boldsymbol{x} = \sum_{i=1}^{N} \langle \tilde{\boldsymbol{u}}_i, \boldsymbol{x} \rangle \, \boldsymbol{u}_i}$

  Biorthogonality (=biorthonormality)

$$\tilde{\boldsymbol{U}}^H \boldsymbol{U} = \begin{bmatrix} \langle \tilde{\boldsymbol{u}}_1, \boldsymbol{u}_1 \rangle & \cdots & \langle \tilde{\boldsymbol{u}}_1, \boldsymbol{u}_N \rangle \\ \vdots & \ddots & \vdots \\ \langle \tilde{\boldsymbol{u}}_N, \boldsymbol{u}_1 \rangle & \cdots & \langle \tilde{\boldsymbol{u}}_N, \boldsymbol{u}_N \rangle \end{bmatrix} = \boldsymbol{U}^{-1} \boldsymbol{U} = \boldsymbol{I} \quad \Leftrightarrow \quad \langle \tilde{\boldsymbol{u}}_i, \boldsymbol{u}_j \rangle = \delta_{i-j}$$

Orthogonal transforms

- Orthonormal basis $\Leftrightarrow$ $\langle \boldsymbol{u}_i, \boldsymbol{u}_j \rangle = \delta_{i-j}$ $\Leftrightarrow$ $\tilde{\boldsymbol{u}}_i = \boldsymbol{u}_i$

$$\boldsymbol{G}_u = \boldsymbol{U}^H \boldsymbol{U} = \boldsymbol{I} \quad \Leftrightarrow \quad \tilde{\boldsymbol{U}} = \boldsymbol{U} = \left(\boldsymbol{U}^{-1}\right)^H \quad \text{(own dual)}$$

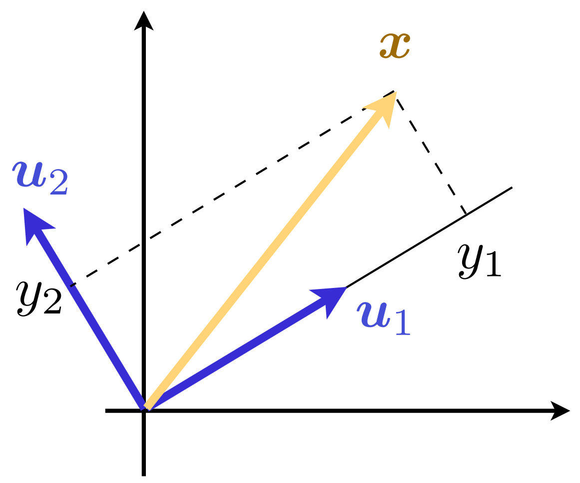
$$\Leftrightarrow \quad \boldsymbol{U} = \begin{bmatrix} \boldsymbol{u}_1 & \cdots & \boldsymbol{u}_N \end{bmatrix} \quad \text{is unitary}$$

- Orthonormal transform: $\boldsymbol{U}^{-1} = \boldsymbol{U}^H$

$$\boldsymbol{y} = \boldsymbol{U}^H \boldsymbol{x} \quad \Leftrightarrow \quad y_i = \langle \boldsymbol{u}_i, \boldsymbol{x} \rangle$$

- Reconstruction formula

$$\boldsymbol{x} = \boldsymbol{U} \cdot \boldsymbol{y} \quad \Leftrightarrow \quad \boldsymbol{x} = \sum_{i=1}^{N} \underbrace{\langle \boldsymbol{u}_i, \boldsymbol{x} \rangle}_{y_i} \boldsymbol{u}_i$$



Transforms: equivalent notations

	2D Sequences $\ell_2(\mathbb{Z}^2)$	Vectors $\mathbb{R}^N$
Input image	$x[k,l]$	$\boldsymbol{x}$
Transform coefficients	$y_{m,n}$	$\boldsymbol{y}$
Basis functions	$\varphi_{m,n}[k,l]$	$\boldsymbol{u}_i$
Analysis functions	$\tilde{\varphi}_{m,n}[k,l]$	$\tilde{\boldsymbol{u}}_i$
Transform formula	$y_{m,n} = \sum_{k,l} x[k,l]\, \tilde{\varphi}^*_{m,n}[k,l]$	$\boldsymbol{y} = \boldsymbol{A} \cdot \boldsymbol{x} = \tilde{\boldsymbol{U}}^H \boldsymbol{x}$
Inner products	$y_{m,n} = \langle \tilde{\varphi}_{m,n}, x \rangle_{\ell_2}$	$y_i = \langle \tilde{\boldsymbol{u}}_i, \boldsymbol{x} \rangle$
Reconstruction formula	$x[k,l] = \sum_{m,n} y_{m,n}\, \varphi_{m,n}[k,l]$	$\boldsymbol{x} = \boldsymbol{U} \cdot \boldsymbol{y}$
Invertibility condition	$\langle \tilde{\varphi}_{m,n}, \varphi_{m',n'} \rangle = \delta_{m-m',n-n'}$	$\tilde{\boldsymbol{U}}^H \boldsymbol{U} = \boldsymbol{I}$
Orthonormality	$\tilde{\varphi}_{m,n} = \varphi_{m,n}$	$\tilde{\boldsymbol{U}} = \boldsymbol{U}$

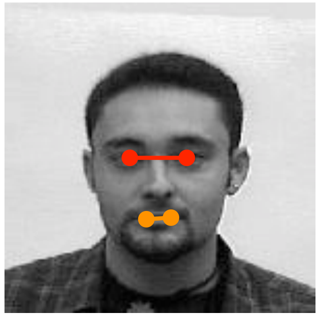
8.2 OPTIMAL TRANSFORMS

- 2^{nd}-order statistics
- Karhunen-Loève transform
- Circulant matrices and DFT

Motivation: face recognition

- Training process with normalized data

Training image



Normalization (eyes + mouth)
Remove background

2^{nd} order vector statistics

Random vector $\boldsymbol{x} = (x_1, \dots, x_N) \in \mathbb{R}^N$

Multivariate probability density function: $p(\boldsymbol{x})$

- Expectation operator: $E\left\{\boldsymbol{f}(\boldsymbol{x})\right\} = \int_{\mathbb{R}^N} \boldsymbol{f}(\boldsymbol{x}) p(\boldsymbol{x}) \,\mathrm{d}x_1 \cdots \mathrm{d}x_N$

- Mean vector: $\boldsymbol{m}_x = \begin{bmatrix} m_1 \\ \vdots \\ m_N \end{bmatrix} = E\{\boldsymbol{x}\} = \int_{\mathbb{R}^N} \boldsymbol{x} \cdot p(\boldsymbol{x}) \,\mathrm{d}x_1 \cdots \mathrm{d}x_N$ (N-vector)

- Covariance matrix: $\boldsymbol{C}_x = E\left\{(\boldsymbol{x} - E\{\boldsymbol{x}\}) \cdot (\boldsymbol{x} - E\{\boldsymbol{x}\})^T\right\}$ (symmetric, $N \times N$)

  Covariances: $[\boldsymbol{C}_x]_{i,j} = E\left\{(x_i - m_i)(x_j - m_j)\right\} = E\{x_i x_j\} - m_i m_j$

  Variances: $[\boldsymbol{C}_x]_{i,i} = \mathrm{Var}\{x_i\} = E\left\{(x_i - m_i)^2\right\} = E\left\{x_i^2\right\} - m_i^2$

Empirical statistics

Image as a vector $\boldsymbol{x} = (x_1, \ldots, x_N) \in \mathbb{R}^N$

Observed set of images: $\{\boldsymbol{x}_k\},\ (k = 1, \ldots, K)$

- Empirical expectation operator: $\hat{E}\left\{\boldsymbol{f}(\boldsymbol{x})\right\} = \dfrac{1}{K}\sum_{k=1}^{K} \boldsymbol{f}(\boldsymbol{x}_k)$

- Average image: $\hat{\boldsymbol{m}}_x = \begin{bmatrix} \hat{m}_1 \\ \vdots \\ \hat{m}_N \end{bmatrix} = \hat{E}\{\boldsymbol{x}\} = \dfrac{1}{K}\sum_{k=1}^{K} \boldsymbol{x}_k; \qquad \hat{m}_i = \dfrac{1}{K}\sum_{k=1}^{K} x_{i,k}$

- Scatter matrix (empirical covariance)

$$\boldsymbol{S}_x = \hat{E}\left\{\left(\boldsymbol{x} - \hat{E}\{\boldsymbol{x}\}\right) \cdot \left(\boldsymbol{x} - \hat{E}\{\boldsymbol{x}\}\right)^T\right\}$$

symmetric $N \times N$ matrix

$$= \hat{E}\left\{\boldsymbol{x}\boldsymbol{x}^T\right\} - \hat{E}\{\boldsymbol{x}\}\hat{E}\{\boldsymbol{x}\}^T = \left(\frac{1}{K}\sum_{k=1}^{K} \boldsymbol{x}_k \boldsymbol{x}_k^T\right) - \hat{\boldsymbol{m}}_x \hat{\boldsymbol{m}}_x^T$$

$$\mathrm{Scatter}\{x_i\} = \hat{E}\left\{(x_i - \hat{m}_i)^2\right\} = \frac{1}{K}\sum_{k=1}^{K} (x_{i,k} - \hat{m}_i)^2$$ (sum of squares)

Linear transform of random vector variables

- Linear transform: $\boldsymbol{x} = (x_1, \ldots, x_N) \in \mathbb{R}^N \quad \longrightarrow \quad \boldsymbol{y} = (y_1, \ldots, y_M) \in \mathbb{R}^M$

  $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ where $\boldsymbol{A} = \begin{bmatrix} \boldsymbol{u}_1 & \cdots & \boldsymbol{u}_M \end{bmatrix}^T$ ($M \times N$ matrix)

  or $y_i = \boldsymbol{u}_i^T \boldsymbol{x} = \langle \boldsymbol{u}_i, \boldsymbol{x} \rangle, \quad (i = 1, \ldots, M)$

- Mean/average vectors: $E\{\boldsymbol{y}\} = E\{\boldsymbol{A}\boldsymbol{x}\} = \boldsymbol{A}E\{\boldsymbol{x}\} \quad \Rightarrow \quad \boldsymbol{m}_y = \boldsymbol{A}\boldsymbol{m}_x$

  Likewise, $\hat{\boldsymbol{m}}_y = \boldsymbol{A}\hat{\boldsymbol{m}}_x$

- Covariance/scatter matrices: $E\Big\{(\boldsymbol{y} - E\{\boldsymbol{y}\}) \cdot (\boldsymbol{y} - E\{\boldsymbol{y}\})^T\Big\} = \boldsymbol{C}_y = \boldsymbol{A}\boldsymbol{C}_x\boldsymbol{A}^T$

  Likewise, $\boldsymbol{S}_y = \boldsymbol{A}\boldsymbol{S}_x\boldsymbol{A}^T$

  Justification: $\boldsymbol{C}_y = E\Big\{(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}E\{\boldsymbol{x}\}) \cdot (\boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}E\{\boldsymbol{x}\})^T\Big\} = \boldsymbol{A} \cdot E\Big\{(\boldsymbol{x} - E\{\boldsymbol{x}\}) \cdot (\boldsymbol{x} - E\{\boldsymbol{x}\})^T\Big\} \cdot \boldsymbol{A}^T$

- Dispersion measures: $\mathrm{Var}\{y_i\} = \boldsymbol{u}_i^T \boldsymbol{C}_x \boldsymbol{u}_i$

  Likewise, $\mathrm{Scatter}\{y_i\} = \boldsymbol{u}_i^T \boldsymbol{S}_x \boldsymbol{u}_i$

Karhunen-Loève transform

$\{\boldsymbol{x}\}$: N-vector process with known covariance matrix $\boldsymbol{C}_x$

- Eigenvectors of symmetric matrix $\boldsymbol{C}_x$

Defining property: $\quad \boldsymbol{C}_x \cdot \boldsymbol{v}_n = \lambda_n \boldsymbol{v}_n, \quad (n = 1, \ldots, N)$

Orthonormality of eigenvectors: $\quad \langle \boldsymbol{v}_m, \boldsymbol{v}_n \rangle = \delta_{m-n}$

Ordered eigenvalues: $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_N$ with $\quad \lambda_n = \boldsymbol{v}_n^T \boldsymbol{C}_x \boldsymbol{v}_n$

- Karhunen-Loève transform of $\{\boldsymbol{x}\}$

$$\boldsymbol{A}_{\mathrm{KLT}} = \begin{bmatrix} \boldsymbol{v}_1 & \cdots & \boldsymbol{v}_N \end{bmatrix}^H$$

$$\boldsymbol{C}_x \cdot \boldsymbol{v}_n = \lambda_n \boldsymbol{v}_n \quad \Leftrightarrow \quad \boldsymbol{C}_x = \begin{bmatrix} \boldsymbol{v}_1 & \cdots & \boldsymbol{v}_N \end{bmatrix} \cdot \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_N \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{v}_1^H \\ \vdots \\ \boldsymbol{v}_N^H \end{bmatrix}$$

Using $\boldsymbol{S}_x$ instead of $\boldsymbol{C}_x \quad \Rightarrow \quad$ Principal components (PC)

Properties of the KLT

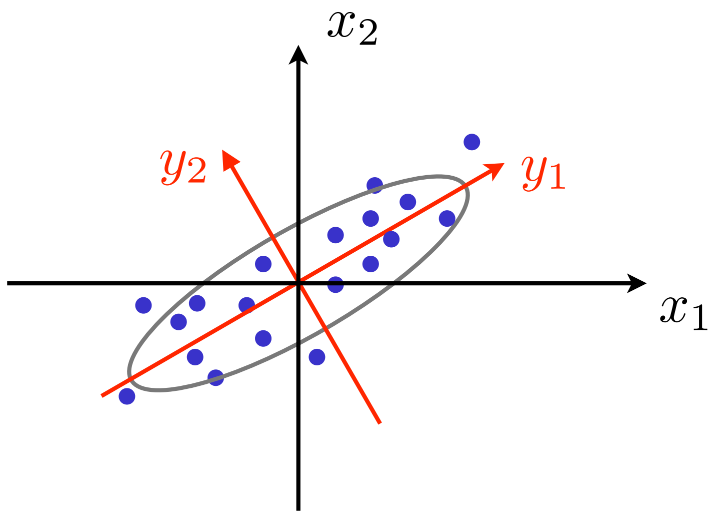
- Decorrelation: $\boldsymbol{C}_{\mathrm{KLT}} = \boldsymbol{A}_{\mathrm{KLT}} \cdot \boldsymbol{C}_x \cdot \boldsymbol{A}_{\mathrm{KLT}}^H = \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_N \end{bmatrix}$

- Maximum energy compaction

$\mathrm{Var}\{y_1\} = \lambda_1 = \boldsymbol{v}_1^H \boldsymbol{C}_x \boldsymbol{v}_1$: maximum variance or scatter

$\vdots$

$\mathrm{Var}\{y_N\} = \lambda_N = \boldsymbol{v}_N^H \boldsymbol{C}_x \boldsymbol{v}_N$: minimum variance or scatter



- Minimum basis-restriction error

Orthonormal transform: $\boldsymbol{A} = \begin{bmatrix} \boldsymbol{u}_1 & \cdots & \boldsymbol{u}_N \end{bmatrix}^H$

Subspace approximation: $\boldsymbol{x}_n = \sum_{i=1}^{n} \langle \boldsymbol{u}_i, \boldsymbol{x} \rangle \, \boldsymbol{u}_i \qquad (n < N)$

Basis-restriction error: $E\left\{\|\boldsymbol{x} - \boldsymbol{x}_n\|^2\right\} = \sum_{i=n+1}^{N} E\left\{|y_i|^2\right\} = \sum_{i=n+1}^{N} \boldsymbol{u}_i^H \boldsymbol{C}_x \boldsymbol{u}_i \geq \sum_{i=n+1}^{N} \lambda_i$

Best approximation: The KLT minimizes the basis-restriction error for $1 \leq n < N$.

KLT optimality: General variance-based criterion

Theorem: Among all orthonormal transforms, the KLT optimizes the criterion

$$J_n = \sum_{m=1}^{n} G\left(\boldsymbol{u}_m^H \boldsymbol{C}_x \boldsymbol{u}_m\right) = \sum_{m=1}^{n} G\left(\mathrm{Var}\{y_m\}\right), \qquad \text{for } n \leqslant N$$

where $G(\cdot)$ is any monotonic *concave* or *convex* function.

Examples:

$G(\sigma^2) = \sigma^2$ $\Rightarrow$ basis-restriction error

$G(\sigma^2) = \max\left\{0, \frac{1}{2}\log_2(\sigma^2/D)\right\}$ $\Rightarrow$ bit rate for distortion D (Gaussian process)

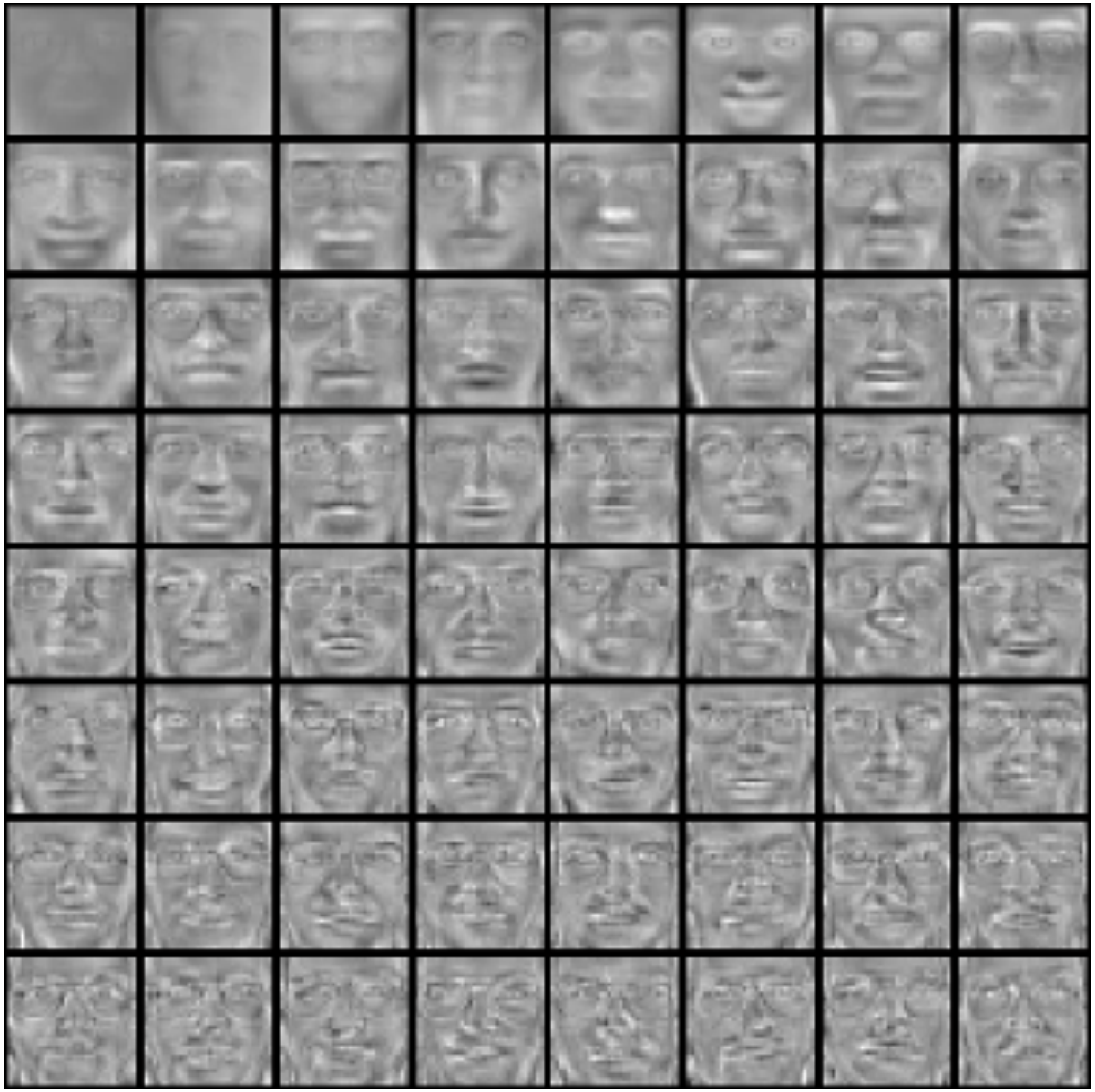
$G(\sigma^2) = \log(\sigma^2)$ $\Rightarrow$ variance product

Determinant inequality: $\det(\boldsymbol{C}_x) = \prod_{n=1}^{N} \lambda_n \leqslant \prod_{n=1}^{N} \underbrace{[\boldsymbol{C}_x]_{n,n}}_{\sigma_n^2}$

Application of the KLT: face recognition

- Eigenfaces

Mean

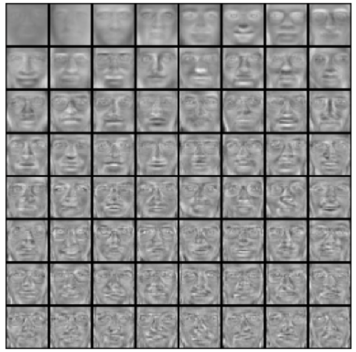


Eigenvectors of scatter matrix

Application of the KLT: face recognition

- Eigenfaces

Mean



Eigenvectors of scatter matrix

Each face has its “face space” coordinates from which it can be reconstructed:

$\mathbf{x}$

$=$

$+$

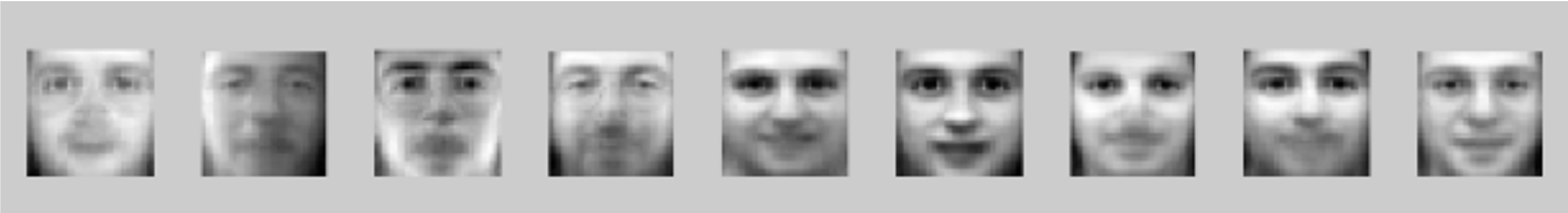
$\langle \mathbf{x}, \mathbf{v}_1 \rangle \, \mathbf{v}_1 \quad \langle \mathbf{x}, \mathbf{v}_2 \rangle \, \mathbf{v}_2 \quad \langle \mathbf{x}, \mathbf{v}_3 \rangle \, \mathbf{v}_3 \quad \langle \mathbf{x}, \mathbf{v}_4 \rangle \, \mathbf{v}_4 \quad \ldots$

Application of the KLT: face recognition

- Navigate in "face space"

Leading eigenvectors $\mathbf{v}_n$ of scatter matrix

Reconstruct mean + 3 $\sqrt{\lambda_n}\mathbf{v}_n$



Reconstruct mean - 3 $\sqrt{\lambda_n}\mathbf{v}_n$

From KLT to variational autoencoders

- KLT:

encoder (analysis) = inner products with eigenvectors $\mathbf{v}_n$

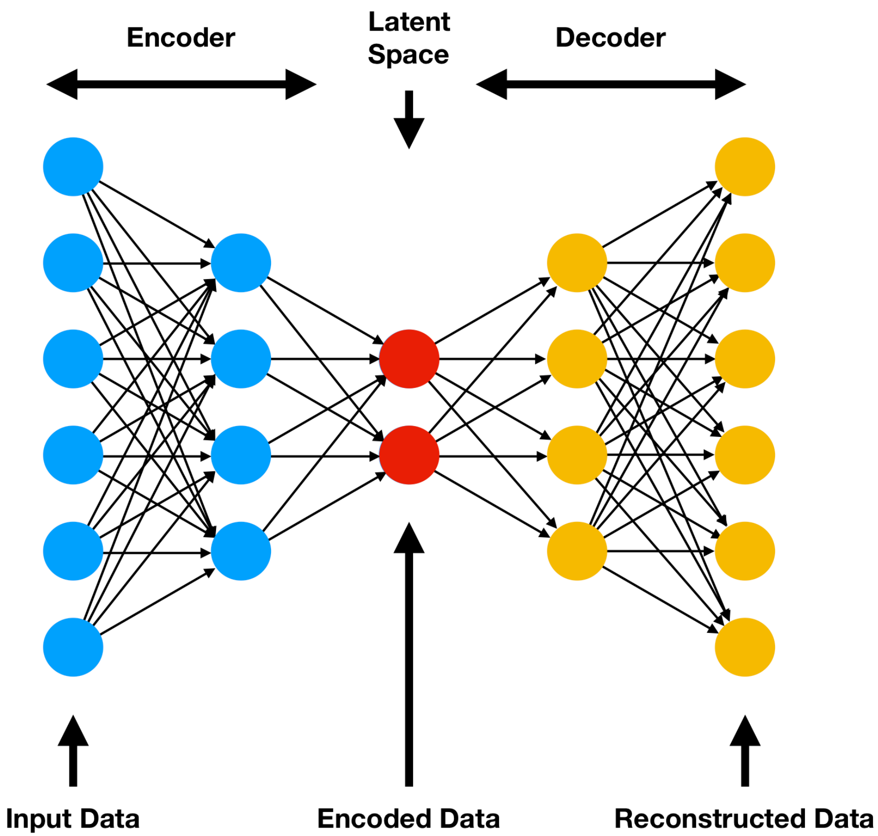
decoder (synthesis) = reconstruction with eigenvectors $\mathbf{v}_n$

- Variational autoencoder

encoder and decoder become **non-linear** with the aim that latent space representations map faces/images to a standard normal distribution

Example: interpolation in latent space between two sketch drawings

[Ha and Eck, *ICLR* 2018]



Circulant and Toeplitz matrices

Back to 1D

- Circulant matrices

$$\boldsymbol{C}_h = \begin{bmatrix} h[0] & h[1] & \cdots & & h[N-1] \\ h[N-1] & h[0] & h[1] & \cdots & \\ & \ddots & \ddots & & \\ h[1] & h[2] & & h[N-1] & h[0] \end{bmatrix}$$

1D convolution with periodic boundary conditions: $y_{\text{per}}[k] = (h * x_{\text{per}})[k] \quad \Leftrightarrow \quad \boldsymbol{y} = \boldsymbol{C}_h \boldsymbol{x}$

- Toeplitz matrices

$$\boldsymbol{T}_r = \begin{bmatrix} r[0] & r[1] & \cdots & \cdots & \cdots & r[N-1] \\ r[-1] & r[0] & & & & \\ & \ddots & \ddots & \ddots & & \\ \cdots & \cdots & r[-1] & r[0] & r[1] & \cdots \end{bmatrix} \quad \Leftrightarrow \quad [\boldsymbol{T}_r]_{i,j} = r[j-i]$$

Covariance matrix of a stationary process: $E\left\{x[i]^\star x[j]\right\} = r_x[i-j] \quad \Leftrightarrow \quad \boldsymbol{C}_x = \boldsymbol{T}_{r_x}^T$

DFT and circulant matrices

- Discrete Fourier transform (DFT)

$N \times N$ matrix: $\boldsymbol{F}_N = \begin{bmatrix} \boldsymbol{u}_1 & \cdots & \boldsymbol{u}_N \end{bmatrix}^H$

Basis vectors: $\boldsymbol{u}_n = (\varphi_n[0], \varphi_n[1], \ldots, \varphi_n[N-1])$

with $\varphi_n[k] = \frac{1}{\sqrt{N}} e^{j\omega_n k}, \quad \omega_n = \frac{2\pi(n-1)}{N}$

- Diagonalization property

Theorem: The DFT diagonalizes all circulant matrices

$$\boldsymbol{C}_h = \boldsymbol{F}_N^H \cdot \mathrm{diag}(\lambda_1, \ldots, \lambda_N) \cdot \boldsymbol{F}_N$$

$$\lambda_n = \boldsymbol{u}_n^H \boldsymbol{C}_h \boldsymbol{u}_n = \sum_{k=0}^{N-1} h[k] e^{-j(n-1)\omega_0 k} = H\left(e^{j(n-1)\omega_0}\right) \quad \text{with} \quad \omega_0 = \tfrac{2\pi}{N}$$

$\Rightarrow$ FFT-based convolution algorithm

$$\boldsymbol{y} = \boldsymbol{C}_h \cdot \boldsymbol{x} = \boldsymbol{F}_N^H \cdot \mathrm{diag}(\lambda_1, \ldots, \lambda_N) \cdot \boldsymbol{F}_N \boldsymbol{x}$$

Optimality of DFT for stationary processes

- Covariance of a wide-sense stationary process
  - Shift-invariant structure: $\forall k, l \in \mathbb{Z}, \quad E\{x[k]x[l]\} = r_x[k-l]$

    $\Rightarrow$ The covariance matrix is a symmetric Toeplitz matrix: $\quad \boldsymbol{C}_x = E\{\boldsymbol{x}\boldsymbol{x}^T\} = \boldsymbol{T}_{r_x}$
  - Asymptotic behavior

    Limit process: let the block size N tend to infinity

    Toeplitz matrix $\quad \longrightarrow \quad$ Circulant matrix of infinite dimension

    (provided $r_x[k]$ has sufficient decay; e.g., $r_x \in \ell_1$)

Theorem: The DFT diagonalizes Toeplitz matrices as the block size N tends to infinity. It is therefore asymptotically equivalent to the KLT.

Justification: $\quad \boldsymbol{T}_{r_x}\boldsymbol{u} \quad \longrightarrow \quad (r_x * u)[k] \quad$ (aperiodic convolution)

The Fourier exponentials $e^{j\omega k}$ are the “eigen-sequences” of shift-invariant systems

$\Rightarrow$ Fourier domain formulation of Wiener filter for stationary processes

8.3 CLASSICAL 2D TRANSFORMS

- Separable transforms
- Discrete Cosine transform (DCT)
- Other transforms

Separable transforms

- Indexing

2D digital image: $(k = 0, \ldots, K-1;\ l = 0, \ldots, L-1)$

2D basis functions: $(m = 1, \ldots, K;\ n = 1, \ldots, L)$

- Separable basis functions: $\varphi_{m,n}[k,l] = \varphi_m[k] \cdot \varphi_n[l]$

Example: DFT

$$\varphi_{m,n}[k,l] = \frac{1}{\sqrt{KL}} e^{j\left(\frac{2\pi(m-1)k}{K} + \frac{2\pi(n-1)l}{L}\right)} = \frac{1}{\sqrt{K}} e^{j\frac{2\pi(m-1)k}{K}} \cdot \frac{1}{\sqrt{L}} e^{j\frac{2\pi(n-1)l}{L}}$$

- Image decomposition

$$x[k,l] = \sum_{m=1}^{K} \sum_{n=1}^{L} y[m,n]\, \varphi_m[k] \varphi_n[l]$$

Separable transform: matrix formalism

- Row transform

Basis vectors: $\boldsymbol{u}_m = (\varphi_m[0], \varphi_m[1], \ldots, \varphi_m[K-1])$

$K \times K$ matrix: $\boldsymbol{A}_R = \begin{bmatrix} \boldsymbol{u}_1 & \cdots & \boldsymbol{u}_K \end{bmatrix}^H$

- Column transform

Basis vectors: $\boldsymbol{v}_n = (\varphi_n[0], \varphi_n[1], \ldots, \varphi_n[L-1])$

$L \times L$ matrix: $\boldsymbol{A}_C = \begin{bmatrix} \boldsymbol{v}_1 & \cdots & \boldsymbol{v}_L \end{bmatrix}^H$

- Separable 2D transform

$$\boldsymbol{A} = \boldsymbol{A}_R \otimes \boldsymbol{A}_C = \begin{bmatrix} \boldsymbol{A}_R\varphi_1[0] & \boldsymbol{A}_R\varphi_1[1] & \cdots & \boldsymbol{A}_R\varphi_1[L-1] \\ \boldsymbol{A}_R\varphi_2[0] & & \cdots & \boldsymbol{A}_R\varphi_2[L-1] \\ \vdots & & & \vdots \\ \boldsymbol{A}_R\varphi_L[0] & & & \boldsymbol{A}_R\varphi_L[L-1] \end{bmatrix}$$

Separable transform: implementation

Input: $f[k,l]$ with $(k=0,\ldots,K-1;\ l=0,\ldots,L-1)$

Transformed image: $g[k,l]$

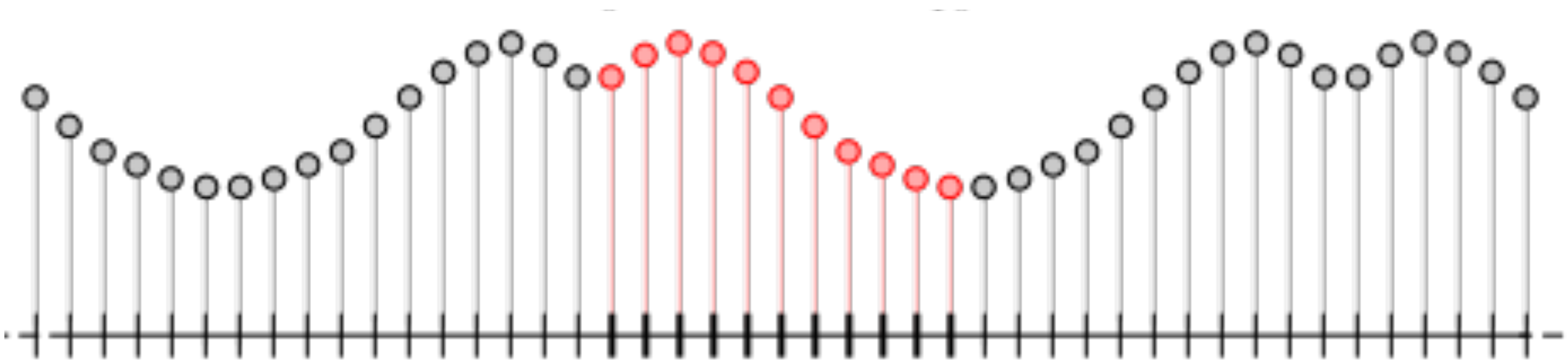
```
For j = 0, L-1  {
   x = getrow(f,j);
   y = A_R x;                 (* fast algorithm *)
   putrow(g,j,y);
}


For i = 0, K-1  {
   x = getcol(g,i);
   y = A_C x;                 (* fast algorithm *)
   putcol(g,i,y);
}


etc...(for 3D or more)
```

Discrete Cosine transform (DCT)

DCT: prototype of an orthogonal block transform

- Basis functions

$$\varphi_m[k] = \begin{cases} \sqrt{\frac{1}{K}}, & m = 1 \\ \sqrt{\frac{2}{K}} \cos\left(\frac{\pi\left(k+\frac{1}{2}\right)(m-1)}{K}\right), & m \in \{2, \ldots, K\} \end{cases}$$



Discrete Cosine transform (DCT)

DCT: prototype of an orthogonal block transform

- Basis functions

$$\varphi_m[k] = \begin{cases} \sqrt{\frac{1}{K}}, & m = 1 \\ \sqrt{\frac{2}{K}} \cos\left(\frac{\pi\left(k+\frac{1}{2}\right)(m-1)}{K}\right), & m \in \{2, \ldots, K\} \end{cases}$$

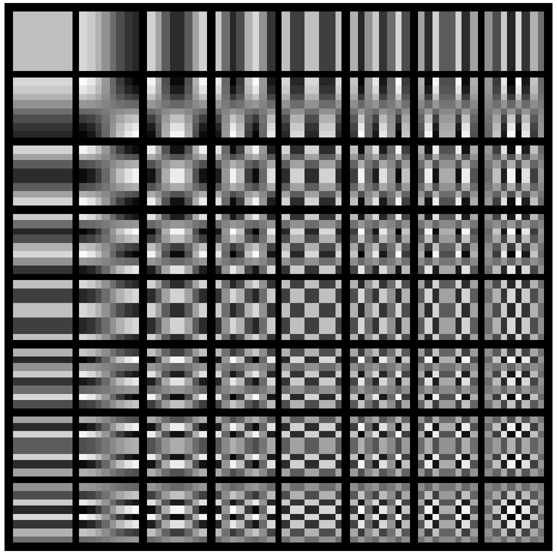
- Properties
  - Good approximation of KLT for strongly correlated stationary processes
  - Excellent energy compaction
  - Fast FFT-like algorithm; real computations

Theorem [U., 1984]. The DCT transform is asymptotically (as the block size N goes to infinity) equivalent to the KLT for all wide-sense stationary processes.

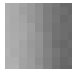
$\Rightarrow$ Good justification for JPEG

8×8 DCT basis functions (zoom)

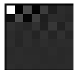
$$\{\boldsymbol{u}_i\}_{i=1,\cdots,64}$$



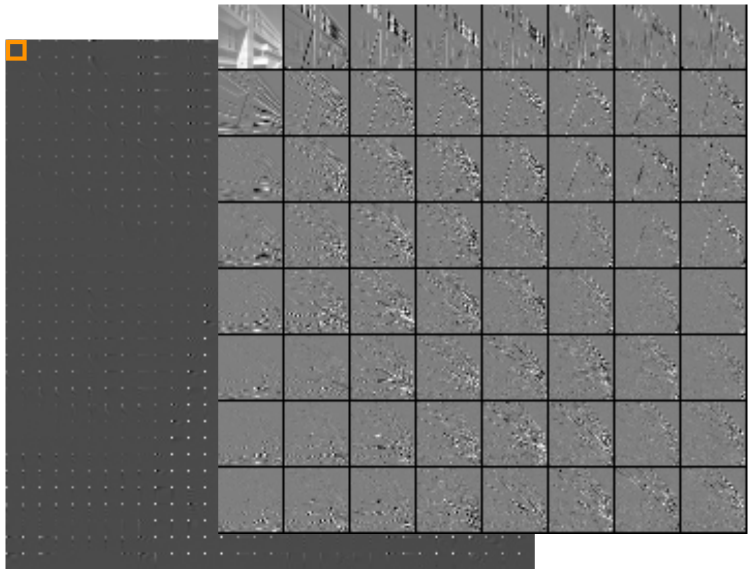
$$\boldsymbol{x}$$



$$\boldsymbol{y} = \begin{bmatrix} \langle \boldsymbol{u}_1, \boldsymbol{x} \rangle \\ \vdots \\ \langle \boldsymbol{u}_{64}, \boldsymbol{x} \rangle \end{bmatrix}$$



Input image



8×8 block DCT transform

Main applications of the DCT

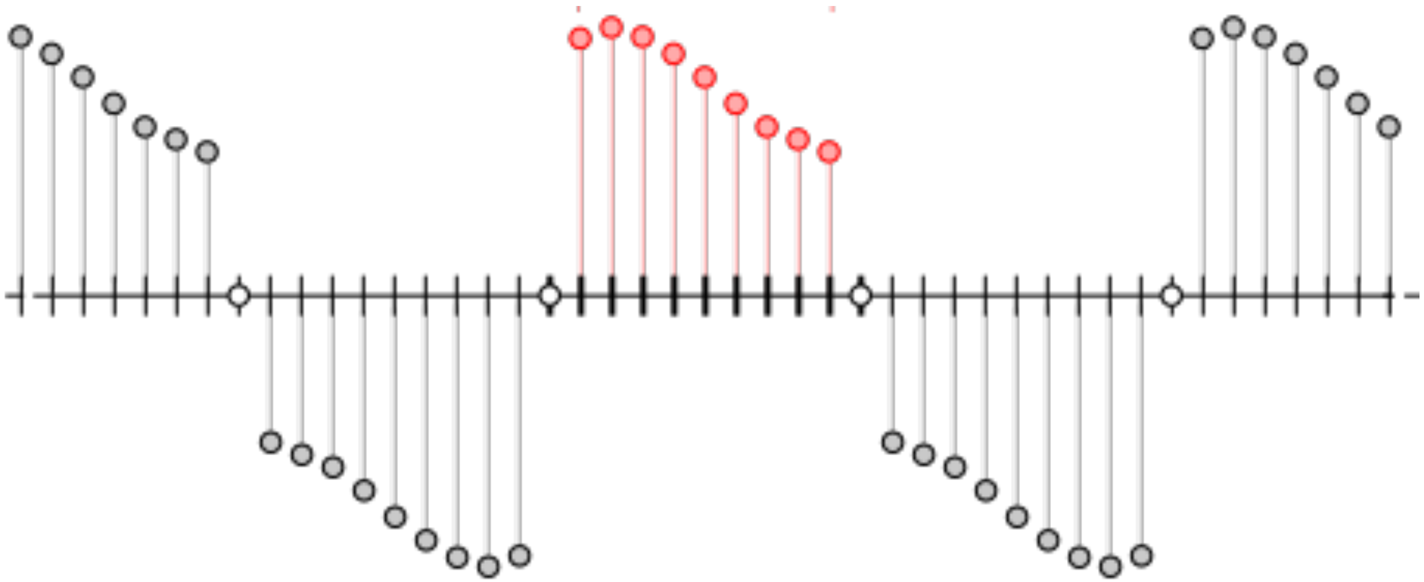
- Coding (JPEG industry standard)
- Adaptive filtering

Other transforms

- Sine transform

$$\varphi_m[k] = \sqrt{\frac{2}{K+1}} \sin\left(\frac{\pi m(k+1)}{K+1}\right) \qquad (m = 1, \ldots, K)$$

  - Good approximation of KLT for weakly-correlated stationary processes
  - Fast FFT-like algorithm; $\mathcal{O}(K \log K)$ real operations



Other transforms (Cont'd)

- Hadamard transform

  Recursive definition: $\boldsymbol{U}_{2N} = \frac{1}{\sqrt{2}} \begin{bmatrix} \boldsymbol{U}_N & \boldsymbol{U}_N \\ \boldsymbol{U}_N & -\boldsymbol{U}_N \end{bmatrix}$ with $\boldsymbol{U}_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$

  - Poor man's version of sinusoidal (DFT-like) transform
  - Very fast algorithm; $\mathcal{O}(K \log K)$ integer adds

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

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

$$(H_n)_{i,j} = \frac{1}{2^{\frac{n}{2}}}(-1)^{i \cdot j}$$

Other transforms (Cont'd)

- Hadamard transform

  Recursive definition: $U_{2N} = \frac{1}{\sqrt{2}} \begin{bmatrix} U_N & U_N \\ U_N & -U_N \end{bmatrix}$

  - Poor man's version of sinusoidal (DFT-like) transfo
  - Very fast algorithm; $\mathcal{O}(K \log K)$ integer adds

- Slant transform

  Somewhat exotic; never seen an application

- Haar transform (cf. Wavelet transform)
  - Average energy compaction, but that's not the who
  - Ultra-fast algorithm; $\mathcal{O}(K)$ integer adds

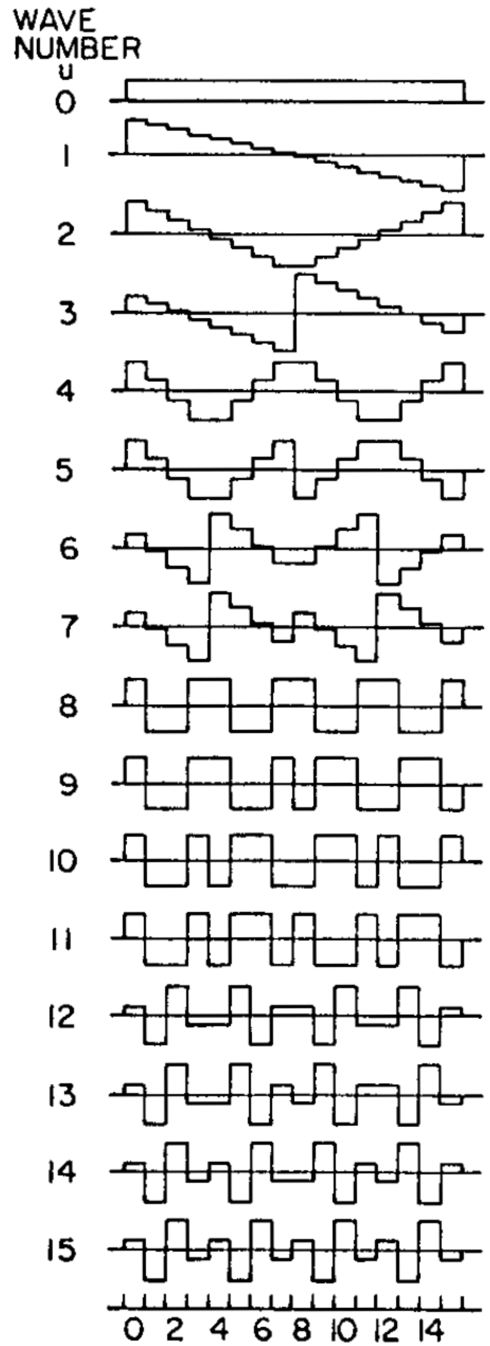


Fig. 3. Slant-transform basis waveforms.

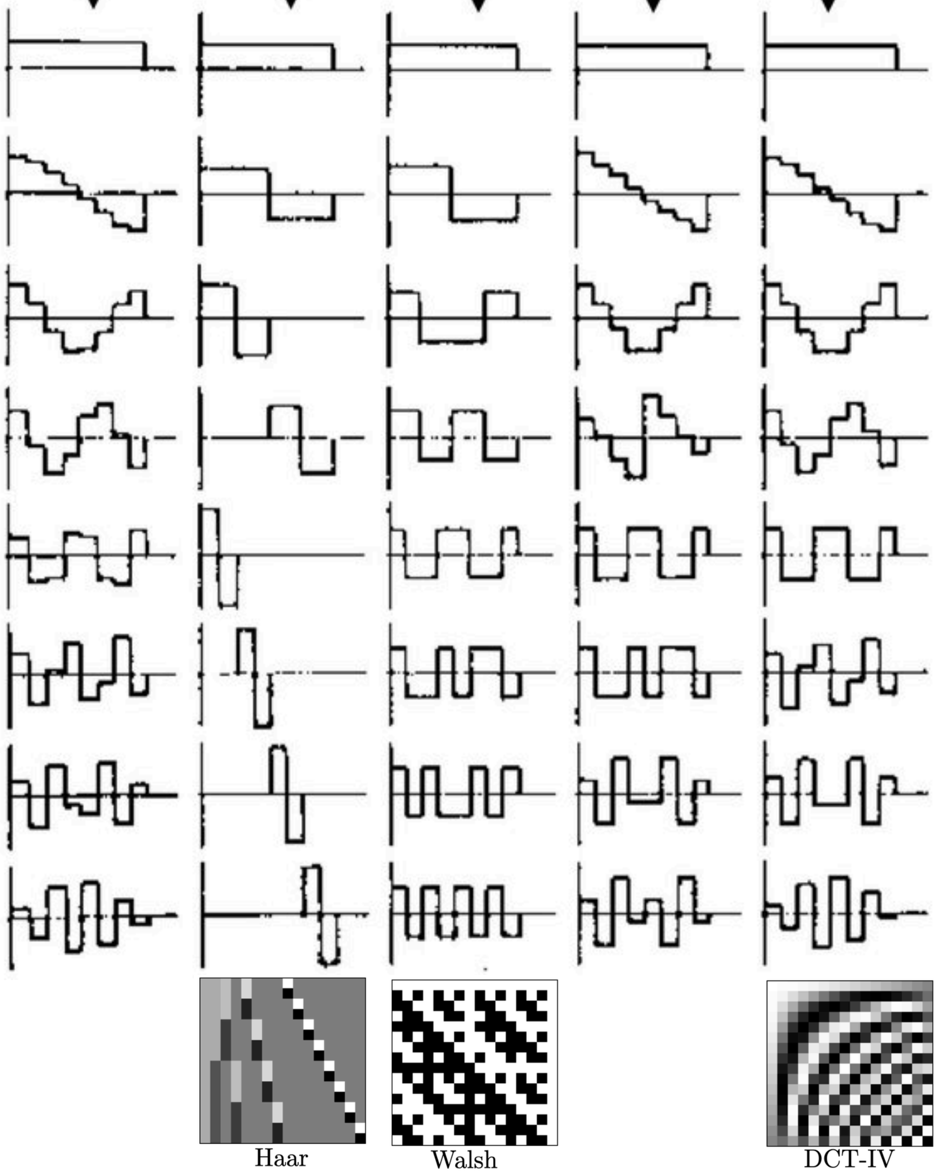
Karhunen Loève transform *[1948/1960]*

Haar transform *[1910]*

Walsh-Hadamard transform *[1923]*

Slant transform *[Enomoto, Shibata, 1971]*

Discrete CosineTransform (DCT) *[Ahmet, Natarajan, Rao, 1974]*



Comparison of 1-d basis functions for block size $N=8$

8.4 WAVELET TRANSFORM

- History
- Signal-processing perspective
- Multirate operations
- Perfect reconstruction filterbank
- Hilbert-space interpretation
- Orthogonal wavelet filters
- Computing wavelet transforms
- Wavelets: Further probing

Famous waveleteer...

The Norwegian Academy of Science and Letters has decided
to award the Abel Prize for 2017 to

Yves Meyer

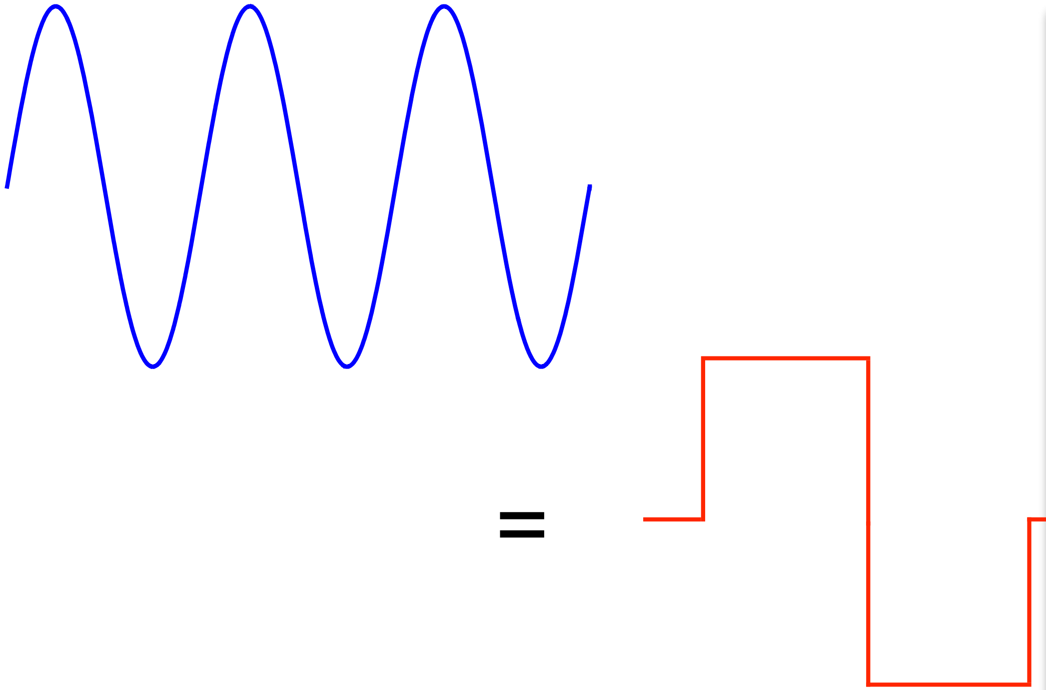
of the École normale supérieure Paris-Saclay, France

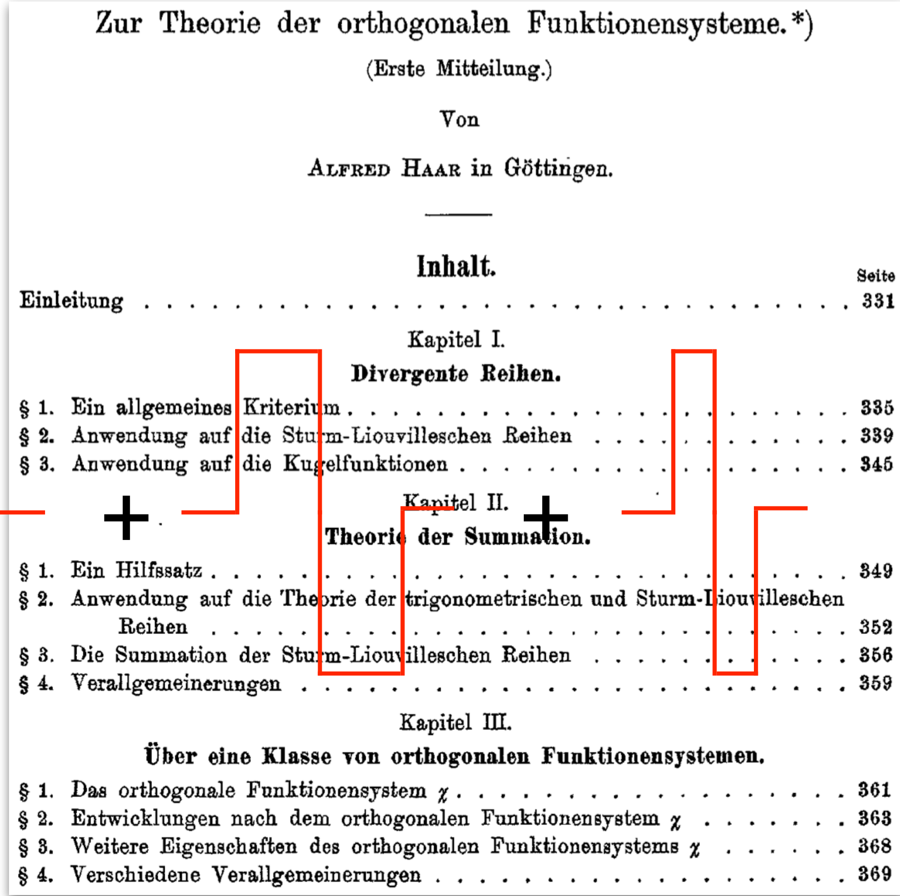
“for his pivotal role in the development of the mathematical theory of wavelets.”

THE ABEL PRIZE 2017

Early start: Haar transform

1910: Alfred Haar invents his own transform
(Note: no fancy title... "On the Theory of Orthogonal Function Systems")





Zur Theorie der orthogonalen Funktionensysteme.*)

(Erste Mitteilung.)

Von

ALFRED HAAR in Göttingen.

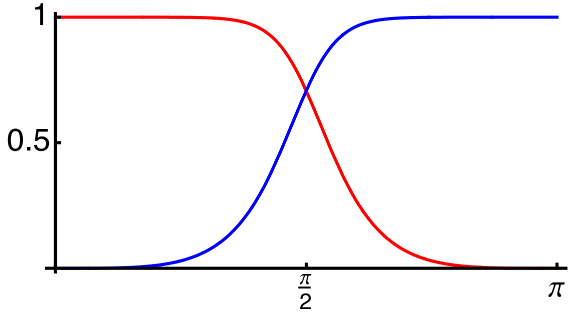
Inhalt.

- Basis functions: $\psi_{i,k} = 2^{-i/2}\psi(x/2^i - k)$
- Wavelet expansion: $f(x) = \sum_{i,k} \underbrace{\langle f, \psi_{i,k} \rangle}_{y_{i,k}} \psi_{i,k}$

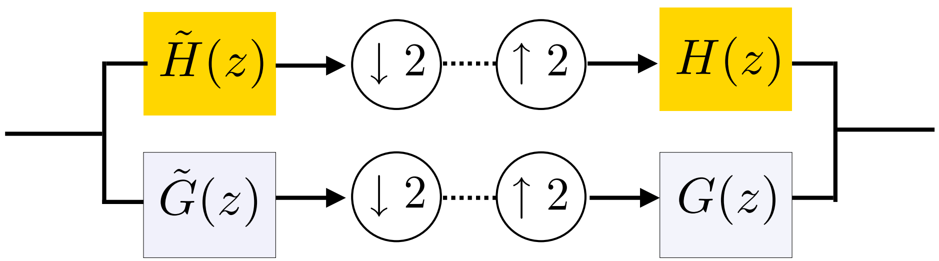
Signal processing developments

1948 - 1980: Dark ages = “DCT era”

1976: Quadrature-mirror filters [Croisier-Esteban-Galand]



Early 1980’s: Perfect-reconstruction filterbanks [Smith-Barnwell, Vaidyanathan, Vetterli]



1984: Vetterli splits a cat into subbands

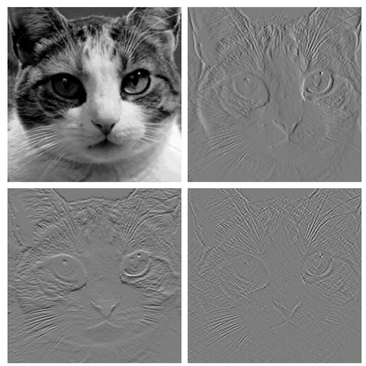
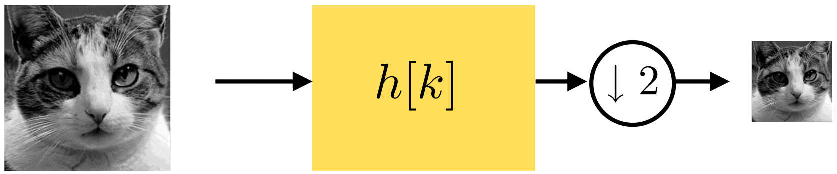


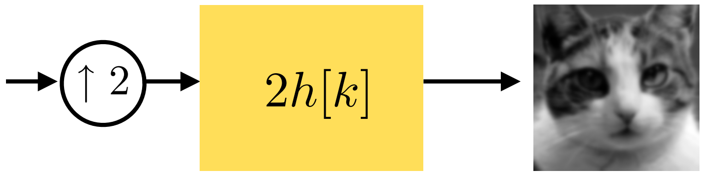
Image pyramids

1983: Burt and Adelson build pyramids

REDUCE

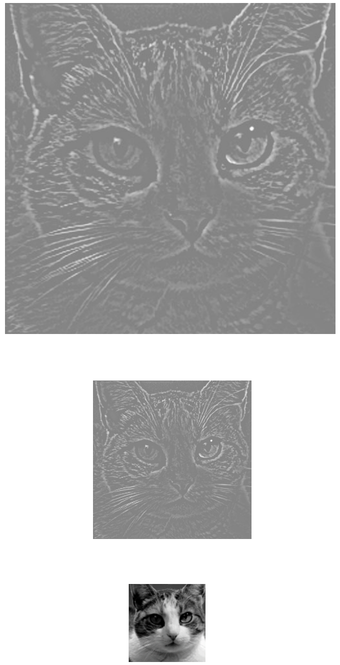


EXPAND



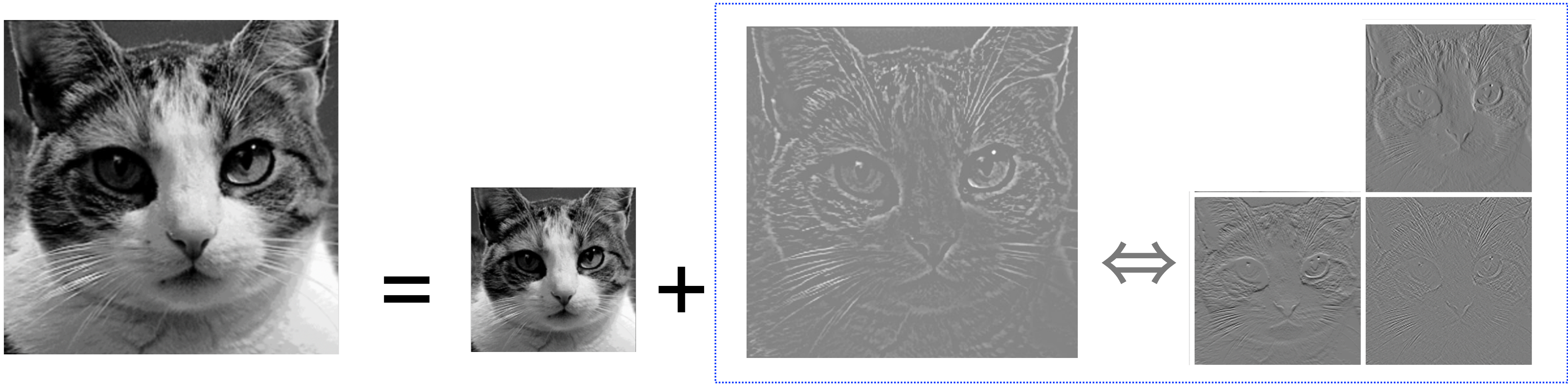
Gaussian pyramid

Laplacian pyramid

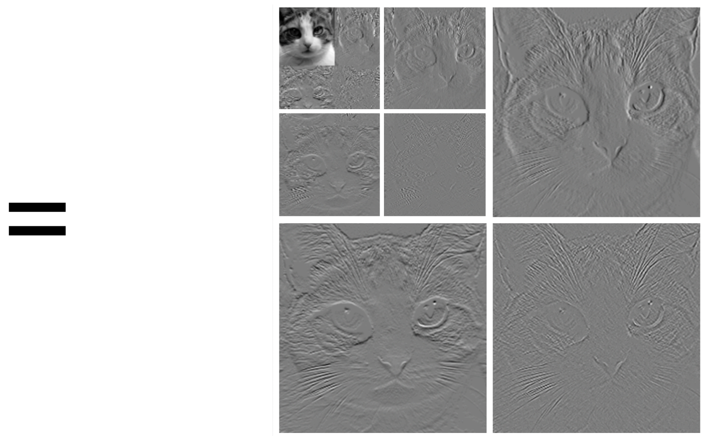


Unification: Multiresolution analysis

1987: Mallat and Meyer set the foundations of the theory



and iterate.... you get the wavelet transform:

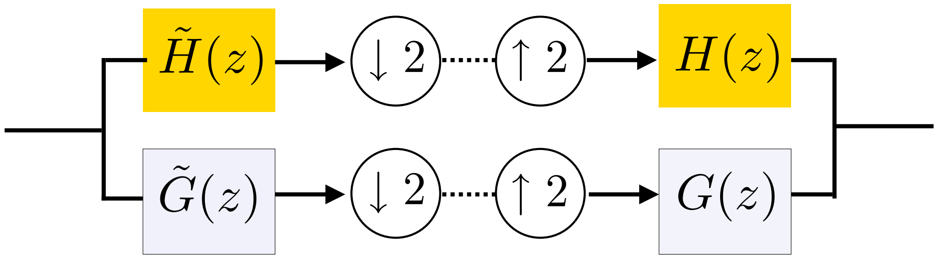


S.G. Mallat, "A theory of multiresolution signal decomposition: the wavelet representation," *IEEE Trans. Pattern Anal. Machine Intell.*, 11 (7), pp. 674-693, 1989

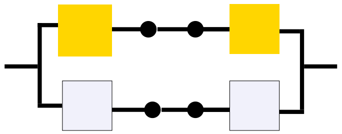
Signal-processing perspective

Splitting and putting together again...

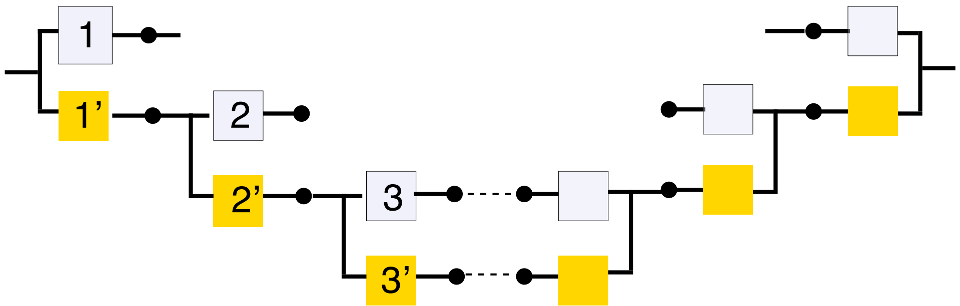
- Perfect-reconstruction filterbank



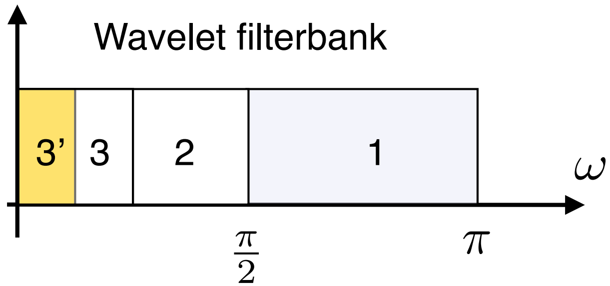
Identity operator:



- Tree-structured wavelet transform

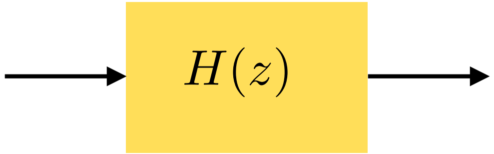


- Subband decomposition



Multirate operations

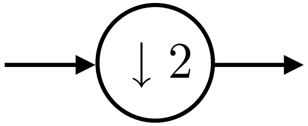
- Filtering



z-transform: $\quad x[k] \overset{z}{\longleftrightarrow} X(z) = \sum_{k\in\mathbb{Z}} x[k] z^{-k}$

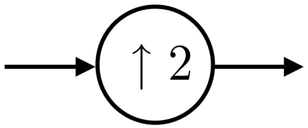
$$(h*x)[k] = \sum_{l\in\mathbb{Z}} h[l]x[k-l] \quad \overset{z}{\longleftrightarrow} \quad H(z)\cdot X(z)$$

- Down-sampling



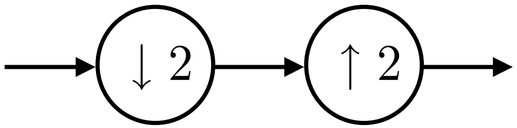
$$(x)_{\downarrow 2}[k] = x[2k] \quad \overset{z}{\longleftrightarrow} \quad \frac{1}{2}\Big(X(z^{1/2}) + X(-z^{1/2})\Big)$$

- Up-sampling



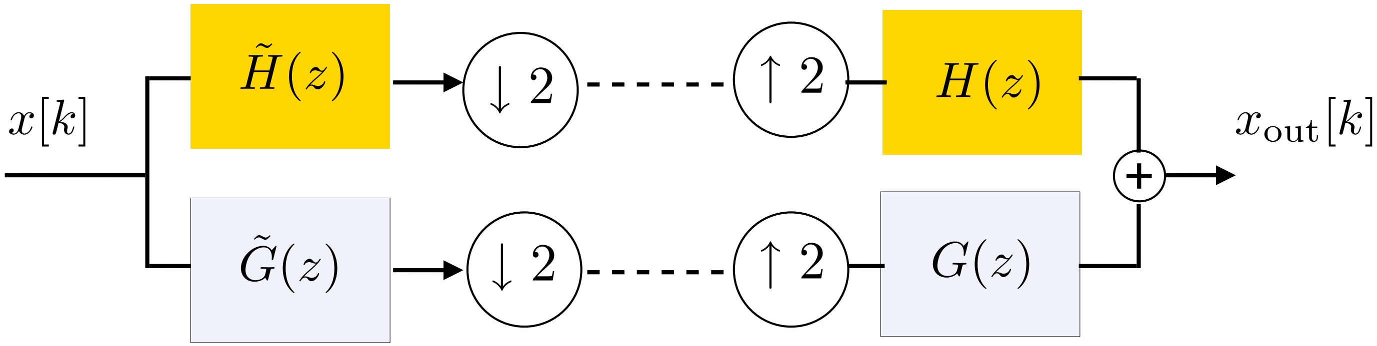
$$(x)_{\uparrow 2}[k] = \begin{cases} 0, & k \text{ odd} \\ x[l], & 2l = k \text{ even} \end{cases} \quad \overset{z}{\longleftrightarrow} \quad X(z^2)$$

- Down-sampling followed by up-sampling



$$(x)_{\downarrow 2 \uparrow 2}[k] \quad \overset{z}{\longleftrightarrow} \quad \frac{1}{2}\left(X(z) + X(-z)\right)$$

Perfect-reconstruction filterbanks



$$X_{\text{out}}(z) = \frac{1}{2}H(z)\left(\tilde{H}(z){\color{blue}X(z)} + \tilde{H}(-z){\color{red}X(-z)}\right) + \frac{1}{2}G(z)\left(\tilde{G}(z){\color{blue}X(z)} + \tilde{G}(-z){\color{red}X(-z)}\right)$$

- Perfect reconstruction conditions

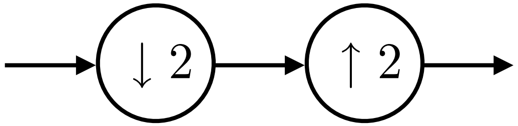
(PR-1) $\quad \tilde{H}(z)H(z) + \tilde{G}(z)G(z) = 2 \quad$ (distortion-free)

(PR-2) $\quad \tilde{H}(-z)H(z) + \tilde{G}(-z)G(z) = 0 \quad$ (aliasing-free)

- Wavelet transform design

Construct 4 filters such that (PR-1) and (PR-2) are satisfied.

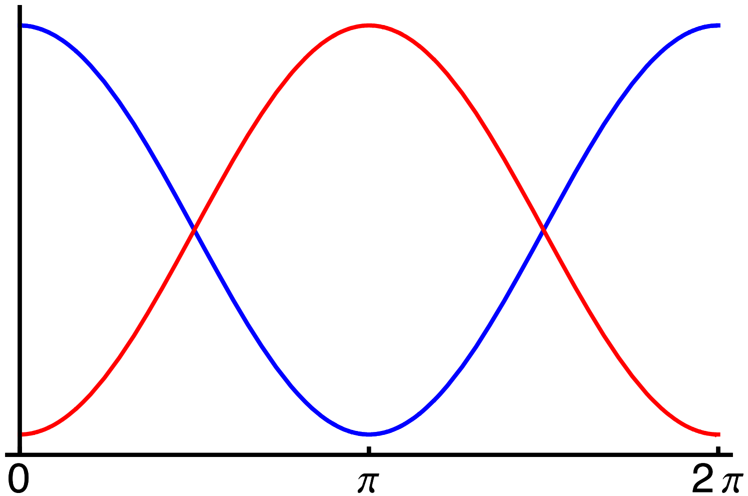
Down-sampling and aliasing



$$(x)_{\downarrow 2 \uparrow 2}[k] \quad \stackrel{z}{\longleftrightarrow} \quad \frac{1}{2}\left(X(z) + X(-z)\right)$$

Baseband component: $X(z) \quad \rightarrow \quad X(e^{j\omega})$

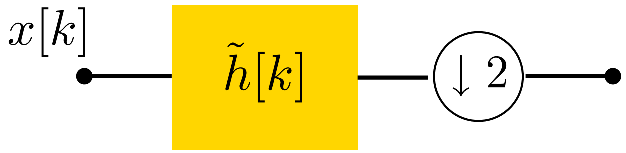
Aliased component: $X(-z) \quad \rightarrow \quad X(-e^{j\omega}) = X(e^{\pm j\pi} e^{j\omega}) = X(e^{j(\omega-\pi)})$



Hilbert-space interpretation

- ℓ_2-inner product: $\langle u, v \rangle_{\ell_2} = \sum_{k \in \mathbb{Z}} u[k] v[k]$

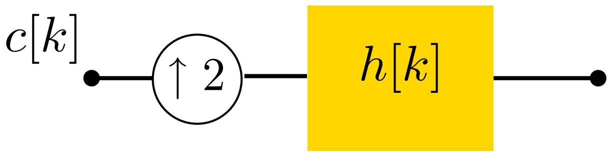
- Filtering followed by down-sampling $\Leftrightarrow$ computing ℓ_2-inner-products



$$c[m] = \sum_{k \in \mathbb{Z}} x[k] \tilde{h}[2m - k] = \langle \tilde{\varphi}_m, x \rangle_{\ell_2}$$

Equivalent analysis sequences: $\tilde{\varphi}_m[k] = \tilde{h}^\vee[k - 2m]$ with $\tilde{h}^\vee[k] = \tilde{h}[-k]$

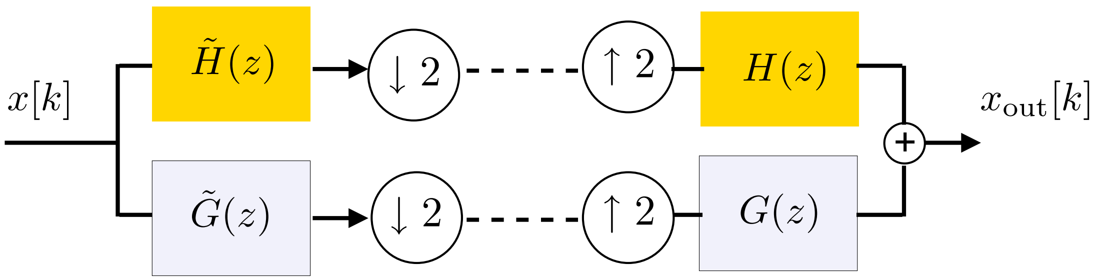
- Up-sampling followed by filtering $\Leftrightarrow$ computing signal expansion



$$x[k] = \sum_{m \in \mathbb{Z}} c[m] h[k - 2m] = \sum_{m \in \mathbb{Z}} c[m] \, \varphi_m[k]$$

Equivalent basis sequences: $\varphi_m[k] = h[k - 2m]$

Wavelet decomposition



- Analysis

$$c[m] = \langle x, \tilde{\varphi}_m \rangle_{\ell_2} \qquad \text{with} \qquad \tilde{\varphi}_m[k] = \tilde{h}^\vee[k - 2m]$$

$$d[m] = \langle x, \tilde{\psi}_m \rangle_{\ell_2} \qquad \text{with} \qquad \tilde{\psi}_m[k] = \tilde{g}^\vee[k - 2m]$$

Equivalent matrix interpretation: $\boldsymbol{y} = \tilde{\mathbf{U}}^T \boldsymbol{x}$

- Synthesis

$$x_{\text{out}} = \sum_m c[m]\varphi_m + \sum_m d[m]\psi_m \quad \text{with} \quad \varphi_m[k] = h[k - 2m] \quad \text{and} \quad \psi_m[k] = g[k - 2m]$$

Equivalent matrix interpretation: $\boldsymbol{x}_{\text{out}} = \mathbf{U}\boldsymbol{y}$

- Perfect reconstruction condition

$$\mathbf{U}\tilde{\mathbf{U}}^T = \mathbf{I} = \tilde{\mathbf{U}}^T\mathbf{U} \quad \Leftrightarrow \quad \langle \tilde{\boldsymbol{u}}_m, \boldsymbol{u}_n \rangle = \delta_{m-n}$$

Orthogonal wavelet filters

- Condition for an orthonormal transform

$$\mathbf{U}^{-1} = \mathbf{U}^T \quad \Leftrightarrow \quad \boldsymbol{u}_i = \tilde{\boldsymbol{u}}_i \quad \Leftrightarrow \quad \tilde{\varphi}_m = \varphi_m \quad \text{and} \quad \tilde{\psi}_n = \psi_n$$

- Orthogonal perfect-reconstruction filterbank

$$\begin{array}{ccc} \tilde{\varphi}_m[k] = \tilde{h}^\vee[k-2m] & = & \varphi_m[k] = h[k-2m] \\ \tilde{\psi}_m[k] = \tilde{g}^\vee[k-2m] & = & \psi_m[k] = g[k-2m] \end{array} \quad \Leftrightarrow \quad \begin{cases} \tilde{H}\left(z^{-1}\right) = H(z) & \text{(i)} \\ \tilde{G}\left(z^{-1}\right) = G(z) & \text{(ii)} \end{cases}$$

- Conjugate-quadrature filter (CQF) solution

Inspired guess: $G(z) = -z^{-1} H\left(-z^{-1}\right)$ (iii) $\Rightarrow$ $\tilde{G}(z) = -zH(-z)$

$\Rightarrow$ (PR-2) is satisfied automatically!

$$\tilde{H}(-z)H(z) + \tilde{G}(-z)G(z) = H\left(-z^{-1}\right) H(z) - H(z) H\left(-z^{-1}\right) = 0$$

(PR-1) + (i), (ii), (iii) $\Leftrightarrow$ CQF condition: $H(z)H\left(z^{-1}\right) + H(-z)H\left(-z^{-1}\right) = 2$ (iv)

$\Rightarrow$ Power-complementary filter: $\left|H\left(e^{j\omega}\right)\right|^2 + \left|H\left(e^{j(\omega-\pi)}\right)\right|^2 = 2$

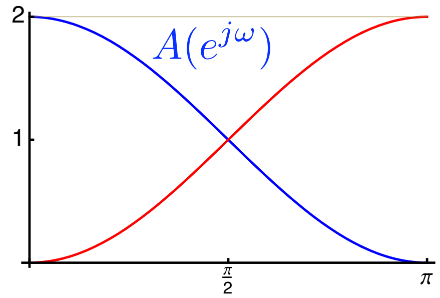
Example: Haar filterbank

- Haar filter: $H(z) = \dfrac{1+z^{-1}}{\sqrt{2}}$

- Autocorrelation: $A(z) = H(z)H\left(z^{-1}\right) = \frac{\left(1+z^{-1}\right)(1+z)}{2} = \dfrac{z+2+z^{-1}}{2}$

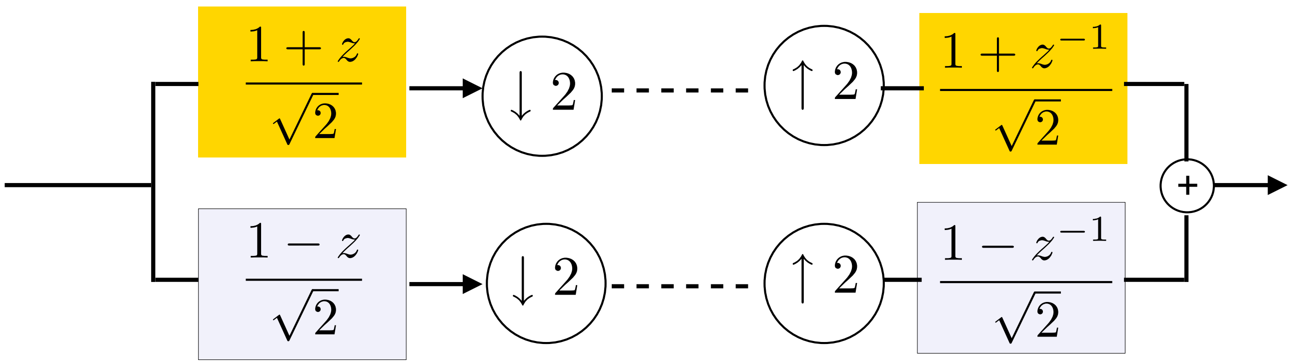
- CQF condition: $A(z) + A(-z) = 2$

$$\Leftrightarrow \quad \left|H\left(e^{j\omega}\right)\right|^2 + \left|H\left(e^{j(\omega-\pi)}\right)\right|^2 = 2$$



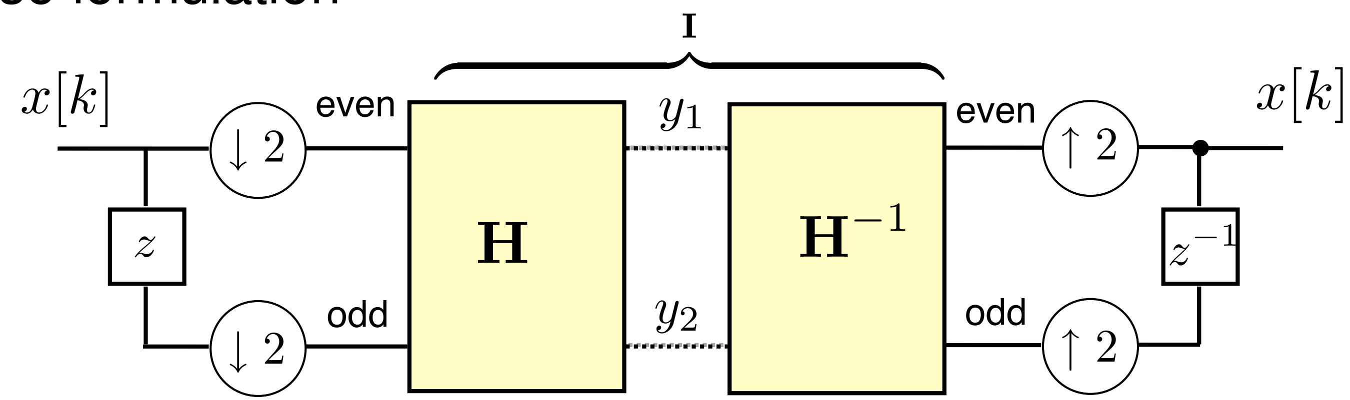
- Wavelet filter: $G(z) = -z^{-1}H\left(-z^{-1}\right) = \dfrac{1-z^{-1}}{\sqrt{2}}$ (shift and modulation)

- Haar perfect-reconstruction filterbank



Haar: polyphase formulation

- Polyphase formulation



Filterbank matrix: $\mathbf{H} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ with $\mathbf{H}^{-1} = \mathbf{H}^T = \mathbf{H}$ (orthonormal matrix)

- Simplified implementation: 2×2 block transform

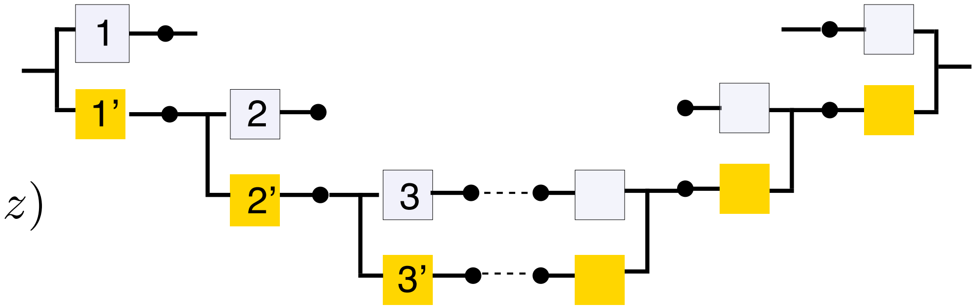
Analysis formula: $$\begin{bmatrix} y_1[k] \\ y_2[k] \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} x[2k] \\ x[2k+1] \end{bmatrix}$$

Synthesis formula: $$\begin{bmatrix} x[2k] \\ x[2k+1] \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} y_1[k] \\ y_2[k] \end{bmatrix}$$

Computing wavelet transforms

- Mallat's tree-structured filterbank algorithm
  - Universal:

    works for all wavelet transforms
  - Algorithm specification: orthogonal filter $H(z)$
  - Speed: $\mathcal{O}(N)$; depends on length of $H(z)$



- Daubechies wavelets

  Shortest orthogonal filters (CQF) such that: $H_L(z) = (1+z^{-1})^L Q(z)$

  Examples:

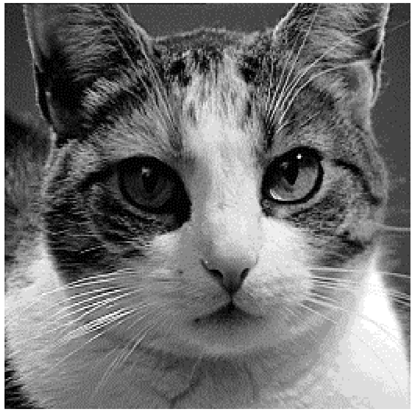
$$H_1(z) = \frac{1}{\sqrt{2}}\left(1+z^{-1}\right)$$

$$H_2(z) = \frac{1}{4\sqrt{2}}\left[\left(1+\sqrt{3}\right) + \left(3+\sqrt{3}\right)z^{-1} + \left(3-\sqrt{3}\right)z^{-2} + \left(1-\sqrt{3}\right)z^{-3}\right]$$

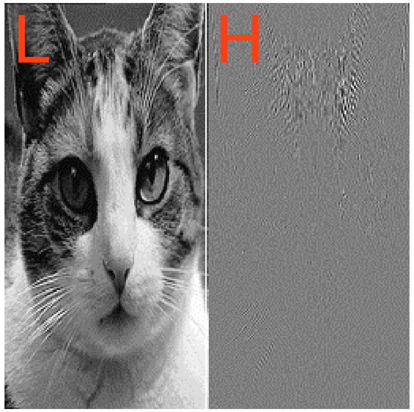
- Battle-Lemarié spline wavelets

  Orthogonal filter $H(z)$ such that the underlying functions are polynomial splines

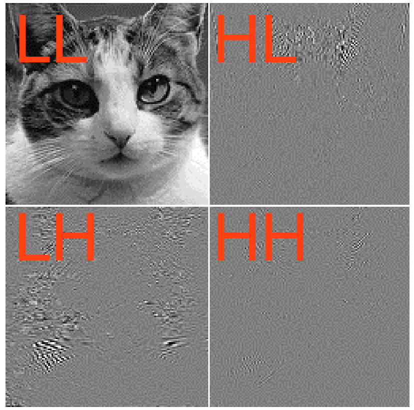
Extension to higher dimensions: separability



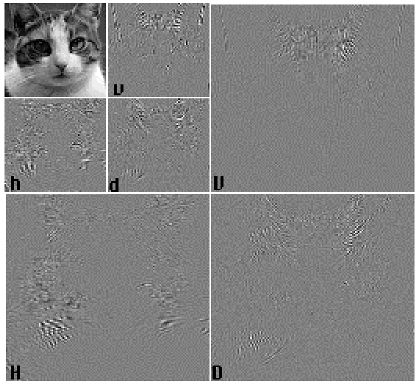
Split rows



Split columns



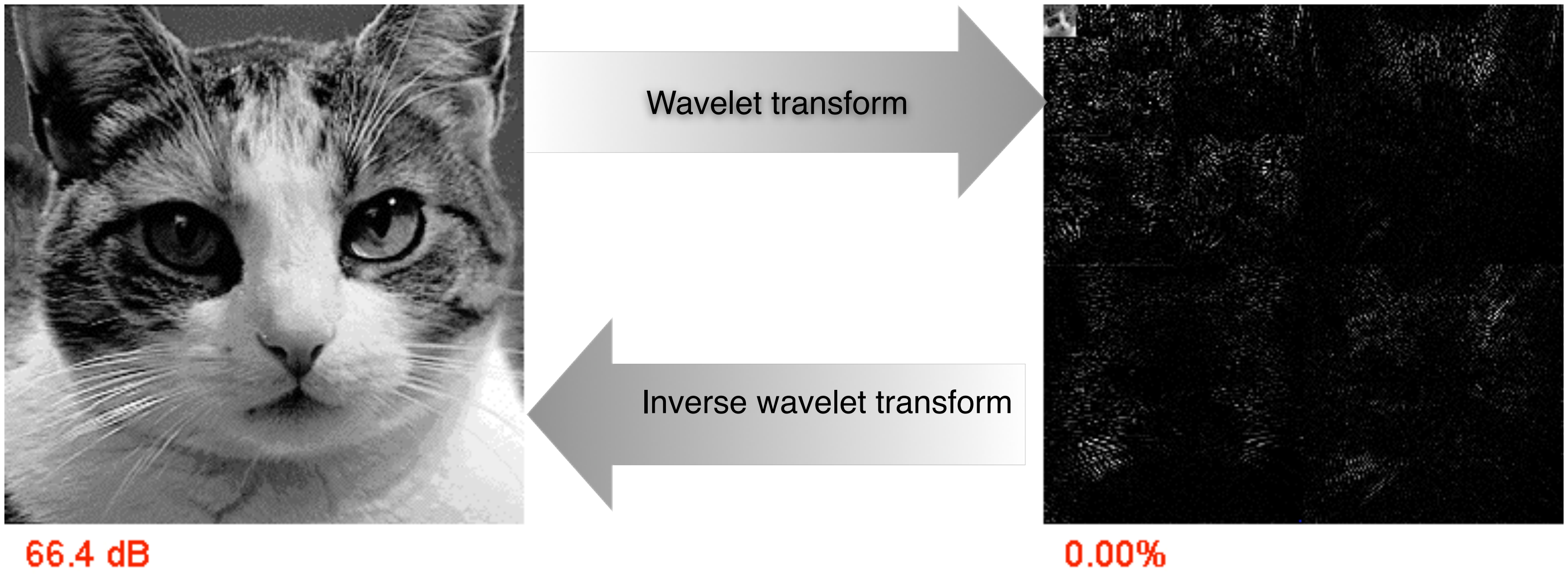
and iterate ...



Tensor-product basis functions

$$\psi_{k_1,\cdots,k_p}(x_1,\cdots,x_d) = \psi_{k_1}(x_1) \times \psi_{k_2}(x_2) \cdots \times \psi_{k_d}(x_d)$$

2D wavelet decomposition: example

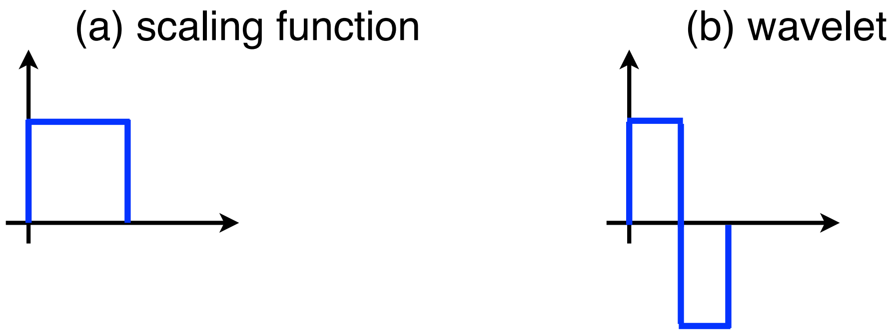


Discarding “small coefficients”

Wavelets: continuous-domain interpretation

Discrete wavelet algorithm $\leftrightarrow$ Continuous-domain interpretation

- Two key underlying functions



- Implicit definition

Solution of the two-scale relation:

$$\varphi(x/2) = \sqrt{2} \sum_{k \in \mathbb{Z}} h[k] \varphi(x - k)$$

$\Rightarrow$

Wavelet equation:

$$\psi(x/2) = \sqrt{2} \sum_{k \in \mathbb{Z}} g[k] \varphi(x - k)$$

- Existence and convergence issues

Depending on $H(z)$, the solution of the two-scale relation is not necessarily in $L_2(\mathbb{R})$

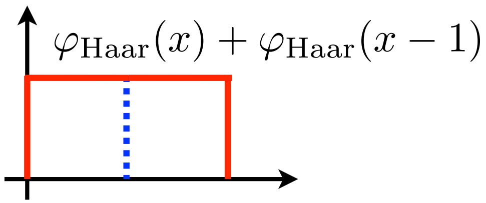
Necessary conditions: $H(z)|_{z=1} = \sqrt{2}$ and $H(z)|_{z=-1} = 0$

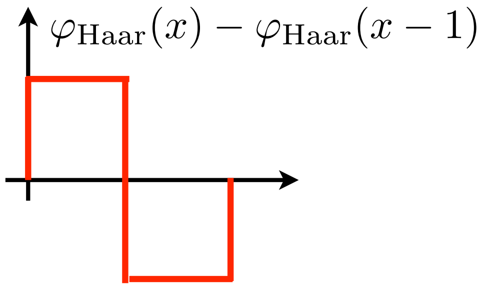
Examples of solutions of the two-scale relation

Two-scale relation: $\varphi(x/2) = \sum_{k\in\mathbb{Z}} h[k]\varphi(x-k)$ (without normalization)

Haar transform:

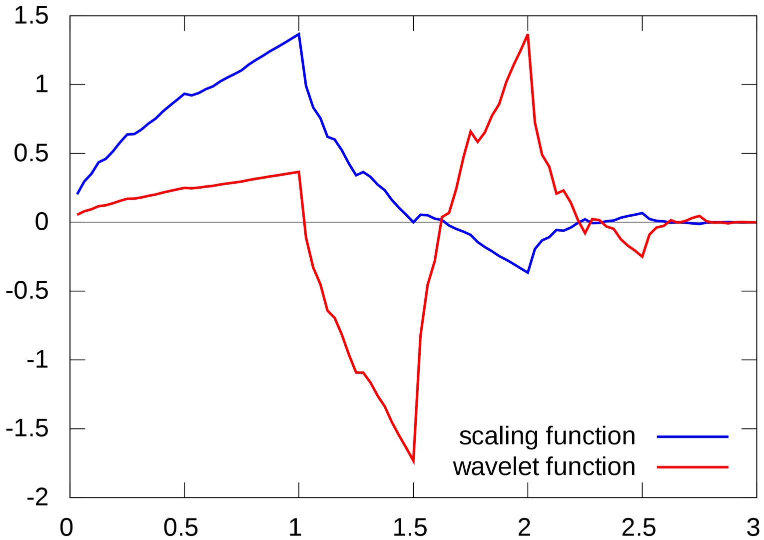
$$H_1(z) = 1 + z^{-1}$$

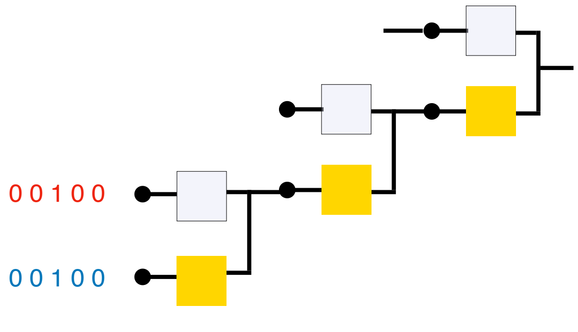




Daubechies of order 2:

$$H_2(z) = \frac{1}{4}\left[\left(1+\sqrt{3}\right) + \left(3+\sqrt{3}\right)z^{-1} + \left(3-\sqrt{3}\right)z^{-2} + \left(1-\sqrt{3}\right)z^{-3}\right]$$



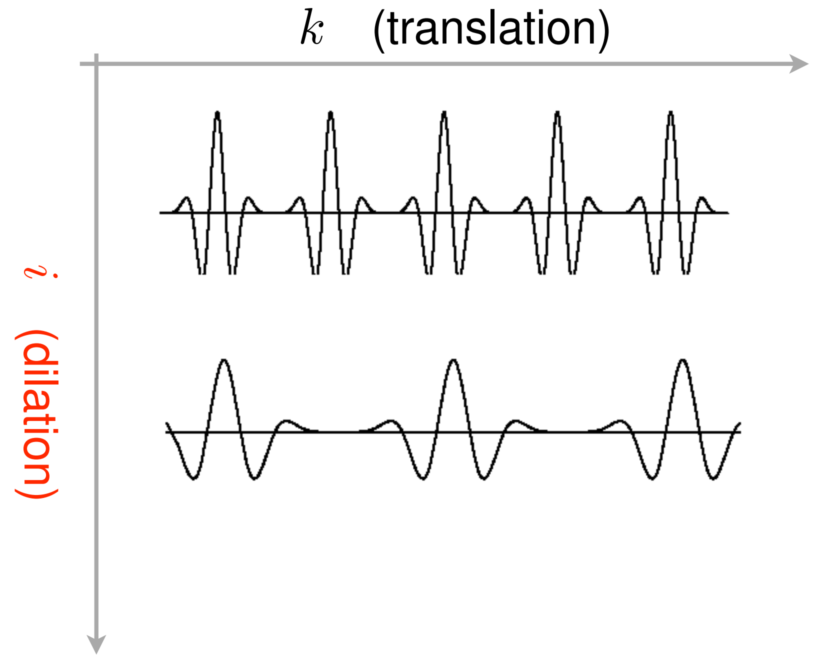


Wavelet basis of L_2

Under suitable conditions on $H(z)$: $\quad \varphi(x) \in L_2(\mathbb{R}) \quad \Rightarrow \quad \psi(x) \in L_2(\mathbb{R})$

- Wavelet basis functions

$$\psi_{i,k} = 2^{-i/2}\psi\left(\frac{x - 2^i k}{2^i}\right) \quad \text{with} \quad (i,k) \in \mathbb{Z}^2$$



- Orthogonal wavelet basis

$$\forall f \in L_2(\mathbb{R}), \quad f(x) = \sum_{i\in\mathbb{Z}}\sum_{k\in\mathbb{Z}} \langle f, \psi_{i,k}\rangle\, \psi_{i,k}(x) \qquad \text{with} \qquad \langle \psi_{i,k}, \psi_{j,l}\rangle = \delta_{i-j,k-l}$$

- Biorthogonal wavelet basis

$$\forall f \in L_2(\mathbb{R}), \quad f(x) = \sum_{i\in\mathbb{Z}}\sum_{k\in\mathbb{Z}} \langle f, \tilde{\psi}_{i,k}\rangle\, \psi_{i,k}(x) \qquad \text{with} \qquad \langle \tilde{\psi}_{i,k}, \psi_{j,l}\rangle = \delta_{i-j,k-l}$$

Wavelets: further probing

- Application areas
  - Signal/image processing: coding, multiscale processing
  - Pattern recognition/analysis: texture, time/frequency analysis
  - Imaging
  - Computer graphics
  - Applied mathematics: PDEs, approximation theory
  - Medicine and biology
  - Physics turbulence, fractals...

- Mathematical wavelet theory
  - Convergence of two-scale relation
  - Vanishing moments
  - Multiscale derivatives
  - Characterization of singularities
  - Reproduction of polynomials
  - Approximation theory
  - Regularity
  - Unconditional basis: Sobolev and Besov spaces
  - **Sparsity** $\rightarrow$ compressed sensing

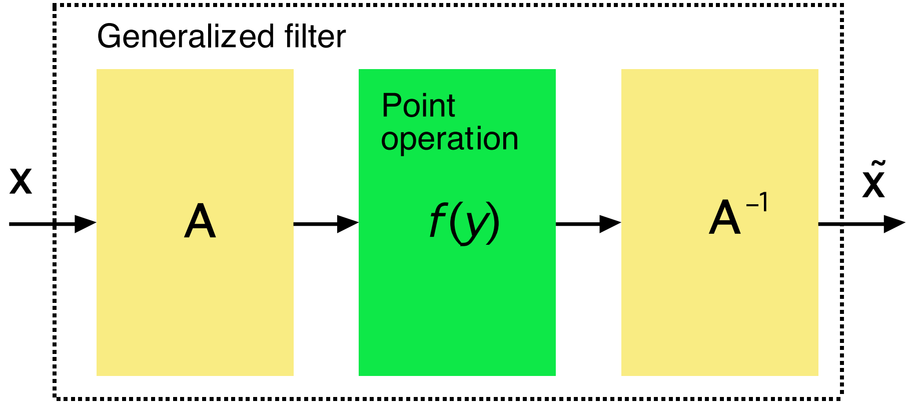
8.5 GENERALIZED FILTERING

Principle: go to the transform domain to simplify processing

Image transform: $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$

Scalar processing: $\tilde{y}_i = f_i(y_i)$

Inverse transform: $\tilde{\boldsymbol{x}} = \boldsymbol{A}^{-1}\tilde{\boldsymbol{y}}$



- Linear algorithm

$\tilde{y}_i = \alpha_i y_i$

Example: filtering in the Fourier domain $\Rightarrow$ $\alpha_i = H\left(e^{j\boldsymbol{\omega}_i}\right)$

- Pointwise non-linearities

Global non-linearity: $\tilde{y}_i = f(y_i)$

Grouping of coefficients in subbands: $\tilde{y}_i = f_j(y_i)$ for $i \in I_j$ (jth band)

Generalized Wiener filter

- Justification for component-wise processing
  - Orthogonal transform: white noise $\Rightarrow$ white noise
    $$\boldsymbol{C}_x = \sigma^2 \boldsymbol{I} \qquad \Rightarrow \qquad \boldsymbol{C}_y = \boldsymbol{A}\boldsymbol{C}_x\boldsymbol{A}^T = \sigma^2 \boldsymbol{A}\boldsymbol{A}^T = \sigma^2 \boldsymbol{I}$$
  - Sinusoidal transforms tend to decorrelate stationary processes
    (asymptotic equivalence with KLT $\Rightarrow$ diagonal covariance)
  - Wavelet transforms tend to whiten self-similar stochastic processes (fractals)

- Signal + noise model: $y_i = s_i + n_i$

- Linear estimator: $\tilde{s}_i = \alpha_i y_i$

- Generalized Wiener filter: $\alpha_i = \dfrac{E\left\{s_i^2\right\}}{E\left\{s_i^2\right\} + E\left\{n_i^2\right\}} \leq 1$

Refresher: MMSE (or Wiener) estimator

Pointwise linear estimator: $\tilde{s} = \alpha y$ with $y = s + n$ (signal + noise)

Mean-square estimation error

$$\begin{aligned}
\varepsilon^2 &= E\left\{(\tilde{s} - s)^2\right\} = E\left\{[\alpha(s+n) - s]^2\right\} \\
&= \alpha^2 E\left\{(s+n)^2\right\} + E\left\{s^2\right\} - 2\alpha\Big(E\left\{s^2\right\} + E\left\{sn\right\}\Big) \\
&= \alpha^2 E\left\{(s+n)^2\right\} + E\left\{s^2\right\} - 2\alpha E\left\{s^2\right\}
\end{aligned}$$

Hypothesis: signal and noise are uncorrelated ($E\{sn\} = 0$)

Minimum-mean-square-error solution

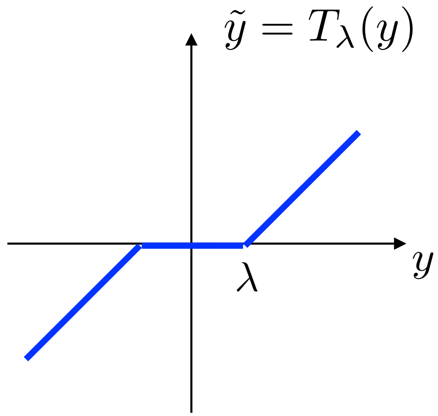
$$\frac{\partial \varepsilon^2}{\partial \alpha} = 0 \qquad \Rightarrow \qquad \alpha = \frac{E\left\{s^2\right\}}{E\left\{(s+n)^2\right\}} = \frac{E\left\{s^2\right\}}{E\left\{s^2\right\} + E\left\{n^2\right\}}$$

Simple wavelet denoising

- Basic idea
  - Orthogonal WT: white noise $\rightarrow$ white noise
  - Signal concentrated in few coefficients, while noise is spread-out evenly

$\Rightarrow$ Noise attenuation is achieved by simple wavelet shrinkage/thresholding

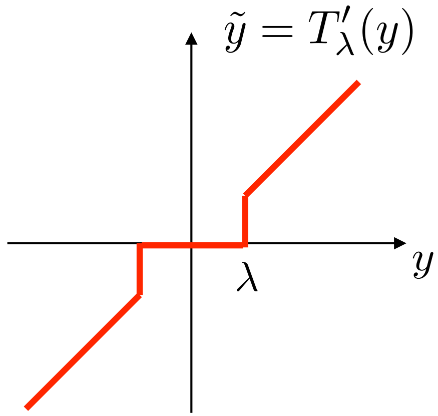
- Soft threshold



$$T_\lambda(y) = \begin{cases} y - \lambda, & y > \lambda \\ 0, & |y| \leqslant \lambda \\ y + \lambda, & y < -\lambda \end{cases}$$

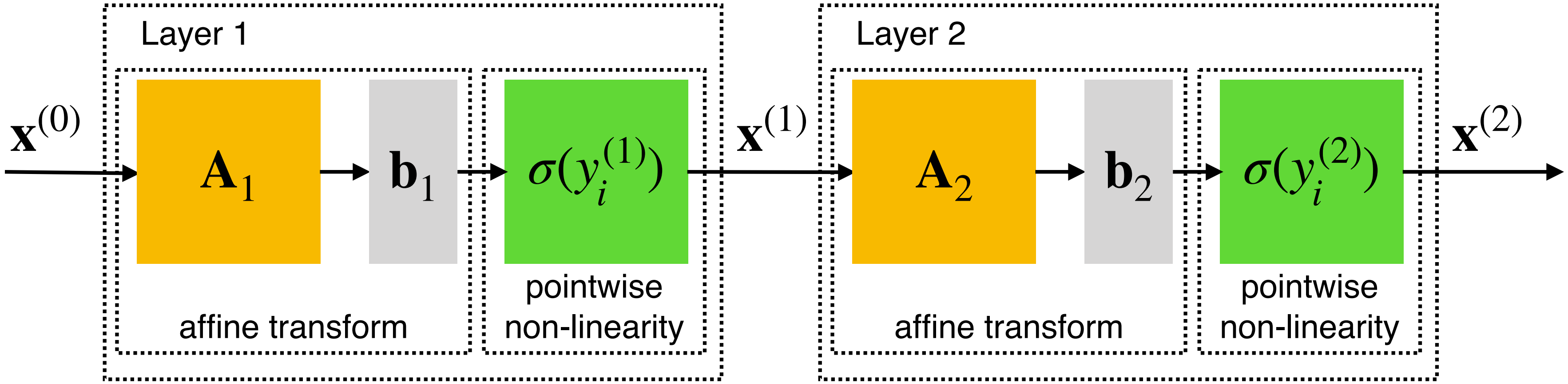
In practice: $\lambda = C \cdot \sigma$ (proportional to noise standard deviation)

- Hard threshold



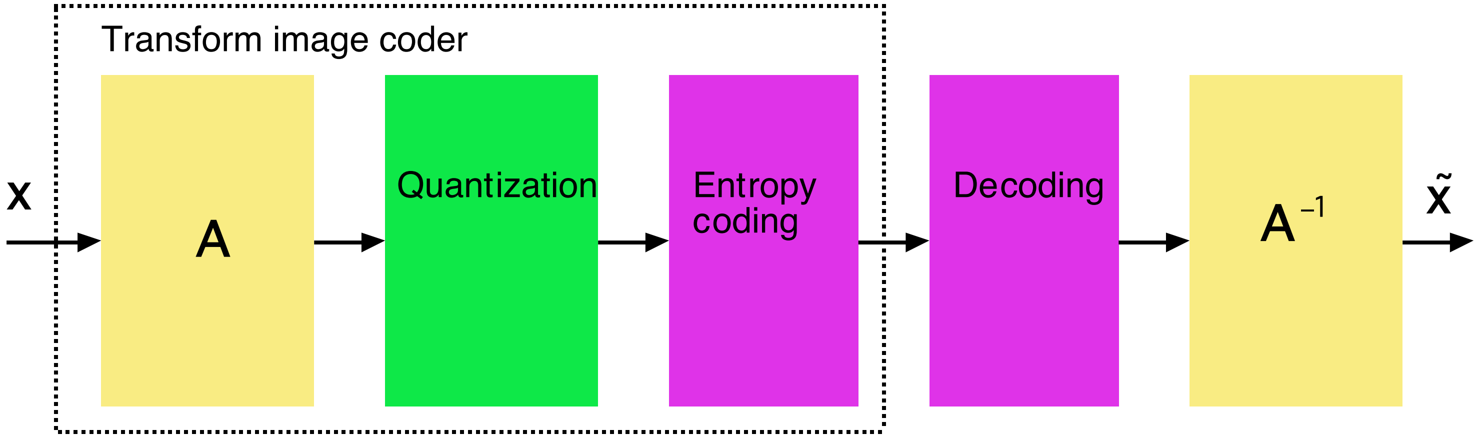
$$T'_\lambda(y) = \begin{cases} y, & |y| > \lambda \\ 0, & |y| \leqslant \lambda \end{cases}$$

Building block for deep neural networks



- Deep neural network with L layers and input $\mathbf{x} = \mathbf{x}^{(0)}$
  $\mathbf{f}_{\text{deep}}(\mathbf{x}) = (\boldsymbol{\sigma}_L \circ \mathscr{A}_L \circ \ldots \boldsymbol{\sigma}_1 \circ \mathscr{A}_1)(\mathbf{x})$
- Affine layers
  $\mathscr{A}_l : \mathbb{R}^{N_{l-1}} \to \mathbb{R}^{N_l}$ such that $\mathscr{A}_l(\mathbf{x}^{(l-1)}) = \mathbf{y}^{(l)} = \mathbf{A}_l \mathbf{x}^{(l-1)} + \mathbf{b}_l$
  - Linear transform with trainable weights $\mathbf{A}_l \in \mathbb{R}^{N_l \times N_{l-1}}$
  - Bias $\mathbf{b}_l \in \mathbb{R}^{N_l}$
- Activation function
  $\boldsymbol{\sigma}_l : \mathbb{R}^{N_l} \to \mathbb{R}^{N_l}$ with $\boldsymbol{\sigma}_l(\mathbf{y}) = (\sigma(y_1), \ldots, \sigma(y_{N_l}))$
  $\mathbf{x}^{(l+1)} = \boldsymbol{\sigma}_l(\mathbf{y}^{(l)})$

8.6 IMAGE CODING



- DCT-based coding (JPEG)
  - Industry standard
  - Good for low distortion coding

    but...
  - Not so good at lower bit rates

    $\Rightarrow$ Blocking artifacts!

- Wavelet coding (EZW, JPEG2000)
  - Multiresolution idea
  - Overlapping basis functions
  - Progressive transmission

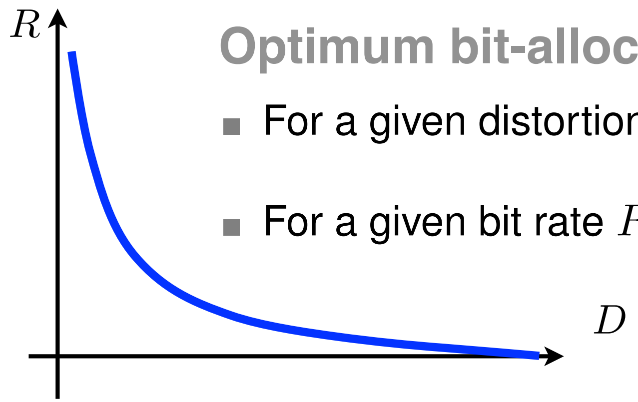
Coding: general considerations

- General principle: transform signal to pack energy and to decorrelate

Transform	$\boldsymbol{y} = \boldsymbol{A} \cdot \boldsymbol{x}$
Scalar quantization	$\tilde{y}_i = Q(y_i)$
Encoding	$\bar{R} = \frac{1}{N} \sum_i R_{\tilde{y}_i}$ (average bit rate)
Reconstruction	$\tilde{\boldsymbol{x}} = \boldsymbol{A}^{-1} \tilde{\boldsymbol{y}}$

- Reconstruction error: $D = \|\boldsymbol{x} - \tilde{\boldsymbol{x}}\|^2 = \|\boldsymbol{y} - \tilde{\boldsymbol{y}}\|^2 \quad \Leftrightarrow \quad$ $\boldsymbol{A}$ is unitary

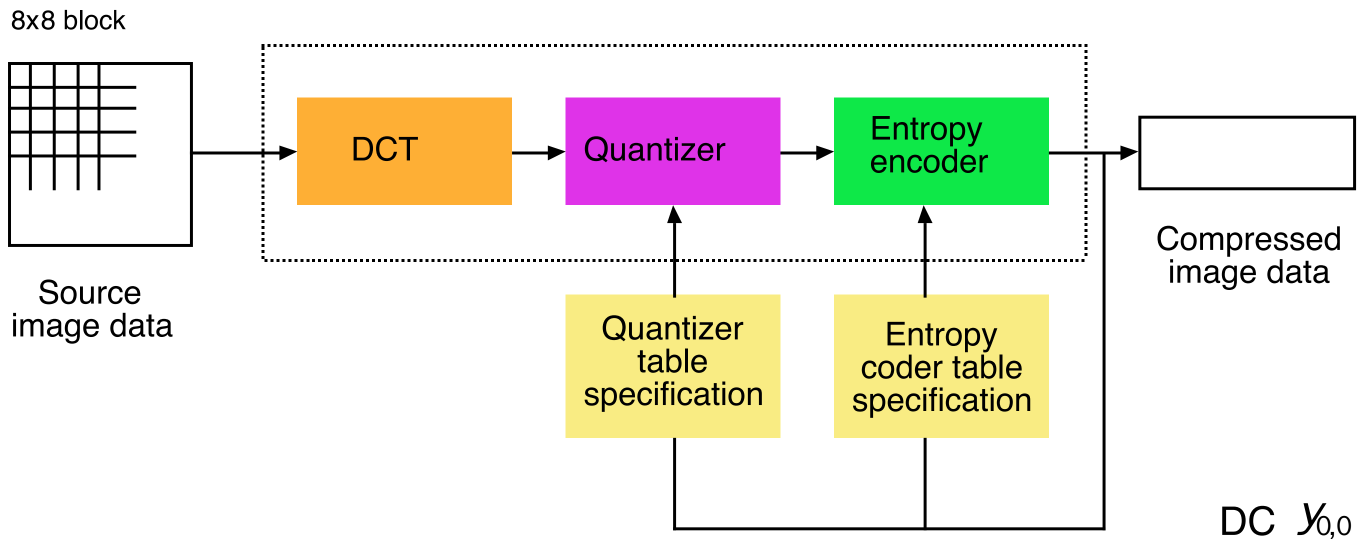
- Rate-distortion (R-D) curve

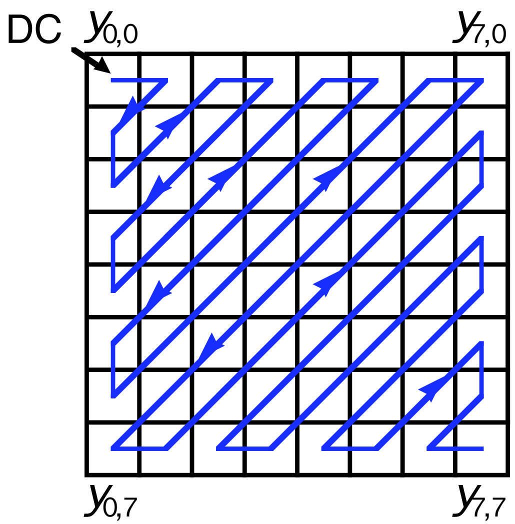


Optimum bit-allocation problem

- For a given distortion D, minimize the bit rate R
- For a given bit rate R, minimize the distorsion D

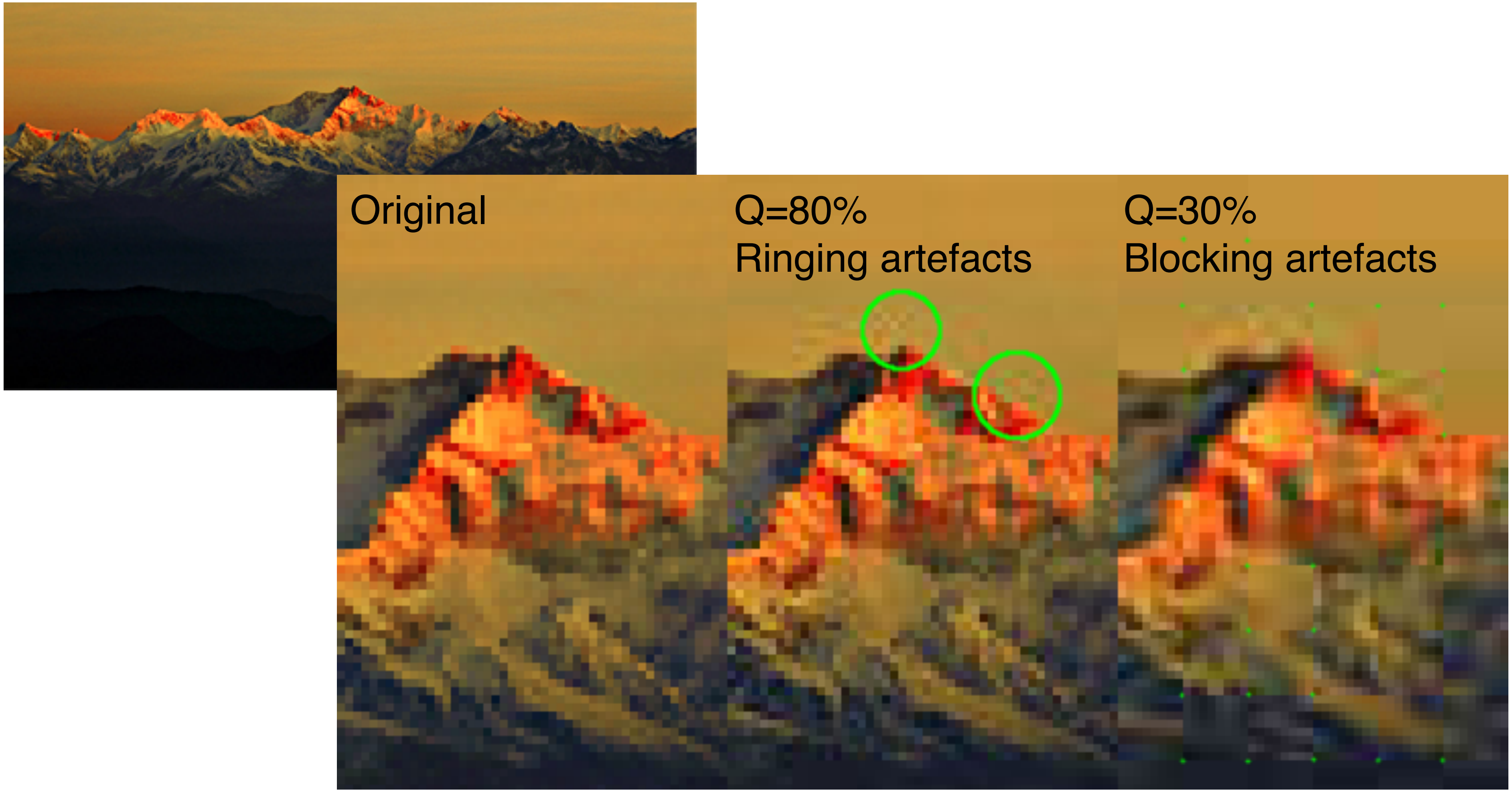
JPEG image coder





- Quantization
  - perceptual weighting
- Entropy coding
  - zig-zag scanning of 8×8 DCT coefficients
  - combined (Run, Size) Huffman code (Run=run length of preceding zeros)
  - End-of-block (EOB) symbol

JPEG compression artefacts



- **Ringing artefacts**: DCT basis vectors become visible around high-contrast edges due to quantization of smaller coefficients
- **Blocking artefacts**: block-wise processing leads to visible edges between blocks at high compression ratios

Compression: Why wavelets?

- Reason 1: The WT produces a lot of clustered near-zero coefficients
  - At low bit rate, the gain of transform coding is mostly due to the “zero bin”

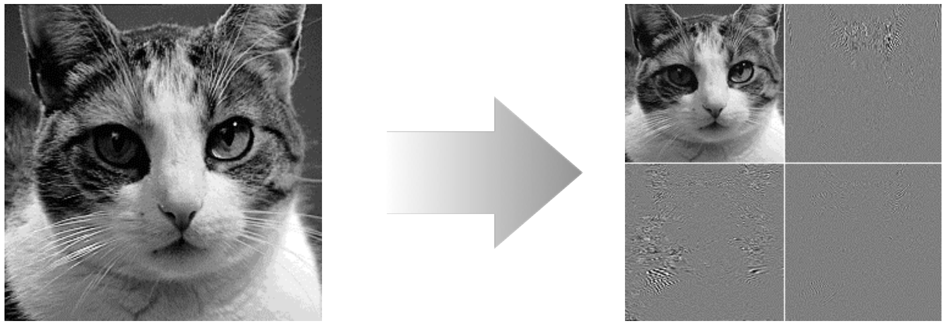
    $\Rightarrow$ Dead-zone quantizer
  - Coding of “zeros” is *non-stationary* $(< 1$ bit per coefficient) and must take advantage of spatial correlation

    $\Rightarrow$ Position coding is the key (run-length, zero-tree)
- Reason 2: Reduced blocking artifacts
  - JPEG/DCT works well at high/medium bit rate but produces blocking artifacts at lower rates
  - Blocking is reduced with the WT because the basis functions are overlapping
- Reason 3: The WT has a built-in multiresolution structure
  - Progressive transmission (by successive approximation)
  - Browsing, the WEB
  - Unequal error protection

Simple wavelet coding

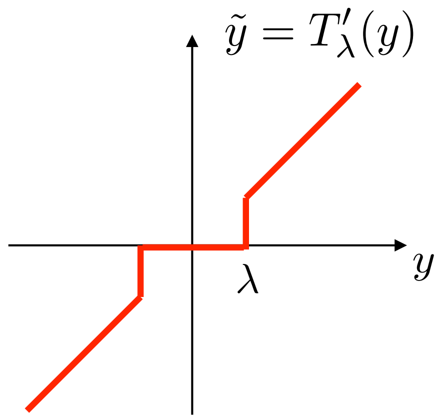
- Basic idea

  WT produces lots of small, near-zero coefficients in slowly-varying regions



$\Rightarrow$ Compression is achieved by setting small coefficients to zero

- Hard threshold



$$T'_\lambda(y) = \begin{cases} y, & |y| > \lambda \\ 0, & |y| \leqslant \lambda \end{cases}$$

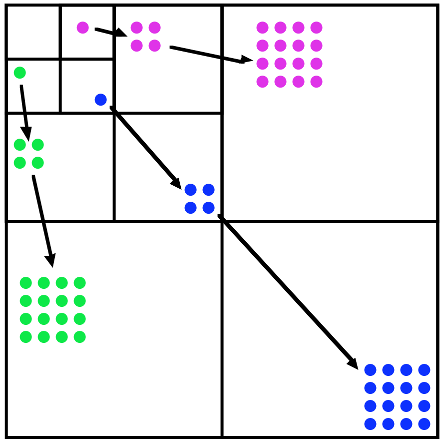
- Qualitative behavior of approx. error for $M = N_{\text{tot}} - N_{\text{discard}}$ coefficients

$$D = \sum_{|y|<\lambda} |y_i|^2 \propto \underbrace{M^{-1}}_{\text{1D linear}}, \underbrace{2^{-M}}_{\text{1D non-linear}}, \underbrace{M^{-\gamma}}_{\text{2D edges non-linear}}$$

EZW coder

- Embedded zero-tree wavelet coder (EZW)

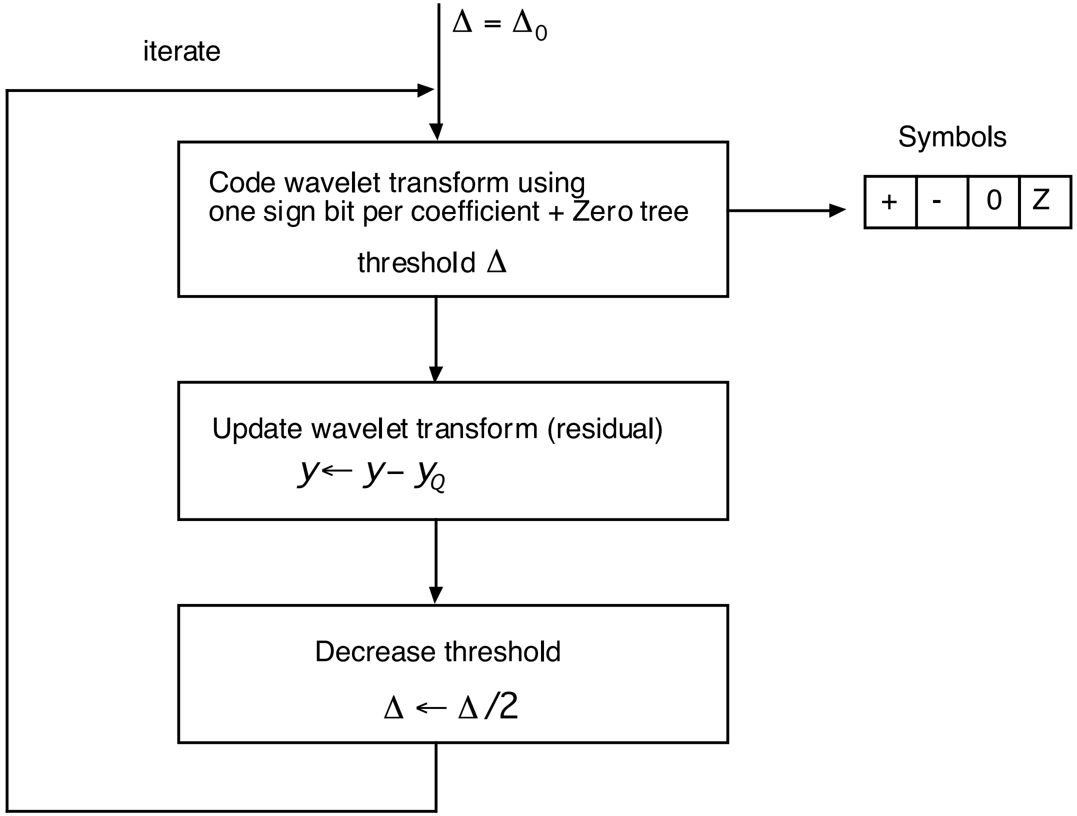
  - Special status of "zero bin".
  - Exploits spatial correlation ⇒ Zero tree



- Conceptual principle

  - abs(coefficient) > Δ :
    positive: **+**
    negative: **-**

  - abs(coefficient) < Δ :
    parent zero: do nothing
    parent non-zero, look descendants:
    descendents zero: **Z**
    descendents non-zero: **0**



8.7 SUMMARY

- Image transforms are used for data processing, data compression, and data analysis. To be truly useful, they must have a fast algorithm.
- An orthonormal transform is characterized by a unitary transform matrix $\boldsymbol{A} = \begin{bmatrix} \boldsymbol{u}_1 & \cdots & \boldsymbol{u}_N \end{bmatrix}^H$ where the $\boldsymbol{u}_i$'s are the basis vectors of the representation.
- The variances in the transformed domain are determined by the covariance matrix of the input vector: $\sigma_i^2 = \boldsymbol{u}_i^T \boldsymbol{C}_x \boldsymbol{u}_i$.
- The Karhunen-Loève transform is specified by the eigenvectors of $\boldsymbol{C}_x$. The KLT compacts the energy in the fewest number of components and decorrelates the data.
- The discrete Fourier transform (DFT) diagonalizes circulant matrices.
- Separable transforms are implemented by successive 1D processing of the rows and columns of an image.
- Useful sinusoidal transforms are the DFT and the DCT; for stationary processes, they are both asymptotically equivalent to the KLT.

- The main tool for the wavelet transform is a perfect reconstruction filterbank which splits a signal into orthogonal lowpass plus highpass components. The splitting is iterated to yield a tree-structured transform.
- A wavelet transform is entirely specified by an orthogonal filter; the highpass filter is obtained by modulation (QMF).
- Generalized filtering is achieved by applying a suitable point function in the transformed domain prior to reconstruction. The operation is either linear (e.g., Wiener filter) or non-linear (e.g., soft-thresholding).
- Transform coding applies a uniform quantization in the transformed domain. JPEG uses 8×8 block DCTs; more recent coders use the wavelet transform (EZW, JPEG2000).
- The key to high compression is the efficient encoding of zero values: run length in JPEG or zero-tree in EWZ.

Further reading

- Classical transforms
  - A.K. Jain, *Fundamentals of digital image processing.* Englewood Cliffs, NJ: Prentice-Hall, 1989; chapter 5.
  - M. Unser, “On the Approximation of the Discrete Karhunen-Loève Transform for Stationary Processes”, *Signal Processing*, vol. 7, no. 3, pp. 231-249, December 1984.
- Wavelet basics
  - B. Burke Hubbard, *The world according to wavelets*, A K Peters, Wellesley, MA, 1998.
  - S. Mallat, *A wavelet tour of signal processing*. San Diego: Academic Press, 1998.
  - G. Strang and T. Nguyen, *Wavelets and filter banks*. Wellesley, MA: Wellesley-Cambridge, 1996.
  - M. Vetterli and J. Kovacevic, *Wavelets and Subband Coding*. Englewood Cliffs, NJ: Prentice Hall, 1995.
- Wavelet coding and denoising
  - J. Shapiro, “Embedded image coding using zerotrees of wavelet coefficients”, *IEEE Trans. Acoustics, Speech and Signal Processing*, vol. 41, no. 12, pp. 3445-3462, 1993.
  - D.L. Donoho,“De-noising by soft-thresholding”, *IEEE Trans. Information Theory*, vol. 41, no. 3, pp. 613-627, 1995.
  - A. Chambolle, R.A. DeVore, N.-Y. Lee and B.J. Lucier, “Nonlinear wavelet image processing: variational problems, compression, and noise removal through wavelet shrinkage”, *IEEE Trans. Image Processing*, vol. 7, no. 33, pp. 319-335, 1998.