Image transforms

CGG MFF UK Praha

pepca@cgg.mff.cuni.cz

http://cgg.mff.cuni.cz/~pepca/

2D image transform

- **data decorrelation**
  – output coefficients are less correlated (dependent)
- **spectral analysis** and synthesis
  – spectral space: frequencies contained in a signal (image, audio)
- often **orthogonal** (or unitary) **transforms**
  – projections to some orthogonal (unitary) basis
- **continuous or discrete** forms of a transform

Inner product space

Inner product $\langle *,* \rangle$:

- real or complex **functions** defined on **T**:

$$\langle f,g \rangle = \int_T f(x) \cdot g(x)\ dx$$

- real or complex **sequences**:

$$\langle a,b \rangle = \sum_i a_i \cdot b_i$$

- real or complex **matrices**:

$$\langle A,B \rangle = \sum_{i,j} a_{ij} \cdot b_{ij}$$

Orthogonal system

System $\mathbf{U} = \{ \mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \ldots \}$ is **orthogonal**, if:

$$\langle \mathbf{u}_m, \mathbf{u}_n \rangle = \begin{cases} \mathbf{c}_m > 0 & \text{for } \mathbf{m} = \mathbf{n} \\ 0 & \text{else} \end{cases}$$

System $\mathbf{U}$ is **orthonormal**, if:

$$\langle \mathbf{u}_m, \mathbf{u}_n \rangle = \delta(\mathbf{m} - \mathbf{n})$$

Complete orthogonal system

Orthogonal system $\mathbf{U} = \{ \mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \ldots \}$ is **complete**, if:

1) in a **finite dimension space** it is a **basis**

2) in a **infinite dimensional space** every point $\mathbf{a}$ can be approximated with arbitrary precision by a sum:

$$\mathbf{a} \cong \sum_{i=1}^{N} \mathbf{A}_i \cdot \mathbf{u}_i$$

Orthogonal transform

Computing coefficients for approximation by an orthogonal system:

$$\min_{A_i} \left\{ \left\| a - \sum_{i=1}^{N} A_i \cdot u_i \right\| \right\}$$

Optimal coefficient:

$$A_i = \frac{1}{c_i} \langle a, u_i \rangle$$

Transforme T:

$$a \rightarrow \{ A_i \}$$

Separable 2D transform

- discrete transform in a $\mathbf{R}^{M \times N}$ space needs $\mathbf{O(M^2N^2)}$ multiplications and additions
  - **M×N** coefficients, **M×N** multiplications for each one

$$A_{ij} = \sum_{k=1}^{M} \sum_{l=1}^{N} a(k,l) \cdot u_{ij}(k,l) \qquad \forall i,j$$

- if the system **U is separable**, only $\mathbf{O(MN(M+N))}$ operations is needed

$$u_{ij}(k,l) = v_i(k) \cdot w_j(l)$$

Separable 2D transform

- systems $\{ \mathbf{v}_i \}$ and $\{ \mathbf{w}_j \}$ has to be **orthogonal as well**
- for square matrices, $\mathbf{v}_i = \mathbf{w}_i$ is often used
- **separable transform** is computed in two steps
  – one step transforms columns, the other rows
  – matrix notation:

$$\mathbf{A} = \mathbf{V} \cdot \mathbf{a} \cdot \mathbf{W} = \mathbf{A}^{\mathbf{V}} \cdot \mathbf{W} = \mathbf{V} \cdot \mathbf{A}^{\mathbf{W}}$$

Derivation

$$A_{ij} = \sum_{k=1}^{M}\sum_{l=1}^{N} a(k,l)\cdot u_{ij}(k,l) = \qquad \forall i,j$$

$$= \sum_{k=1}^{M}\sum_{l=1}^{N} a(k,l)\cdot v_i(k)\cdot w_j(l) =$$

$$= \sum_{k=1}^{M} v_i(k)\cdot \sum_{l=1}^{N} a(k,l)\cdot w_j(l) = \sum_{k=1}^{M} v_i(k)\cdot A_j^{w}(k)$$

$$= \sum_{l=1}^{N} w_j(l)\cdot \sum_{k=1}^{M} a(k,l)\cdot v_i(k) = \sum_{l=1}^{N} w_j(l)\cdot A_i^{v}(l)$$

Complex Fourier series

$$\left\{ \exp\left(\frac{2\pi nt}{T_0} i\right) = \cos\left(\frac{2\pi nt}{T_0}\right) + i \cdot \sin\left(\frac{2\pi nt}{T_0}\right) \right\}_{n=-\infty}^{\infty}$$

Complex function **g** with period T_0:

$$g(t) = \sum_{n=-\infty}^{\infty} A_n \cdot \exp\left(\frac{2\pi nt}{T_0} i\right)$$

$$A_n = \frac{1}{T_0} \int_{-T_0/2}^{T_0/2} g(t) \cdot \exp\left(-\frac{2\pi nt}{T_0} i\right) dt$$

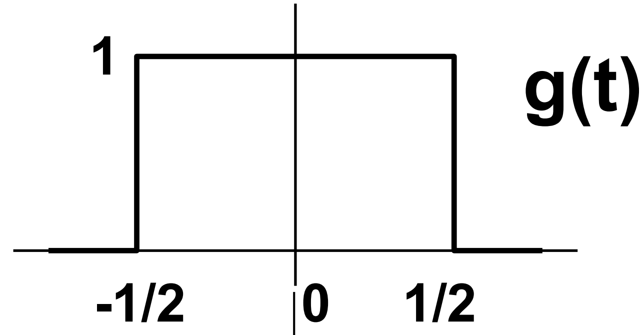
Continuous Fourier transform

Complex function **g(t)** with finite energy →
→ complex **spectral function G(f)**

$$G(f) = \int_{-\infty}^{\infty} g(t) \cdot \exp(-2\pi f t i)\ dt$$

$$g(t) = \int_{-\infty}^{\infty} G(f) \cdot \exp(2\pi f t i)\ df$$

Example of continuous FT

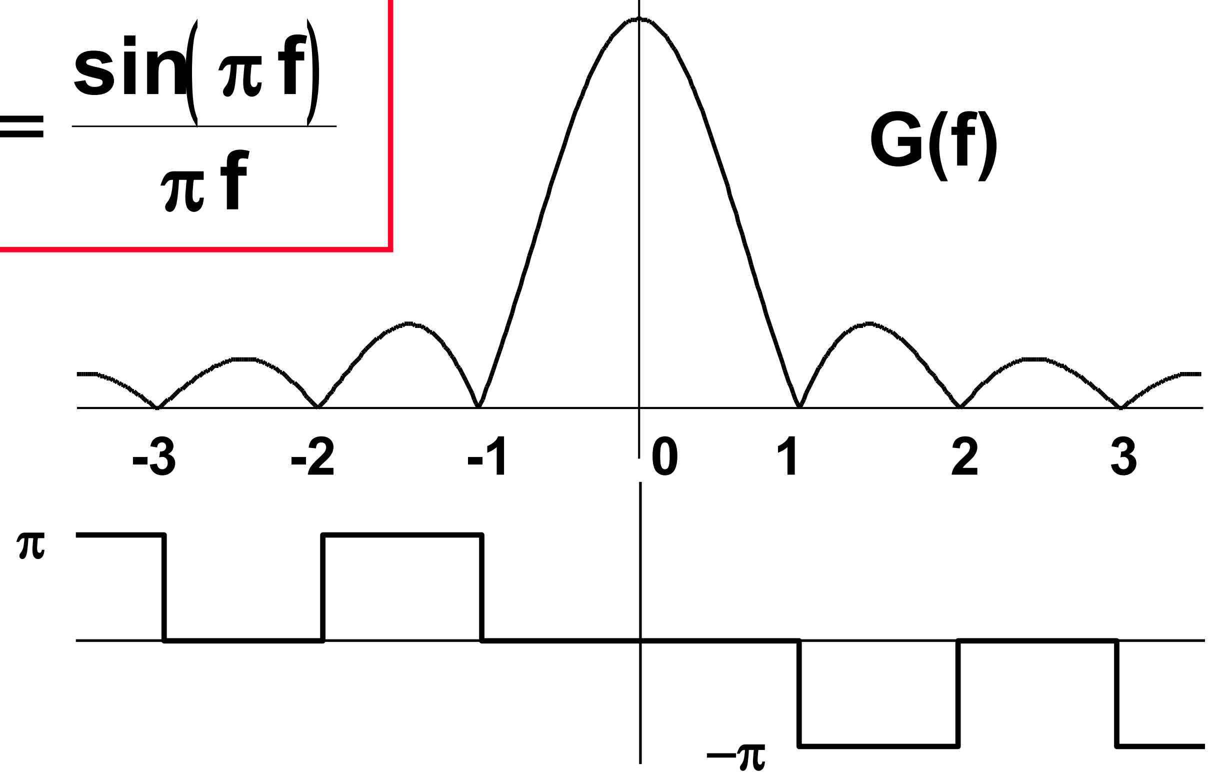


g(t) – rectangular pulse:

$$G(f) = \mathrm{sinc}(\pi f) = \frac{\sin(\pi f)}{\pi f}$$

amplitude:

phase:



Discrete Fourier transform

Basis: $$\left\{ f_k(n) = \frac{1}{\sqrt{N}} \exp\left(\frac{2\pi kn}{N} i\right); \quad 0 \le k,n < N \right\}$$

Unitary transform:

$$G(k) = \sum_{n=0}^{N-1} g(n) \cdot f_k(-n)$$

$$g(n) = \sum_{k=0}^{N-1} G(k) \cdot f_k(n)$$

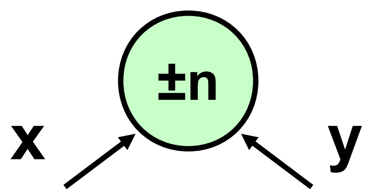
Fast algorithms for DFT

- make use of a property of **complex conjugates**:

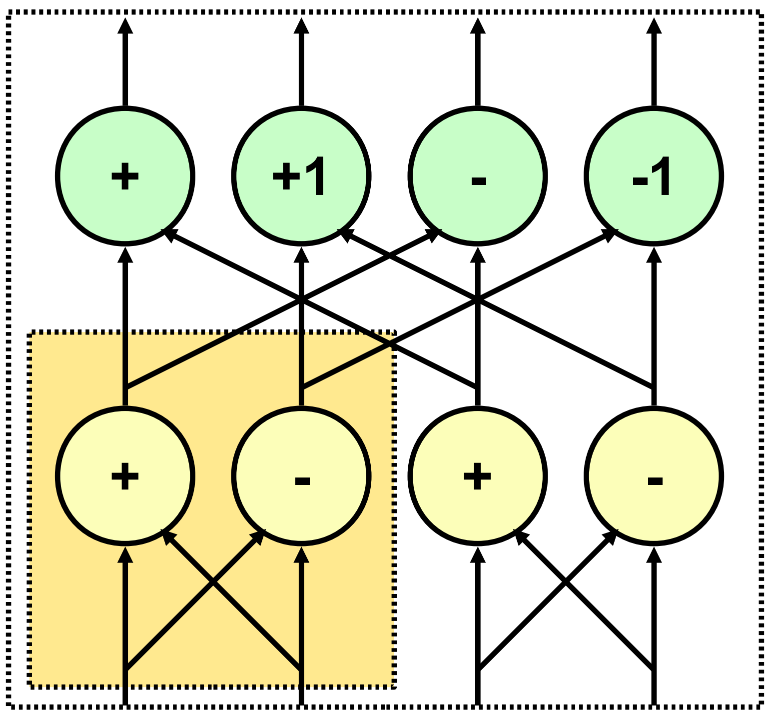
$$G\left(\frac{N}{2}+k\right)=\overline{G}\left(\frac{N}{2}-k\right)$$

- „**divide and conquer**” principle (Cooley ‘65) $\mathbf{FFT_N}$:
  – separate odd and even coefficients (**N=2M**)
  – each group is computed separately by the $\mathbf{FFT_M}$
  – result is obtained using **N** operations
- complexity of 1D FFT is $\mathbf{O(N \log_2 N)}$, using parallel HW $\mathbf{O(\log_2 N)}$

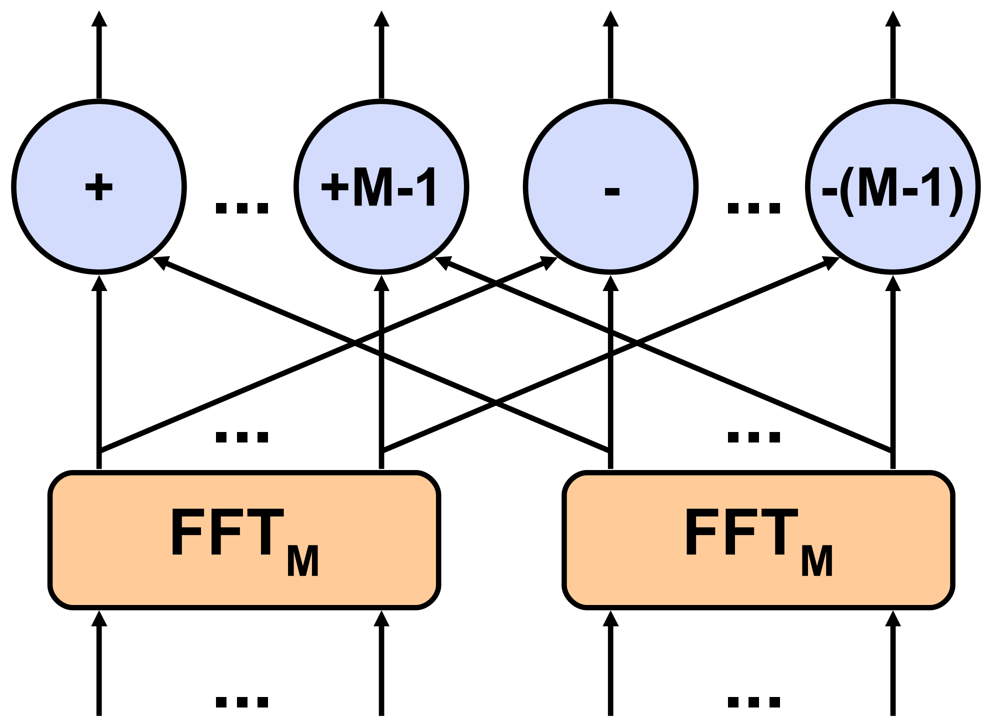
Fast DFT (FFT) scheme



$$x \pm y \cdot \exp\left(-\frac{2\pi n}{M} i\right)$$



2 → 4



Discrete sine transform

Suitable for image data with correlation coefficient **< 0.5**

Basis: $\left\{ s_k(n) = \sqrt{\frac{2}{N+1}} \sin \frac{\pi(k+1)(n+1)}{N+1}; \quad 0 \le k,n < N \right\}$

Unitary transform:

$$G(k) = \sum_{n=0}^{N-1} g(n) \cdot s_k(n)$$

$$g(n) = \sum_{k=0}^{N-1} G(k) \cdot s_k(n)$$

Discrete cosine transform

For image data with high correlation coefficients

Basis: $\left\{ C_k c_k(n) = C_k \cos \frac{\pi(2n+1)k}{2N}; \quad 0 \le k,n < N \right\}$

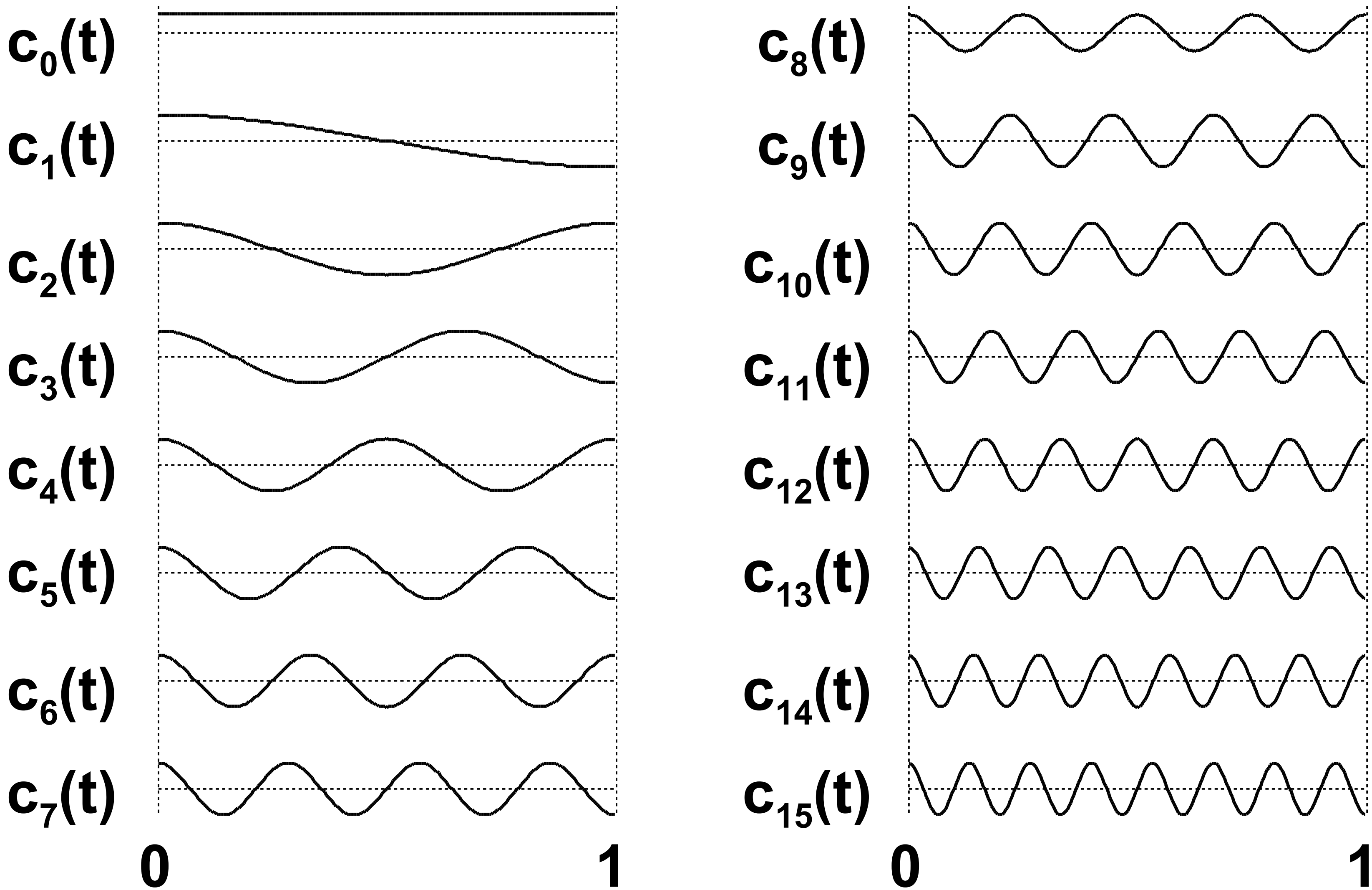
Unitary transform:

$$G(k) = C_k \sum_{n=0}^{N-1} g(n) \cdot c_k(n)$$

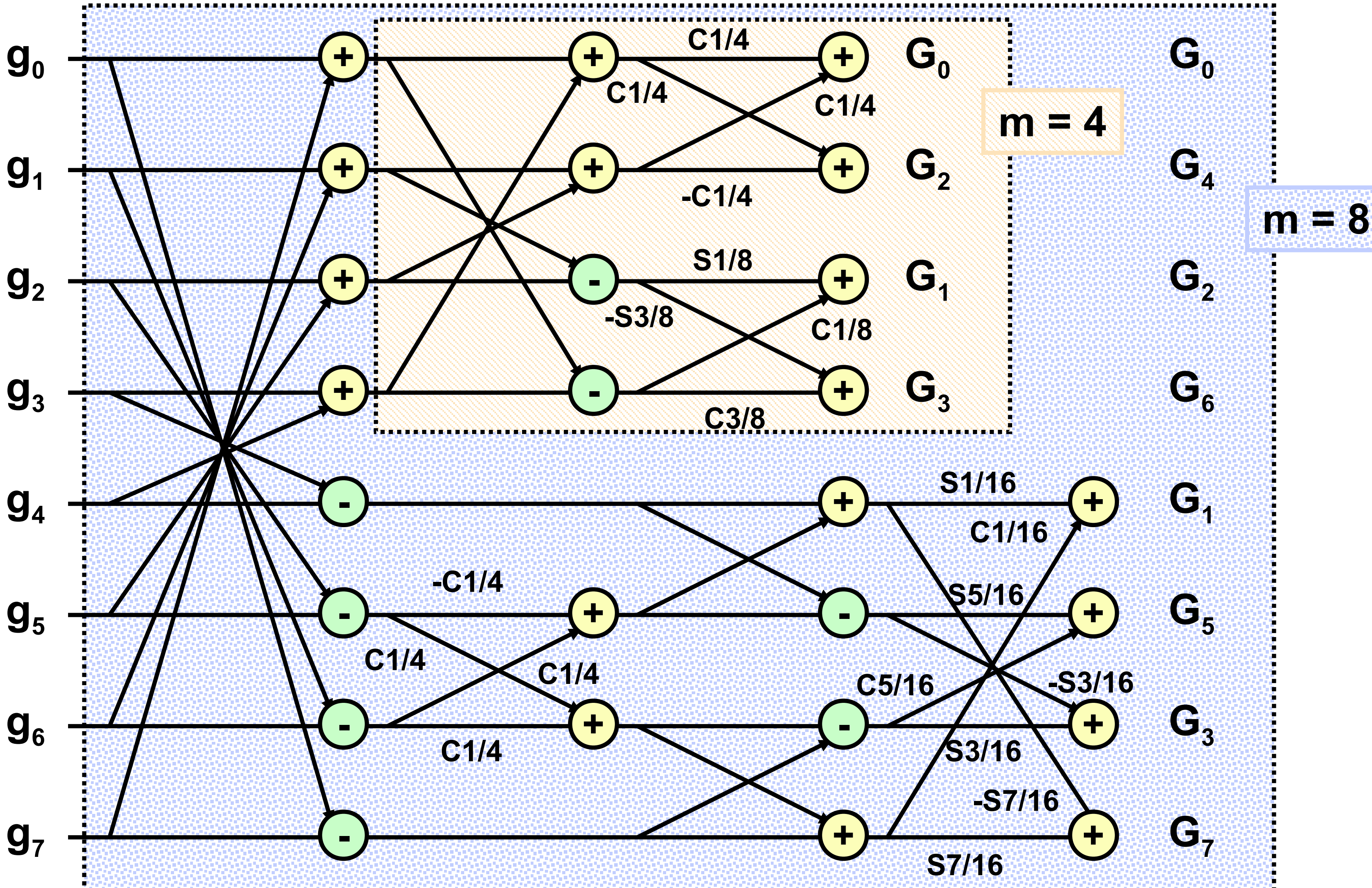
$$g(n) = \sum_{k=0}^{N-1} C_k \cdot G(k) \cdot c_k(n)$$

$$C_k = \begin{cases} \sqrt{\frac{1}{N}} & \text{for } k = 0 \\ \sqrt{\frac{2}{N}} & \text{else} \end{cases}$$

Example of 1D cosine bases



Fast DCT scheme



Discrete Hartley transform

Real-number variant of discrete Fourier transform

Basis: $$\left\{ h_k(n) = \frac{1}{\sqrt{N}} \operatorname{cas} \frac{2\pi kn}{N}; \quad 0 \le k,n < N \right\}$$

Unitary transform:

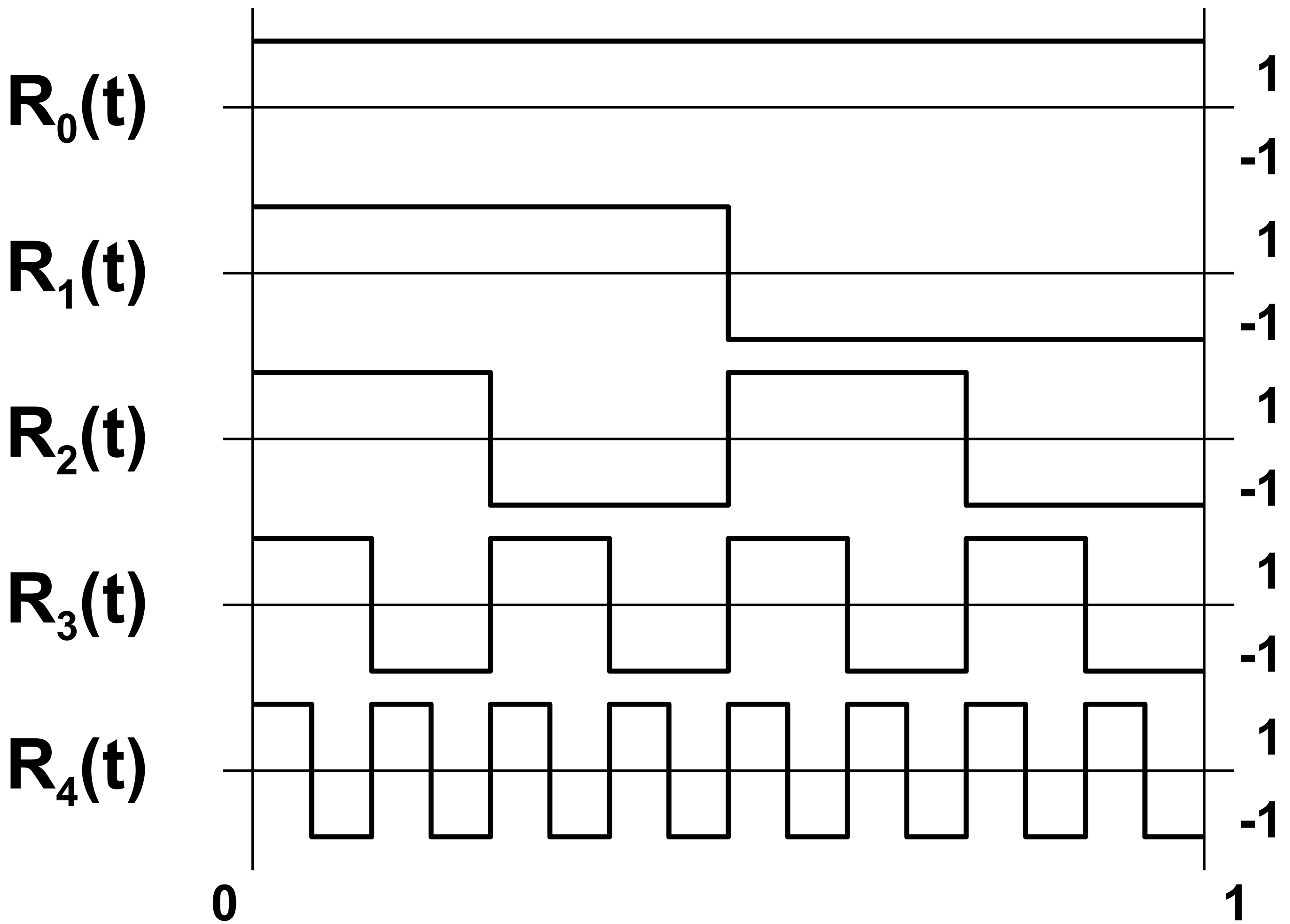
$$\operatorname{cas} x = \sin x + \cos x$$

$$G(k) = \sum_{n=0}^{N-1} g(n) \cdot h_k(n)$$

$$g(n) = \sum_{k=0}^{N-1} G(k) \cdot h_k(n)$$

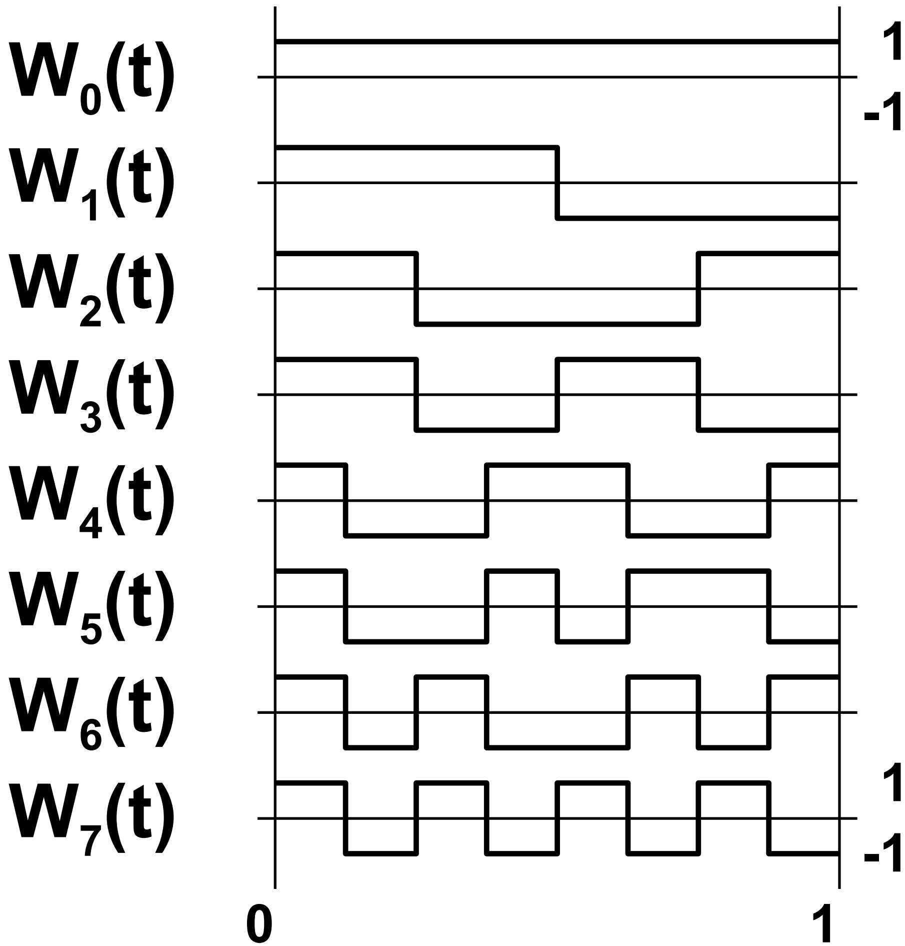
Rademacher functions (1922)

Orthogonal **incomplete** system of functions on **⟨0,1⟩**



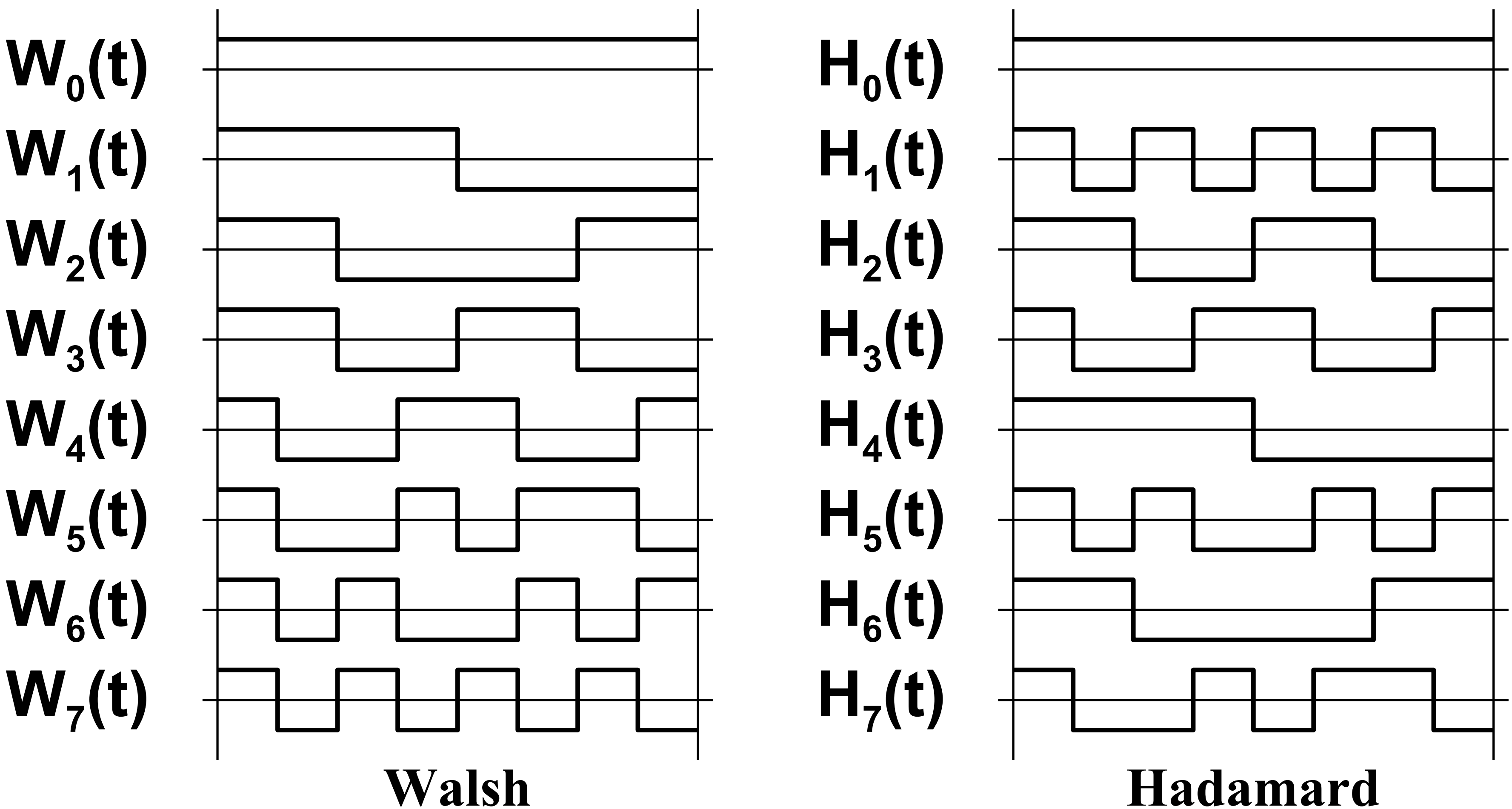
Walsh functions (1923)

Complete system made from Rademacher functions:



Hadamard's ordering

Ordering by frequency vs. recursive decomposition:



Hadamard transform

For image data with higher correlation coefficients

Basis: $\left\{ H_k(n) = \frac{1}{\sqrt{N}}(-1)^{b(k,n)};\quad 0 \le k,n < N \right\}$

Unitary transform:

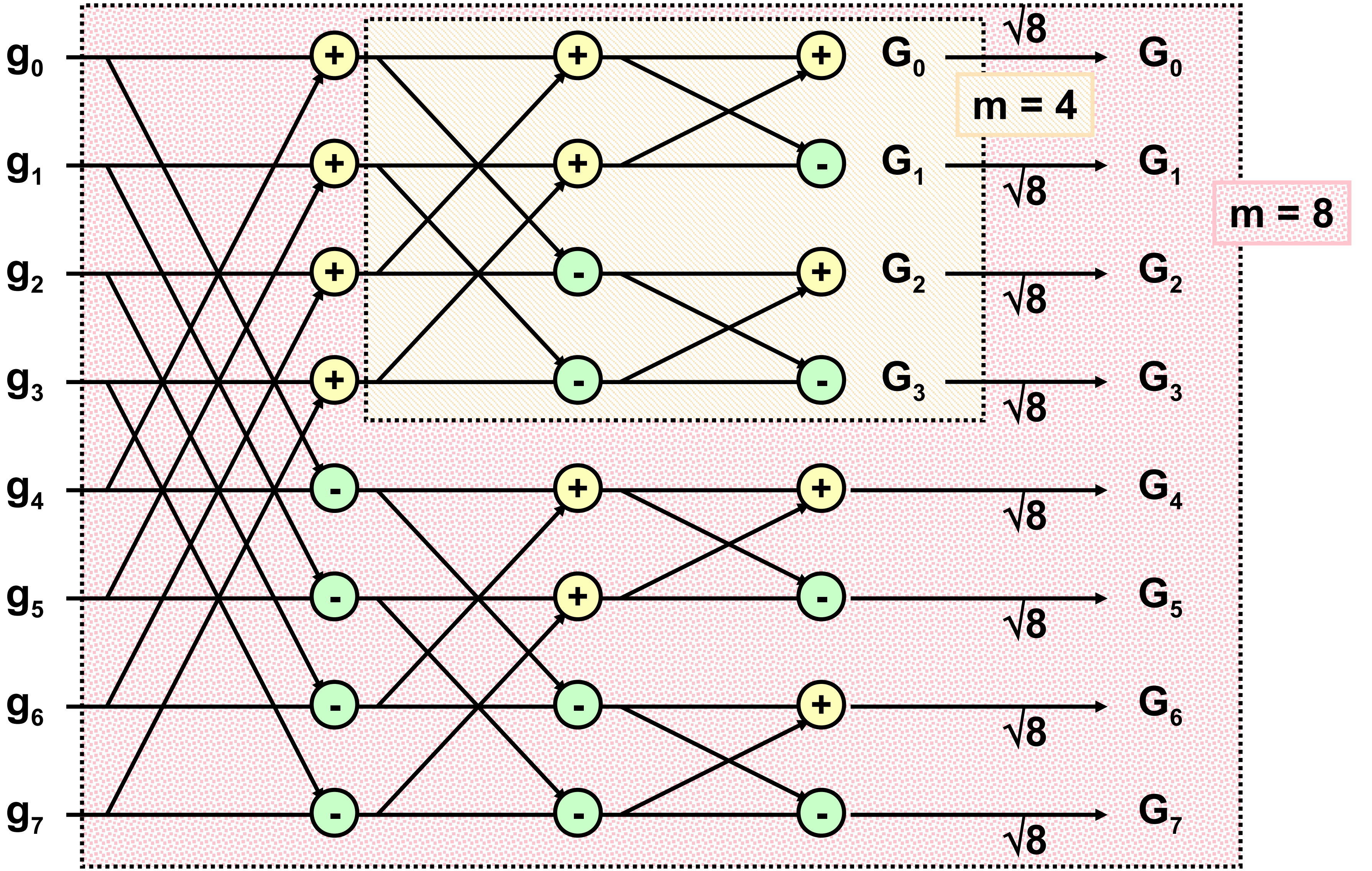
$$G(k) = \sum_{n=0}^{N-1} g(n) \cdot H_k(n)$$

$$g(n) = \sum_{k=0}^{N-1} G(k) \cdot H_k(n)$$

$$b(k,n) = \sum_{i=0}^{N-1} k_i\, n_i$$

binary notations

Fast Hadamard transform



Hadamard in matrix form

Recursive definition of Hadamard transform in matrix form (only for $\mathbf{N=2^k}$):

$$\mathbf{H_2} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$\mathbf{H_{2N}} = \frac{1}{\sqrt{2}}\left[\begin{array}{c|c} \mathbf{H_N} & \mathbf{H_N} \\ \hline \mathbf{H_N} & -\mathbf{H_N} \end{array}\right]$$

Matrix transform **H**:

$$\mathbf{G^T = Hg^T} \qquad \mathbf{g = GH}$$

Haar orthonormal basis

A. Haar (1909):

For every $\mathbf{k}$ $\quad 0 \le \mathbf{k} < N = 2^n$ $\quad$ two unique numbers $\mathbf{p}, \mathbf{q}$ exist, such as:

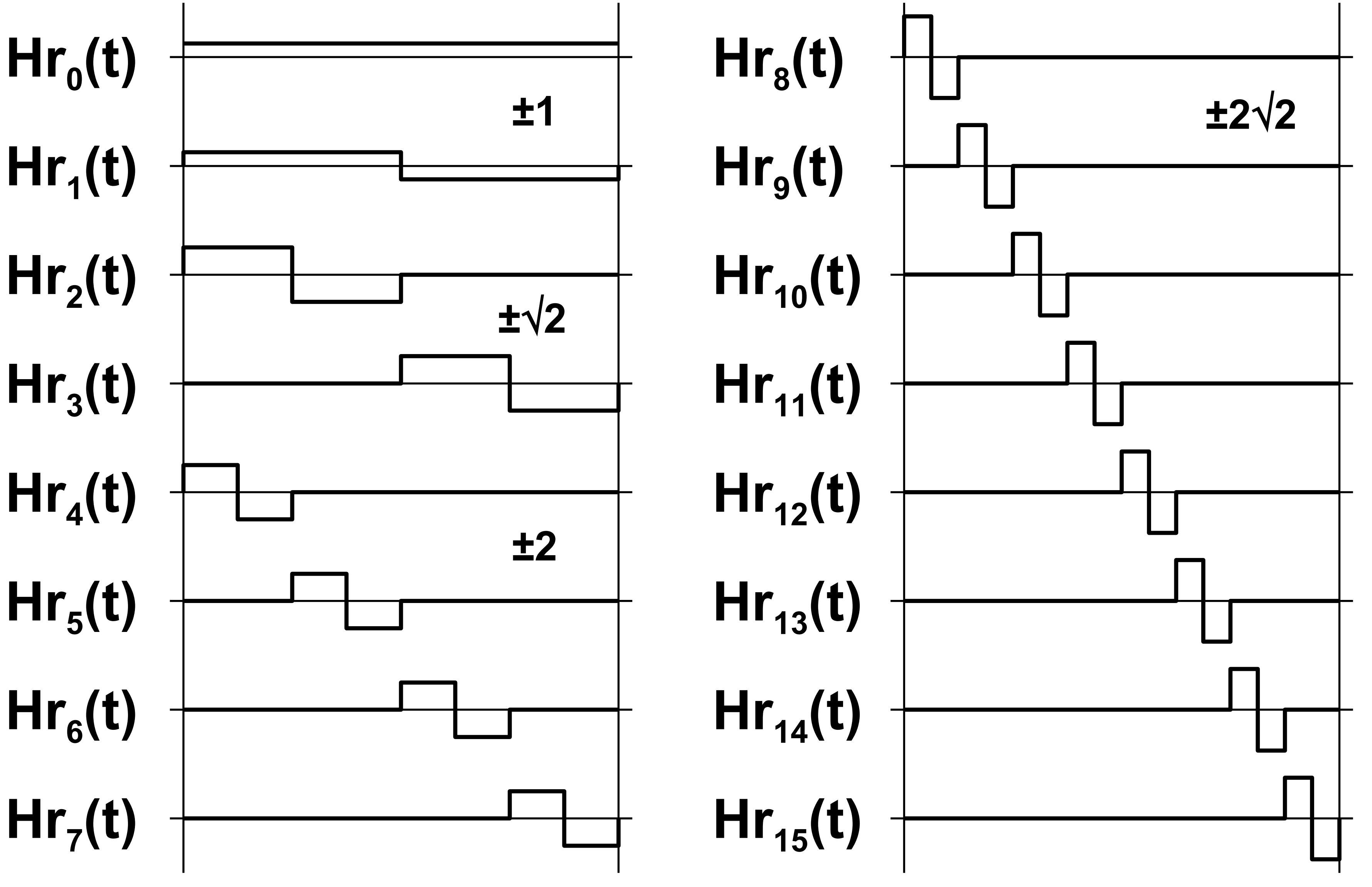
$$k = 2^p + q - 1$$

$$0 \le p < n \qquad 1 \le q \le 2^p$$

Basis on $\langle \mathbf{0,1} \rangle$:

$$Hr_0(x) = \frac{1}{\sqrt{N}} \qquad Hr_k(x) = \frac{1}{\sqrt{N}} \begin{cases} 2^{p/2} & \frac{q-1}{2^p} \le x < \frac{q-1/2}{2^p} \\ -2^{p/2} & \frac{q-1/2}{2^p} \le x < \frac{q}{2^p} \\ 0 & \text{else} \end{cases}$$

Haar basis for N=16



Haar transform

- represents well **local image changes**
  – most of basis members have very limited support
- the simplest **wavelet**
  – hierarchical recursive definition, all basis members can be obtained from one mother function by dilatation and translation only
- fast transform
  – $\mathbf{O(\log_2 N)}$: additive operations and multiplications $\mathbf{2^{p/2}}$

Slant transform

- Slant basis contains **partially linear** functions

Recursive definition of Slant transform matrices:

$$\mathbf{S}_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$\mathbf{S}_4 = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ \frac{3}{\sqrt{5}} & \frac{1}{\sqrt{5}} & \frac{-1}{\sqrt{5}} & \frac{-3}{\sqrt{5}} \\ 1 & -1 & -1 & 1 \\ \frac{1}{\sqrt{5}} & \frac{-3}{\sqrt{5}} & \frac{3}{\sqrt{5}} & \frac{-1}{\sqrt{5}} \end{bmatrix}$$

General recursive definition

$$S_{2N} = \frac{1}{\sqrt{2}} \left[\begin{array}{cc|c|cc|c} 1 & 0 & \multirow{2}{*}{0} & 1 & 0 & \multirow{2}{*}{0} \\ a_{2N} & b_{2N} & & -a_{2N} & b_{2N} & \\ \hline \multicolumn{2}{c|}{0} & I_{N-2} & \multicolumn{2}{c|}{0} & I_{N-2} \\ \hline 0 & 1 & \multirow{2}{*}{0} & 0 & -1 & \multirow{2}{*}{0} \\ -b_{2N} & a_{2N} & & b_{2N} & a_{2N} & \\ \hline \multicolumn{2}{c|}{0} & I_{N-2} & \multicolumn{2}{c|}{0} & -I_{N-2} \end{array} \right] \left[\begin{array}{c|c} S_N & 0 \\ \hline 0 & S_N \end{array} \right]$$

$$a_{2N} = \sqrt{\frac{3N^2}{4N^2 - 1}} \qquad b_{2N} = \sqrt{\frac{N^2 - 1}{4N^2 - 1}}$$

The End

More info:

- **A. Jain**: ***Fundamentals of Digital Image Processing***, Prentice-Hall, 1989, 132-188
- **W. Pratt**: ***Digital Image Processing***, 2nd edition, J. Wiley, New York, 1991, 193-216
- **S. Haykin**: ***An Introduction to Analog and Digital Communications***, J. Wiley, New York, 1989