Wavelets for Computer Graphics

AK Computergrafik
WS 2005/06

Markus Grabner

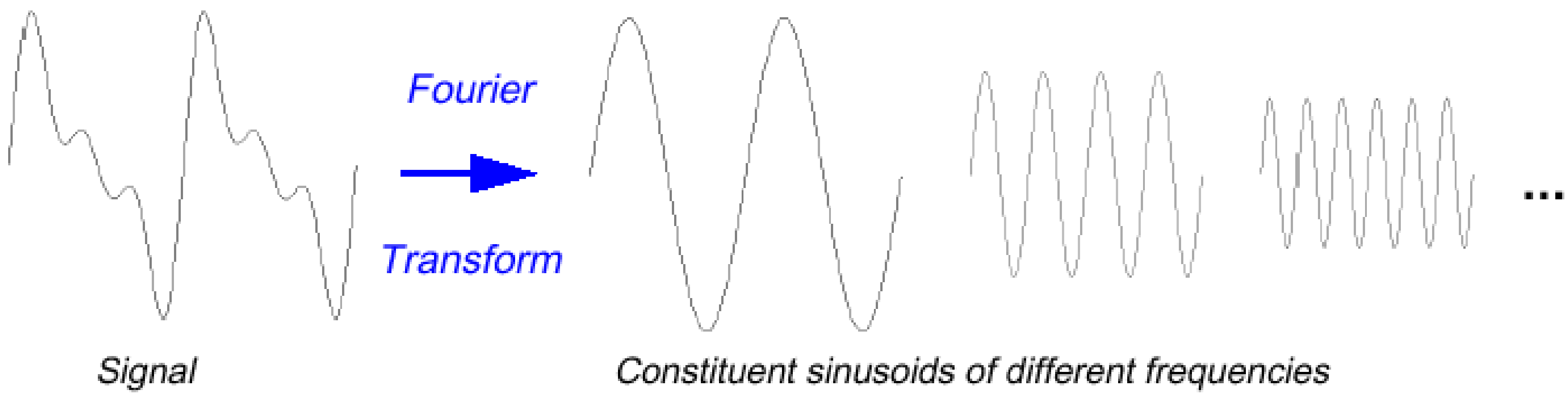
Content

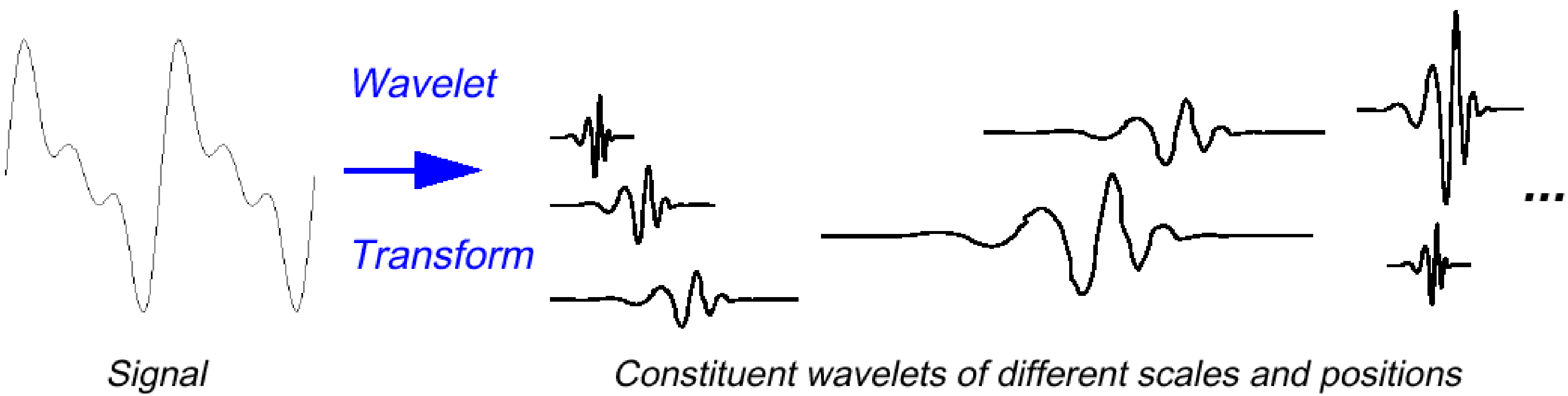
- Introduction
- Simple example (Haar wavelet basis)
- Mathematical background
- Image operations
- Other useful properties
- MATLAB examples

Introduction

- Large data
  - Analyzation
  - Visualization
  - Manipulation
- Single value not meaningful
- Domains
  - Time, space, ...
  - Frequency (Fourier)
  - Mix?

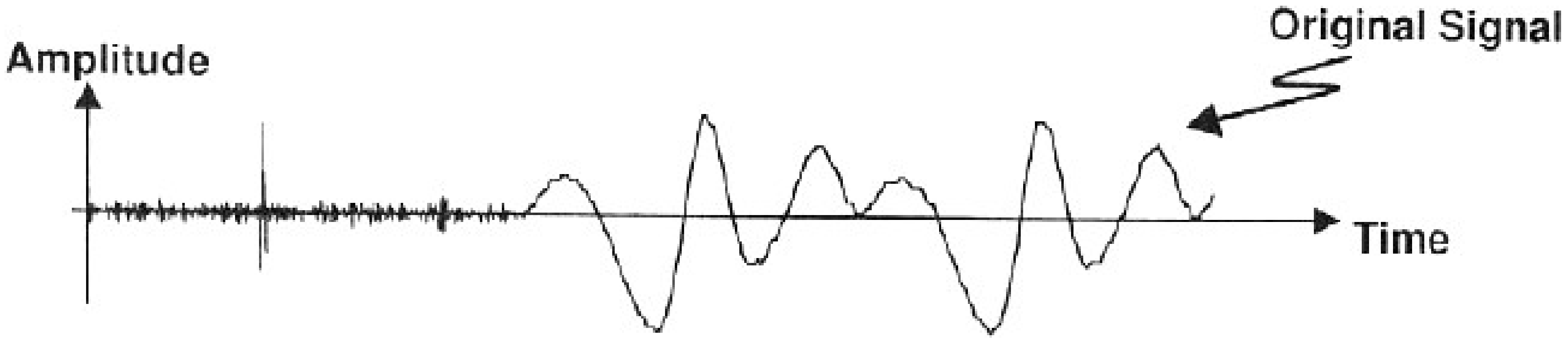
Fourier vs. wavelets (1)





Fourier vs. wavelets (2)

- Fourier: localized in frequency only
  (each coefficient captures entire signal)
- Wavelets: localized in time and frequency



History

- First ideas by Weierstrass (1873)
- Not “invented”, but used in different fields:
  - Seismology (term coined by Ricker, 1940)
  - Physics
  - Signal and image processing
- Multiresolution analysis (Mallat, 1989)

Wavelet properties

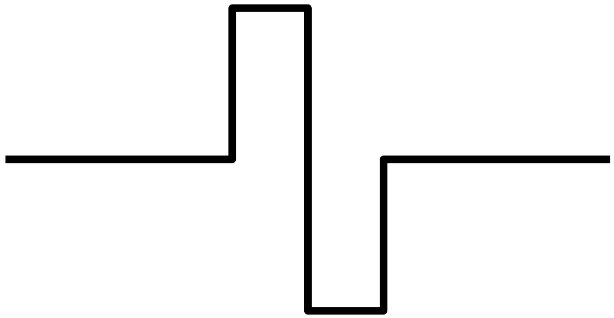
- Hierarchical representations
- Linear time complexity (conversion to/from)
- Sparsity (compression, efficiency)
- Adaptibility (wide variety of functions/domains)

MR group @ Caltech

- Prof. Peter Schröder
- Linear elasticity and wavelets

Haar wavelet basis

- Alfred Haar, 1909
- Simplest wavelet basis
- Example



Example: sequence of 4 values

[9 3 2 6]

Build averages

Averages

[9   3   2   6]

[6   4]

Build detail coefficients

	Averages	Detail Coefficients
	[9 3 2 6]	
	[6 4]	[3 -2]

Repeat procedure

Resolution	Averages	Detail Coefficients
4	[9 3 2 6]	
2	[6 4]	[3 -2]
1	[5]	[1]

Wavelet transform

Resolution	Averages	Detail Coefficients
4	[9 3 2 6]	
2	[6 4]	[3 -2]
1	[5]	[1]

wavelet transform [5 1 3 -2]

Algorithms

- Decomposition
  - Compute averages
  - Compute detail coefficients
  - Continue at lower level
- Reconstruction
  - Apply detail coefficients
  - Continue at higher level

Mathematical background

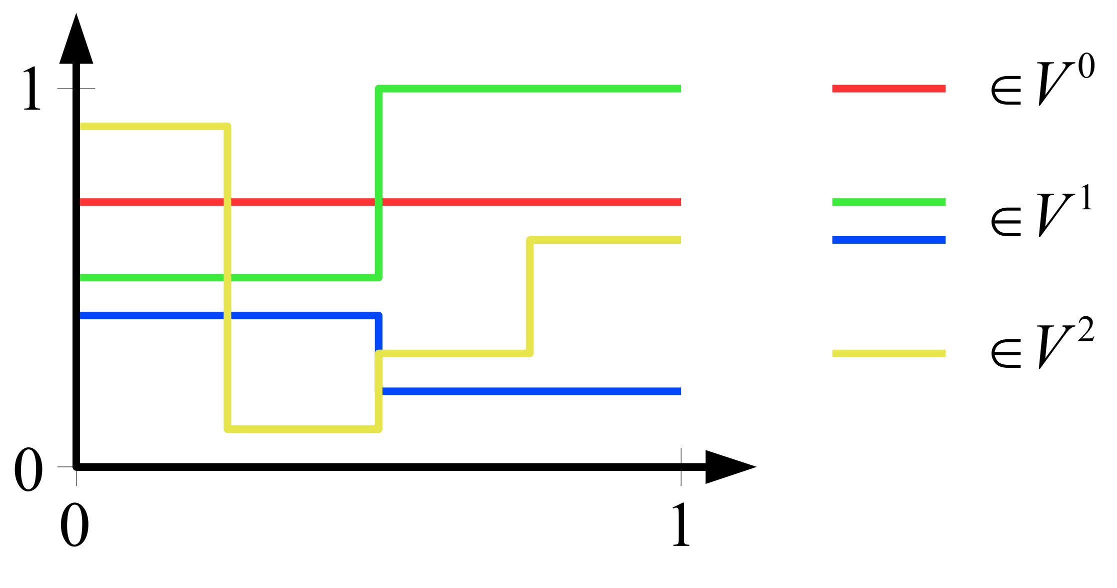
- Vector spaces
- Involved functions
- Orthogonality
- Normalization
- Compression

Vector spaces

- Collection of “things”
- Addition and scalar multiplication
- Can be
  - Arrays of scalar values
  - Functions
  - …
- Example
- Nested spaces

Vector space example

- One pixel image: function constant in [0,1)
- Operations well defined $\rightarrow$ vector space V^0
- Two pixels: constant in [0,0.5) and [0.5,1) $\rightarrow$ V^1



Nested vector spaces

- Previous example: each vector in V^j is also in V^{j+1}
- Formally:

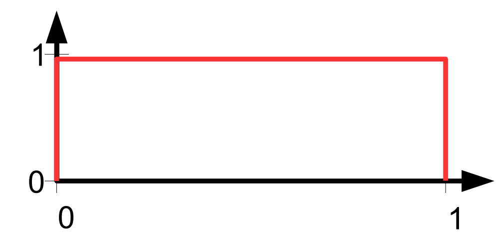
$$V^0 \subset V^1 \subset V^2 \subset \cdots$$

"Tools" for vector spaces

- Basis:

$$\phi_i^j(x) := \phi(2^j x - i), \quad i = 0, \ldots, 2^j - 1$$

$$\phi(x) := \begin{cases} 1 & \text{for } 0 \le x < 1 \\ 0 & \text{otherwise} \end{cases}$$



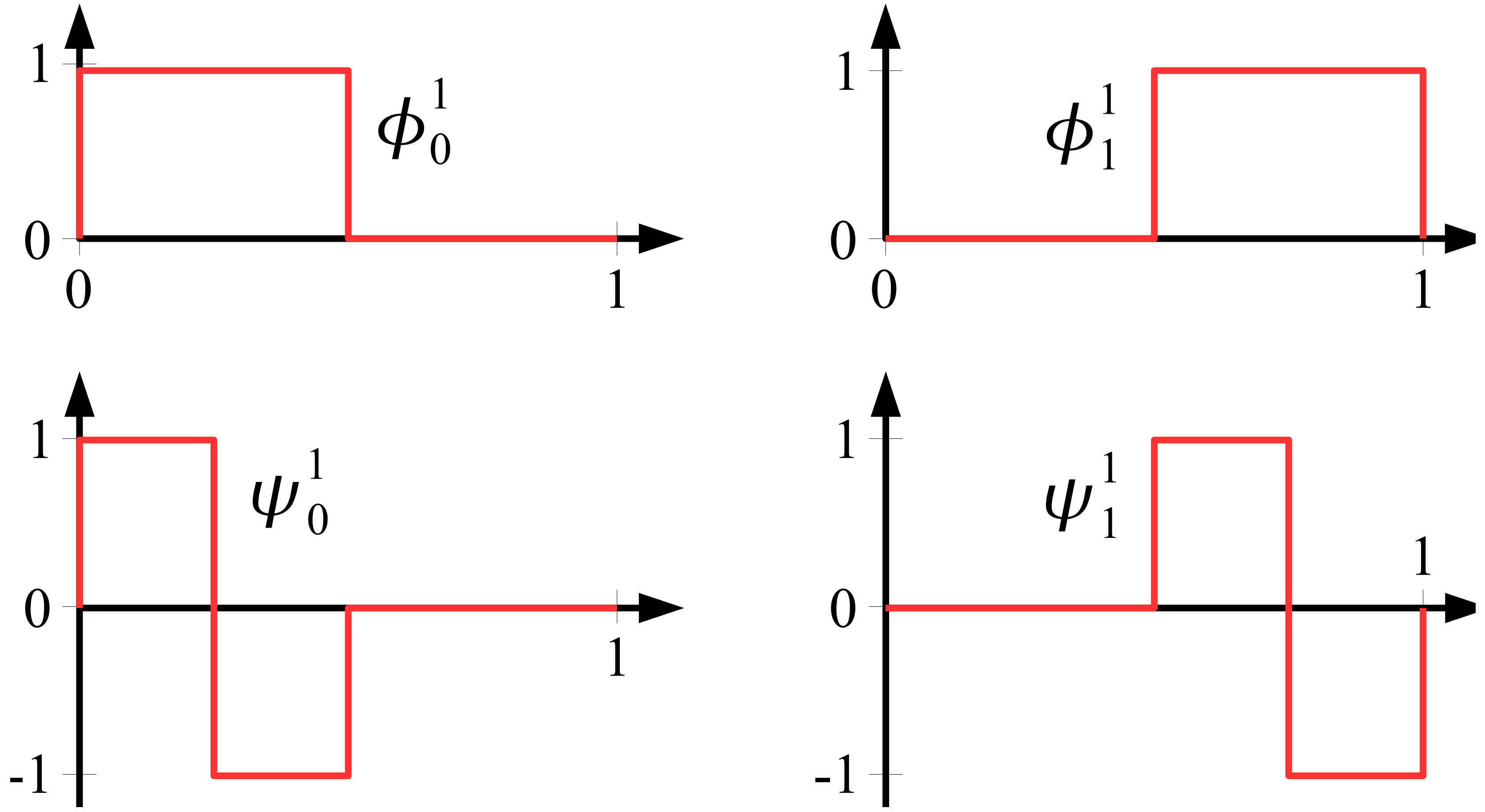
- Support: region where function is nonzero
- Compact support: supported over bounded interval
- Inner product: we choose

$$\langle f | g \rangle := \int_0^1 f(x)\, g(x)\, dx$$

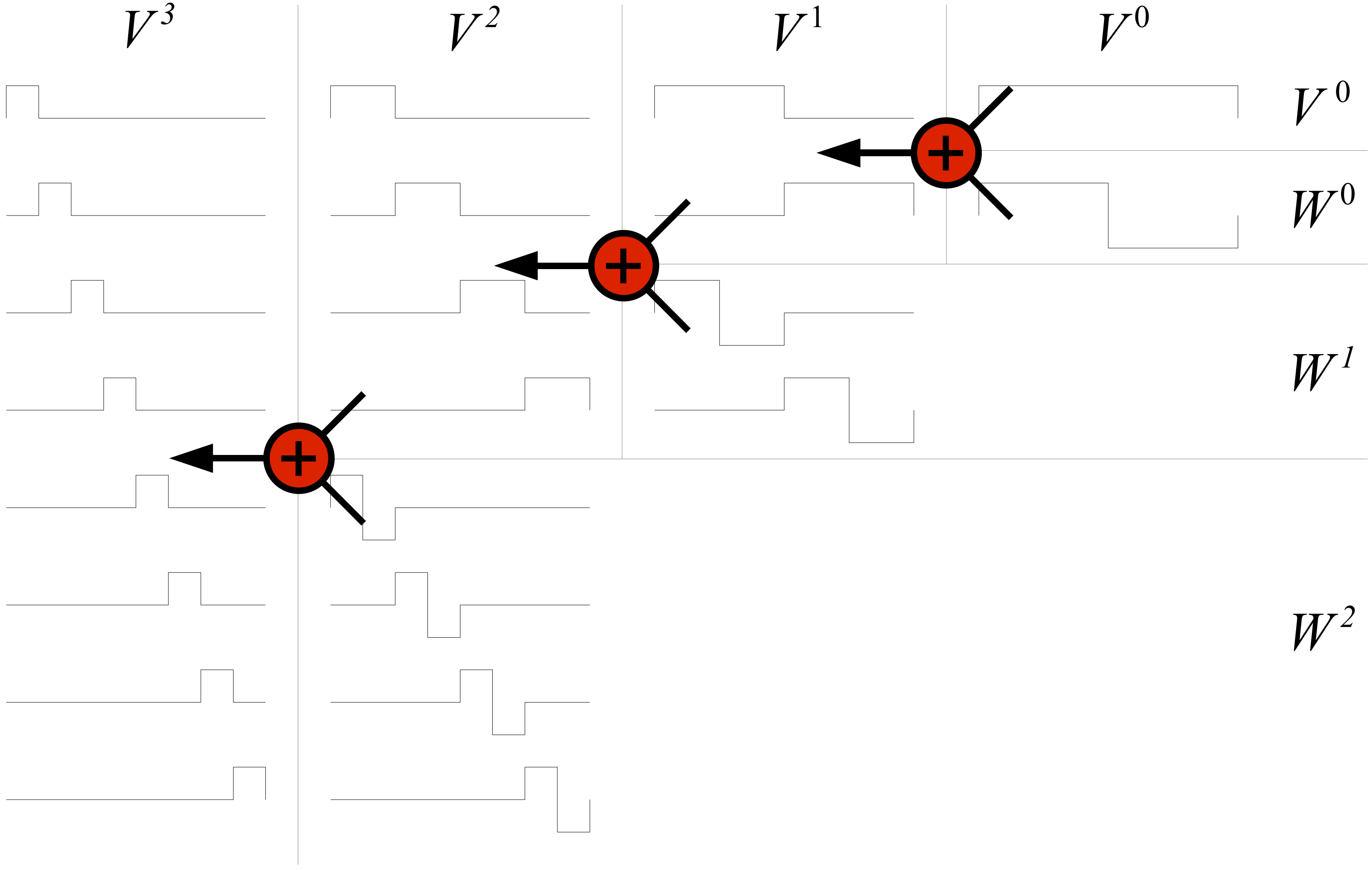
Wavelet definition

- Vectors u, v orthogonal if $\langle u | v \rangle = 0$
- W^j orthogonal complement of V^j in V^{j+1}
- Wavelets: collection of linearly independent functions $\psi_i^j(x)$ spanning W^j
- Properties:
  - $\psi_i^j(x)$ of W^j and $\phi_i^j(x)$ of V^j form basis in V^{j+1}
  - $\psi_i^j(x)$ of W^j and $\phi_i^j(x)$ of V^j are orthogonal
- Informal: wavelets capture details beyond V^j

Scaling functions and wavelets



Relations between vector spaces



Wavelet properties

- Useful properties (not shared by all wavelet bases)
- Orthogonality
  - All basis functions orthogonal to each other
- Normalization
  - Basis function $u(x)$ is normalized if $\langle u | u \rangle = 1$
  - Normalized Haar basis:

$$\phi_i^j(x) = \sqrt{2^j}\,\phi(2^j x - i)$$
$$\psi_i^j(x) = \sqrt{2^j}\,\psi(2^j x - i)$$

Compression basics

- JPEG2000: store twice the number of pictures in your digital camera at the same quality as JPEG
- Express initial data by smaller set of data
- Lossless or lossy
- User-specified error tolerance ε
- In the wavelet context:

$$f(x)=\sum_{i=1}^{m} c_i u_i(x) \;\rightarrow\; \hat{f}(x)=\sum_{i=1}^{\hat{m}} \hat{c}_i \hat{u}_i(x)$$

$$\hat{m}<m, \quad \left\| f(x)-\hat{f}(x) \right\| \le \varepsilon \quad \text{for some norm}$$

Wavelet compression

- We choose a fixed basis, i.e., $\hat{u}_i = u_i, \quad i = 1, \ldots, \hat{m}$
- Coefficient encoding not discussed here
- Which coefficients to select?
- Define permutation $\pi(i)$
- Use some of the coefficients:

$$\hat{f}(x) = \sum_{i=1}^{\hat{m}} \hat{c}_{\pi(i)} \hat{u}_{\pi(i)}(x)$$

Approximation error

- Square of L^2 error of approximation:

$$\left\| f-\hat{f} \right\|_2^2 = \left\langle f-\hat{f} \middle| f-\hat{f} \right\rangle$$

$$= \left\langle \sum_{i=\hat{m}+1}^{m} c_{\pi(i)} u_{\pi(i)} \middle| \sum_{j=\hat{m}+1}^{m} c_{\pi(j)} u_{\pi(j)} \right\rangle$$

$$= \sum_{i=\hat{m}+1}^{m} \sum_{j=\hat{m}+1}^{m} c_{\pi(i)} c_{\pi(j)} \left\langle u_{\pi(i)} \middle| u_{\pi(j)} \right\rangle$$

orthonormal basis:

$$\left\langle u_i \middle| u_j \right\rangle = \delta_{ij} \longrightarrow = \sum_{i=\hat{m}+1}^{m} \left(c_{\pi(i)} \right)^2$$

- Omit smallest coefficients first

Applications

- Image compression
- Image editing
- Image querying

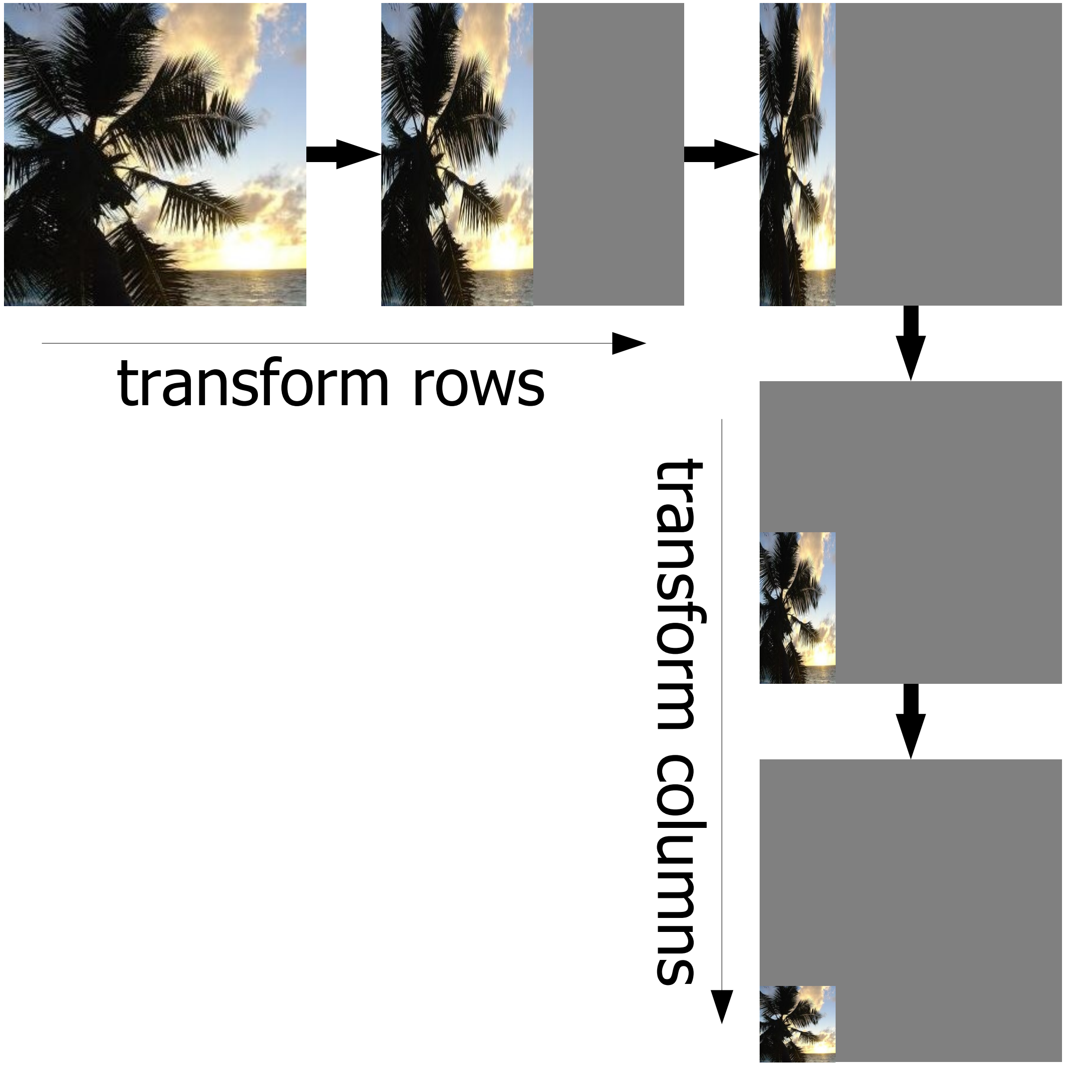
Image compression

- Generalization of Haar wavelets to two dimensions
- Similar framework for entirely different applications
- Two common ways to transform image:
  - Standard decomposition
  - Non-standard decomposition

Standard decomposition (1)

- Apply 1D transform to each row
- Apply 1D transform to each (transformed) column
- Results are
  - Single overall average coefficient
  - Detail coefficients

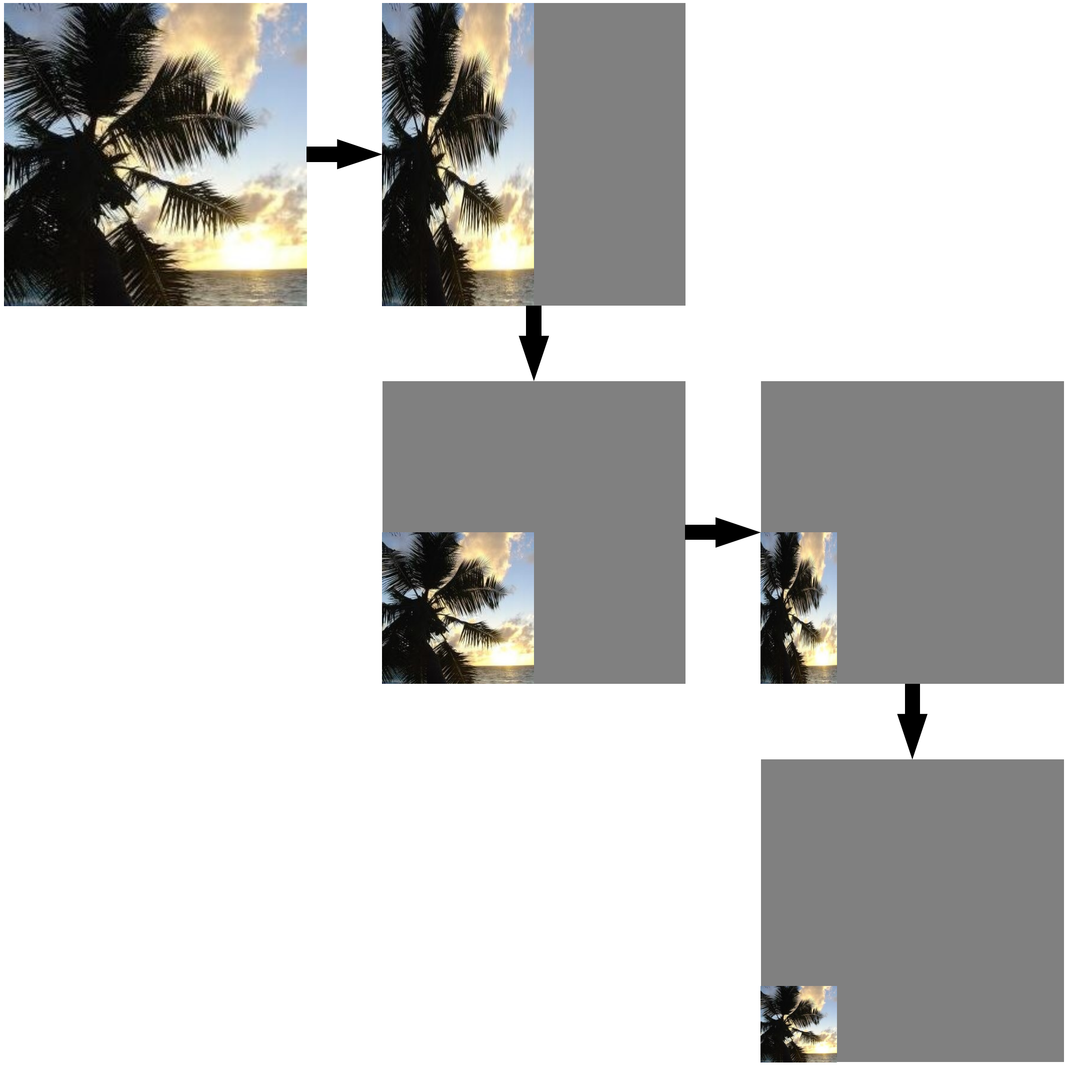
Standard decomposition (2)



Nonstandard decomposition (1)

- Apply averaging/differencing step to each row
- Apply averaging/differencing step to each column
- Repeat procedure on “average” quadrant
- Results are
  - Average coefficients at each level
  - Detail coefficients at each level

Nonstandard decomposition (2)

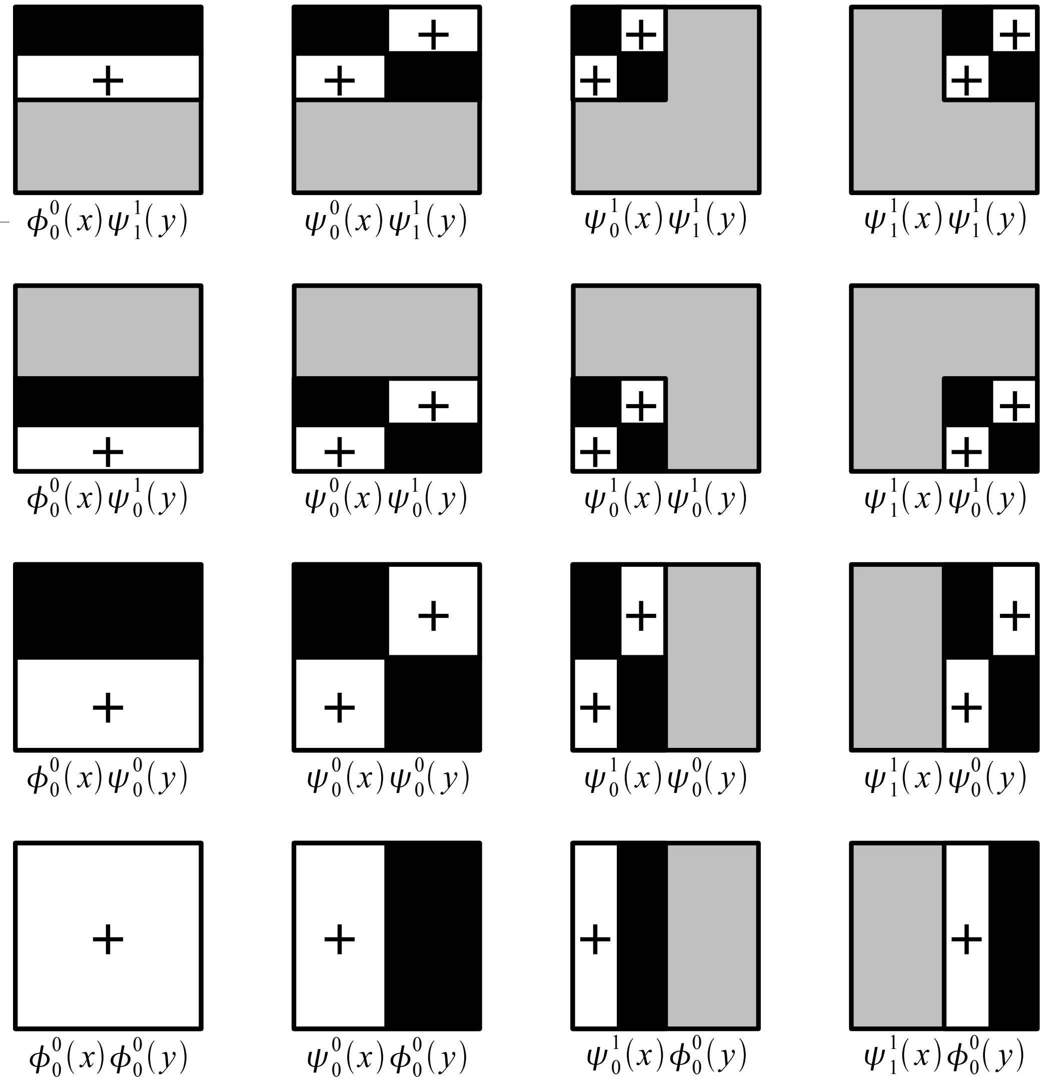


2D Haar basis functions

- Standard construction: tensor products
- Nonstandard construction:
  - One 2D scaling function
  - Three 2D wavelet functions
  - Combine at different levels

Standard construction

$$\phi_0^0(x)\psi_1^1(y)$$



Nonstandard construction (1)

- Scaling function:

$$\phi\phi(x,y)=\phi(x)\phi(y)$$

- Wavelet functions:

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

Nonstandard construction (2)

- Single coarse scaling function:

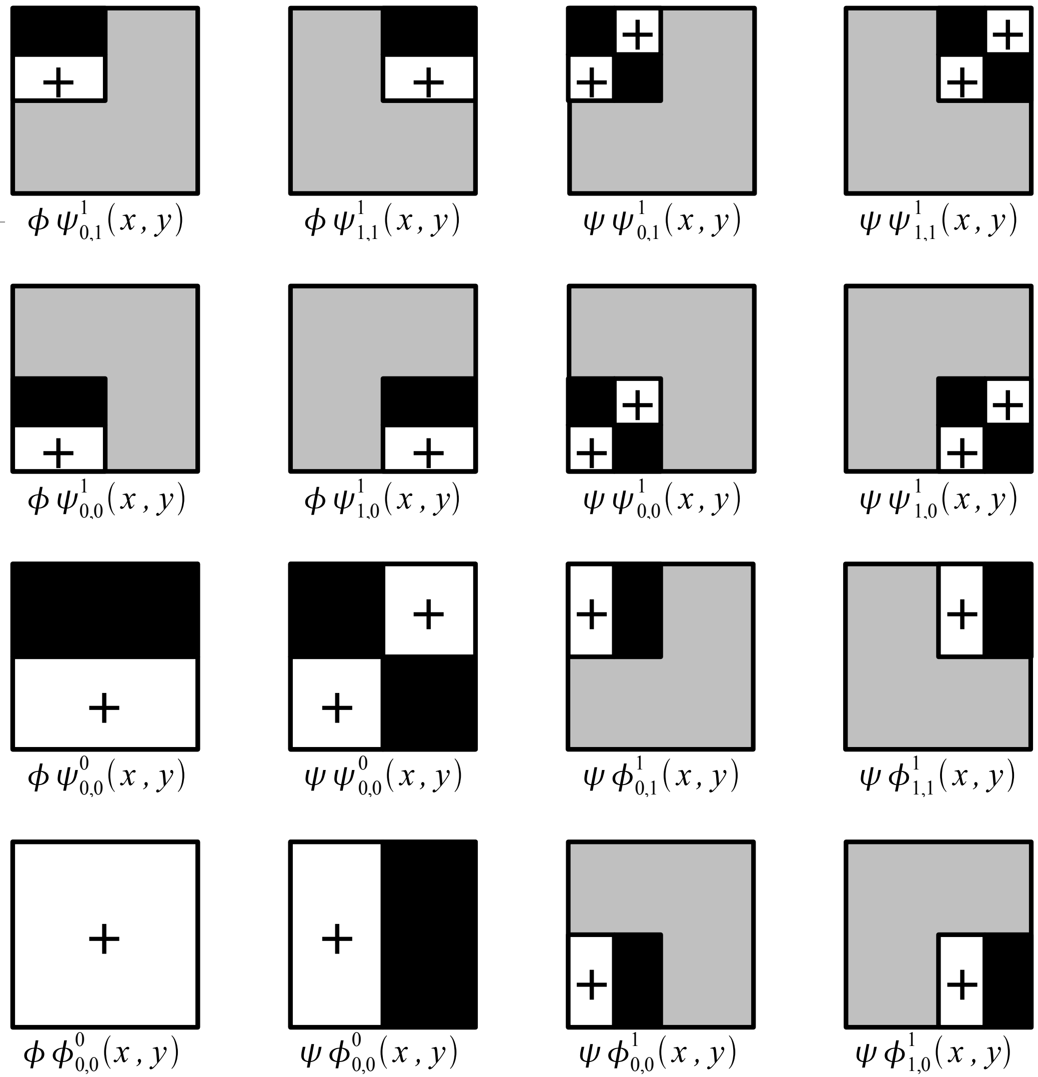
$$\phi\phi^{0}_{0,0}(x,y)=\phi\phi(x,y)$$

- Translated and scaled wavelet functions:

$$\phi\psi^{j}_{k,l}(x,y)=2^{j}\phi\psi(2^{j}x-k,\ 2^{j}y-l)$$
$$\psi\phi^{j}_{k,l}(x,y)=2^{j}\psi\phi(2^{j}x-k,\ 2^{j}y-l)$$
$$\psi\psi^{j}_{k,l}(x,y)=2^{j}\psi\psi(2^{j}x-k,\ 2^{j}y-l)$$

Nonstandard construction (3)

$\phi\,\psi^{1}_{0,1}(x,y)$



Standard vs. nonstandard

- Number of assignments for $n \times n$ image:
  - Standard: $4(n^2-n)$
  - Nonstandard: $8/3(n^2-1)$
- Number of nonzero coefficients for $O(n)$ inputs:
  - Standard: $O(n \log n)$
  - Nonstandard: $O(n)$
- Solution of linear equations:
  - Standard: explicit transformation possible $\rightarrow$ fits well with existing software
  - Nonstandard: implicit transformation $\rightarrow$ doesn't fit well

More applications

- Image editing
  - Edit image at desired scale
  - Quadtree representing wavelet hierarchy
- Image querying
  - Query by content (e.g., rough sketch)
  - Similarity metric
  - Compare m largest wavelet coefficients

Other useful properties

- Solution of linear systems
- Vanishing moments

Solution of linear systems (1)

- Consider set of linear equations $\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$
- If $\mathbf{A}$ is dense: slow direct/iterative solution
- Much faster for sparse matrices
- Q: How can we "make $\mathbf{A}$ sparse"?
- A: By applying the wavelet transform
- All operations linear:
  - Matrix Ψ describes decomposition ($\Psi \cdot \mathbf{v} = \mathbf{v}'$)
  - Matrix Ψ^{-1} describes reconstruction ($\Psi^{-1} \cdot \mathbf{v}' = \mathbf{v}''$)

Solution of linear systems (2)

- We can formally write:

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$$

$$\underbrace{\Psi\,\mathbf{A}\,\Psi^{-1}}_{\mathbf{A}'\ (\text{sparse})} \cdot \Psi\,\mathbf{x} = \Psi\,\mathbf{b}$$

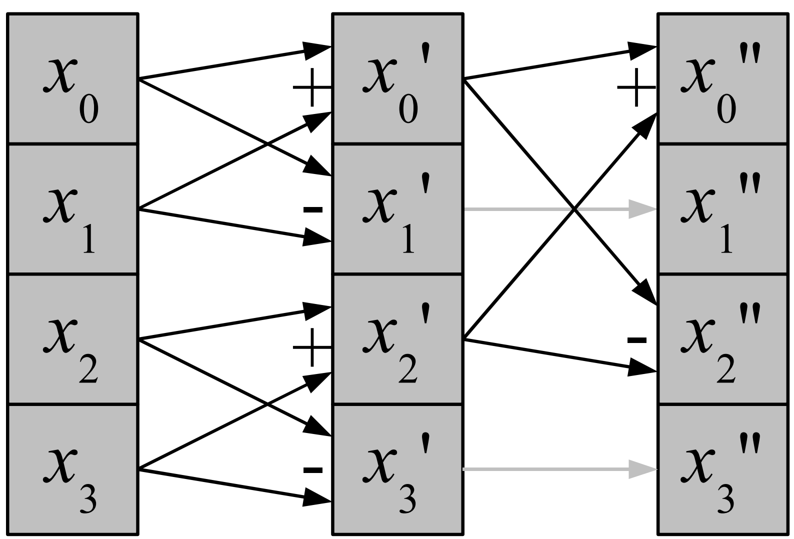
- Solve for $\mathbf{x}' = \Psi \mathbf{x}$ and reconstruct $\mathbf{x} = \Psi^{-1}\mathbf{x}'$
- Left multiplication transforms columns
- Right multiplication transforms rows
- No correspondence between $\mathbf{A}$ and image matrix

Wavelet matrix

- Compact representation of averaging/differencing
- Concatenate all steps: $\psi = \psi_2 \cdot \psi_1$
- Optional rearrangement
- Example (not normalized):

$$\mathbf{x}' = \underbrace{\begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{pmatrix}}_{\Psi_1} \mathbf{x}$$

$$\mathbf{x}'' = \underbrace{\begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{\Psi_2} \mathbf{x}'$$



Vanishing moments

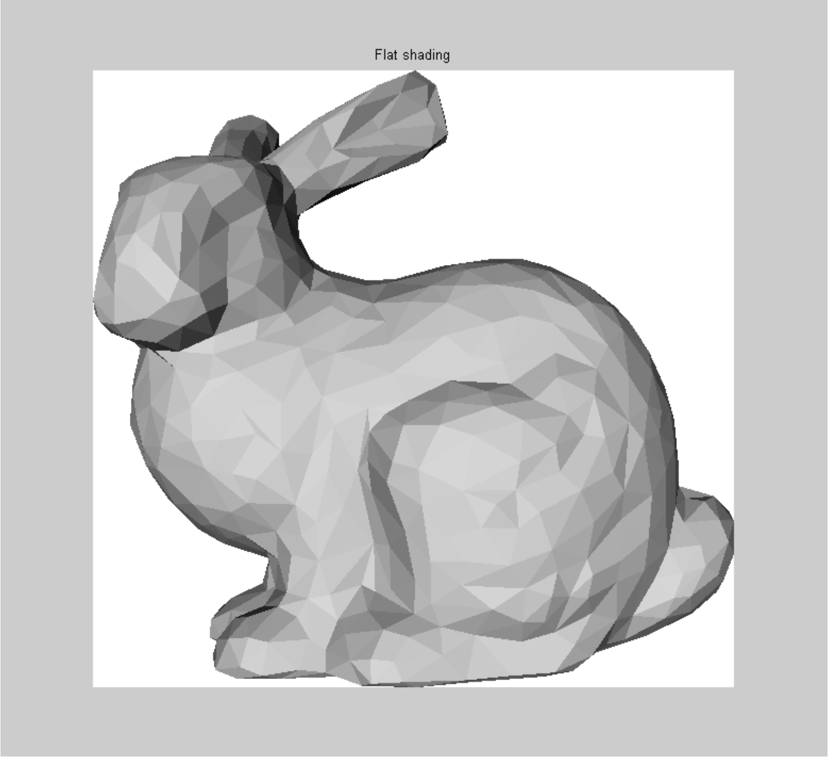
- A wavelet $\psi(x)$ has n vanishing moments if

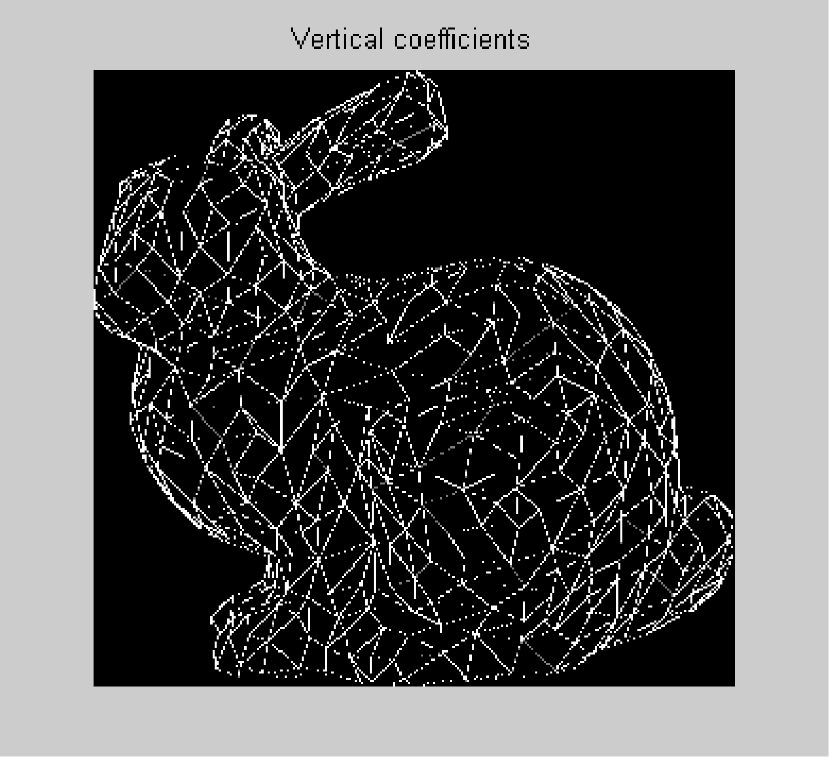
$$\int \psi(x) x^k \, dx = 0$$

  is true for all $k=0,\ldots,n-1$, but not for $k=n$
- Constant regions $\rightarrow$ zero coefficients (Haar basis)
- Higher order wavelets $\rightarrow$
  more vanishing moments $\rightarrow$
  zero coefficients for linear, quadratic, ... regions $\rightarrow$
  better compression of smooth images

Bunny (flat, Haar)

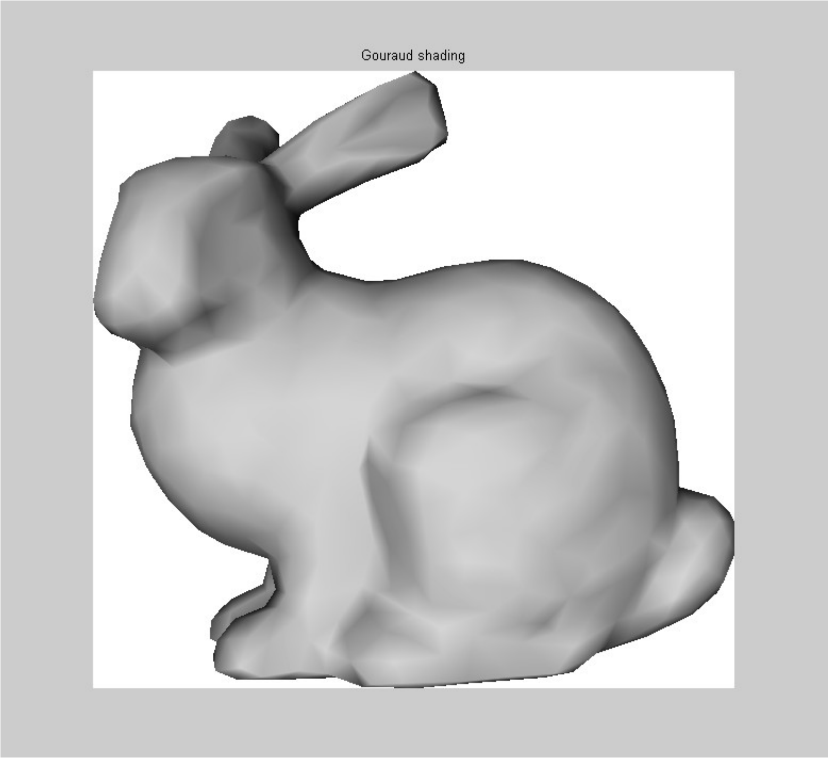
- Constant regions
- One vanishing moment





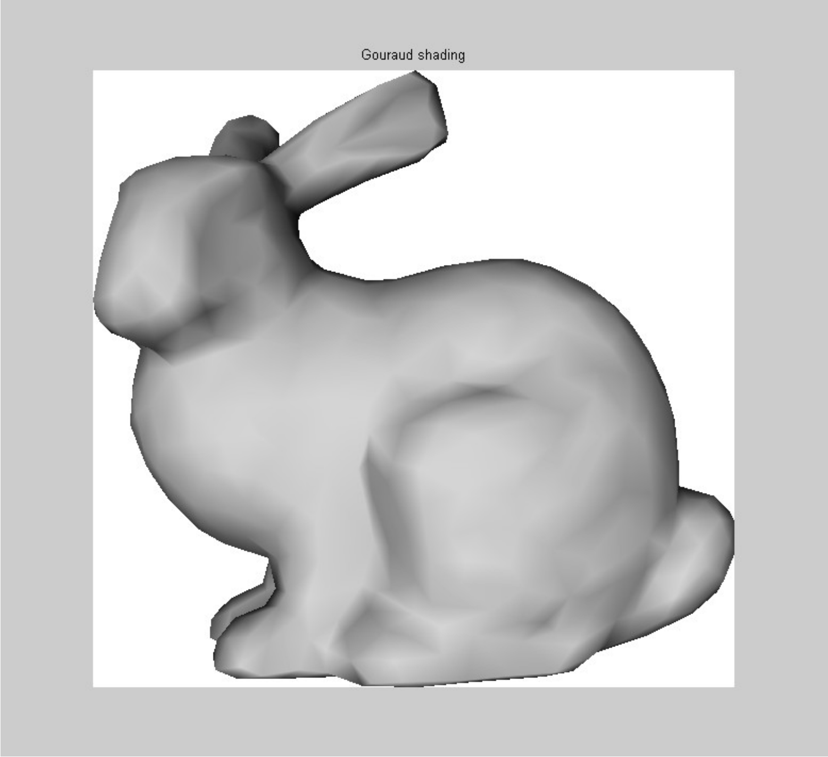
Bunny (Gouraud, Haar)

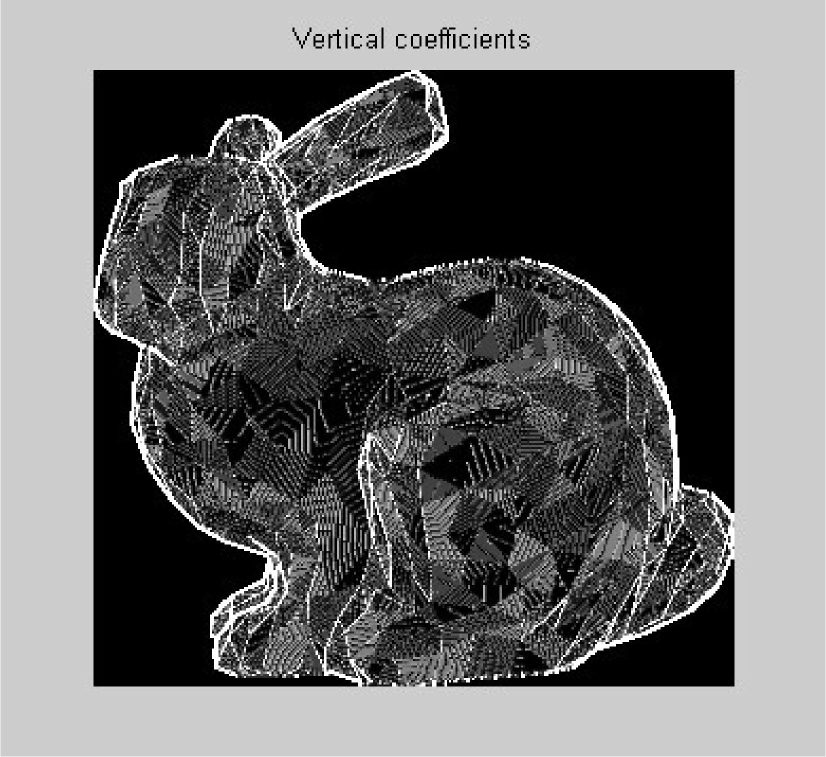
- Linear regions
- One vanishing moment



Bunny (Gouraud, Daubechies)

- Linear regions
- Two vanishing moments





Thank you for your attention!

Praktische Anwendungen

- Wavelets in Matlab
- Bildkompression
- Kantendetektion
- Lösen von Gleichungssystemen

Wavelets in Matlab

Wavelet	wfamily	wname
Haar	'haar'	'haar'
Daubechies	'db'	'db2', 'db3', ..., 'db45'
Coiflets	'coif'	'coif1', 'coif2', ..., 'coif5'
Symlets	'sym'	'sym2', 'sym3', ..., 'sym45'
Discrete Meyer	'dmey'	'dmey'
Biorthogonal	'bior'	'bior1.1', 'bior1.3', 'bior1.5', 'bior2.2', 'bior2.4', 'bior2.6', 'bior2.8', 'bior3.1', 'bior3.3', 'bior3.5', 'bior3.7', 'bior3.9', 'bior4.4', 'bior5.5', 'bior6.8'
Reverse Biorthogonal	'rbio'	'rbio1.1', 'rbio1.3', 'rbio1.5', 'rbio2.2', 'rbio2.4', 'rbio2.6', 'rbio2.8', 'rbio3.1', 'rbio3.3', 'rbio3.5', 'rbio3.7', 'rbio3.9', 'rbio4.4', 'rbio5.5', 'rbio6.8'

Fast Wavelet Transform

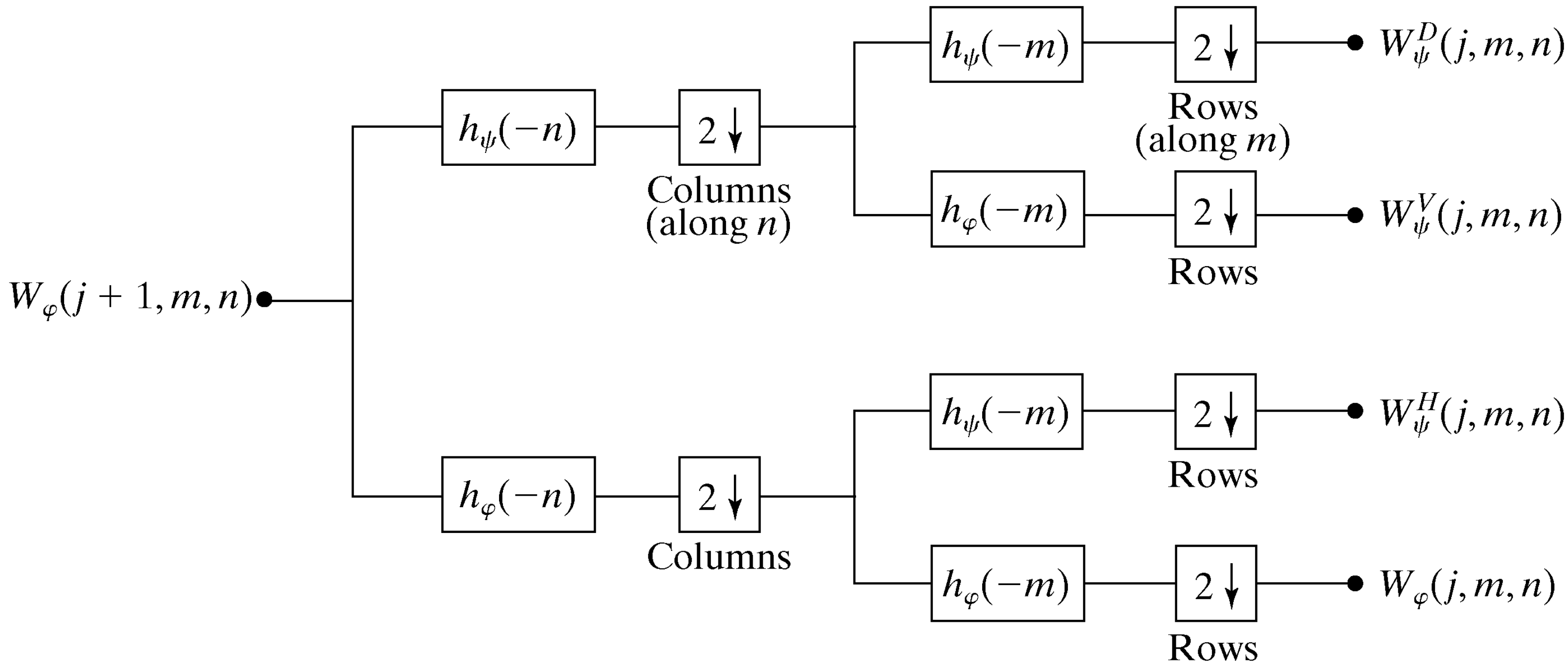
- Decomposition and reconstruction by convolution
- Convolution kernels defined in Wavelet filterbank
- 4 1D-filter masks needed:
  - 2 for decomposition
  - 2 for reconstruction

Haar wavelet filter masks

```
[ Lo_D Hi_D Lo_R Hi_R ] = wfilters('haar')
Lo_D =     0.7071    0.7071
Hi_D =    -0.7071    0.7071
Lo_R =     0.7071    0.7071
Hi_R =     0.7071   -0.7071
```

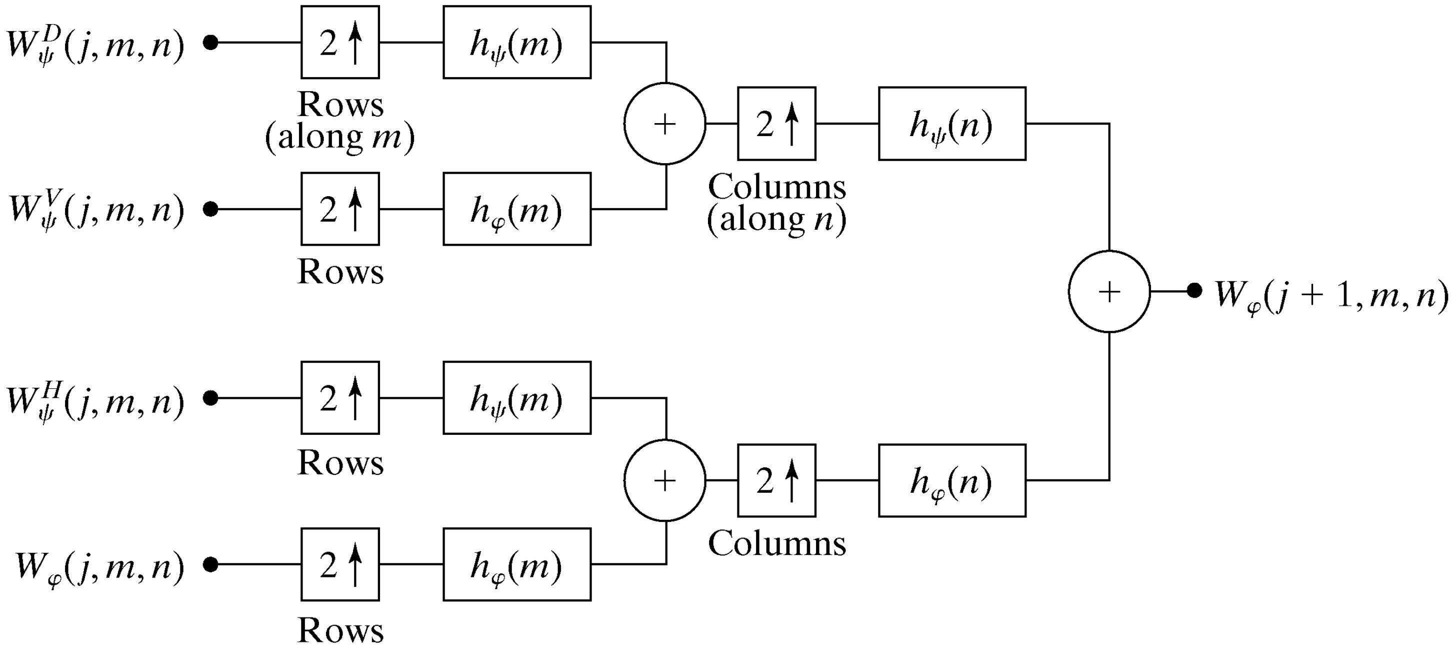
- Size of filters dependent on wavelet type
- Filter pair consists of lowpass filter and highpass filter

Decomposition



- Down arrows mean downsampling of the image by 2.
- First along columns, then along rows

Reconstruction



- Down arrows mean upsampling of the image by 2.
- First along rows, then along columns

Wavelets in Matlab

- Wavelet-Decomposition:
  - wavedec2
  - dwt2
- Wavelet-Reconstruction:
  - waverec2
  - idwt2
- Scaling&Wavelet Function
  - wavefun

dwt2

[CA,CH,CV,CD] = dwt2(X,'wname')

CA … approximation coefficients

CH … horizontal coefficients

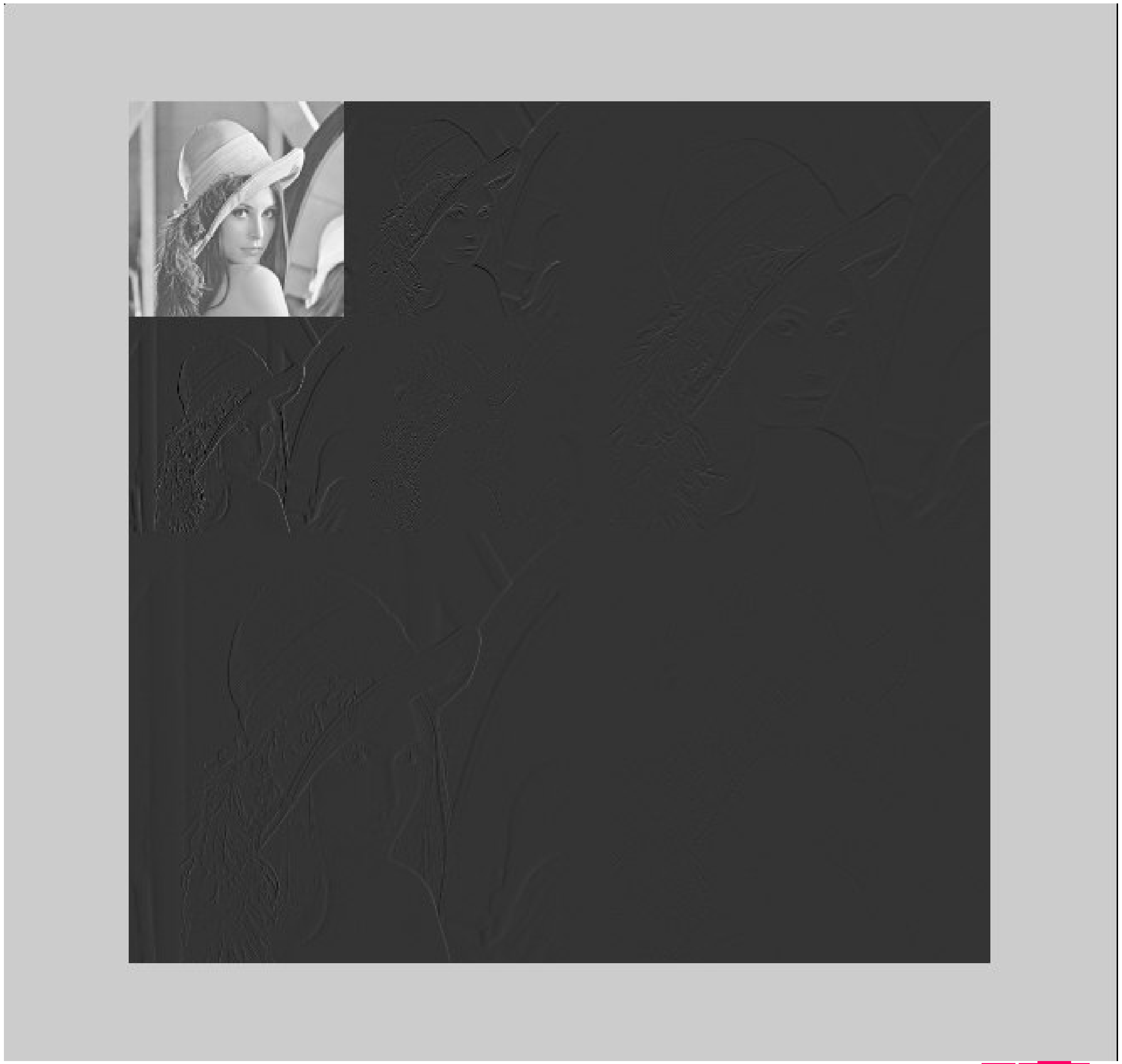
CV … vertical coefficients

CD … diagonal coefficients

Führt nur einen Level der Decomposition durch.

dwt2

```
I = double(imread('lenna.bmp'));
% first level
[CA,CH,CV,CD] = DWT2(I,'haar');
figure;
imshow([CA,CH;CV,CD],[]);
% second level
[CA2,CH2,CV2,CD2] = DWT2(CA,'haar');
figure;
imshow([[CA2,CH2;CV2,CD2],CH;CV,CD],[]);
```



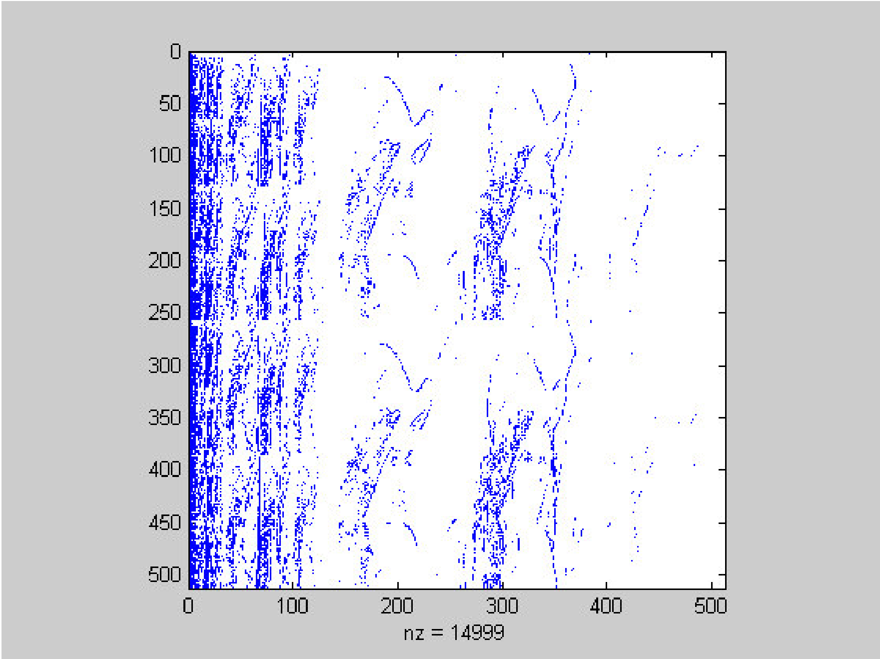
Bildkompression

- Verlustbehaftete Kompression
- Wavelet-Transformation anwenden
- Koeffizienten die sehr klein sind werden durch 0 ersetzt (-> Verlust von Information)
- Nur mehr Koeffizienten speichern die ungleich 0 sind.

Bildkompression

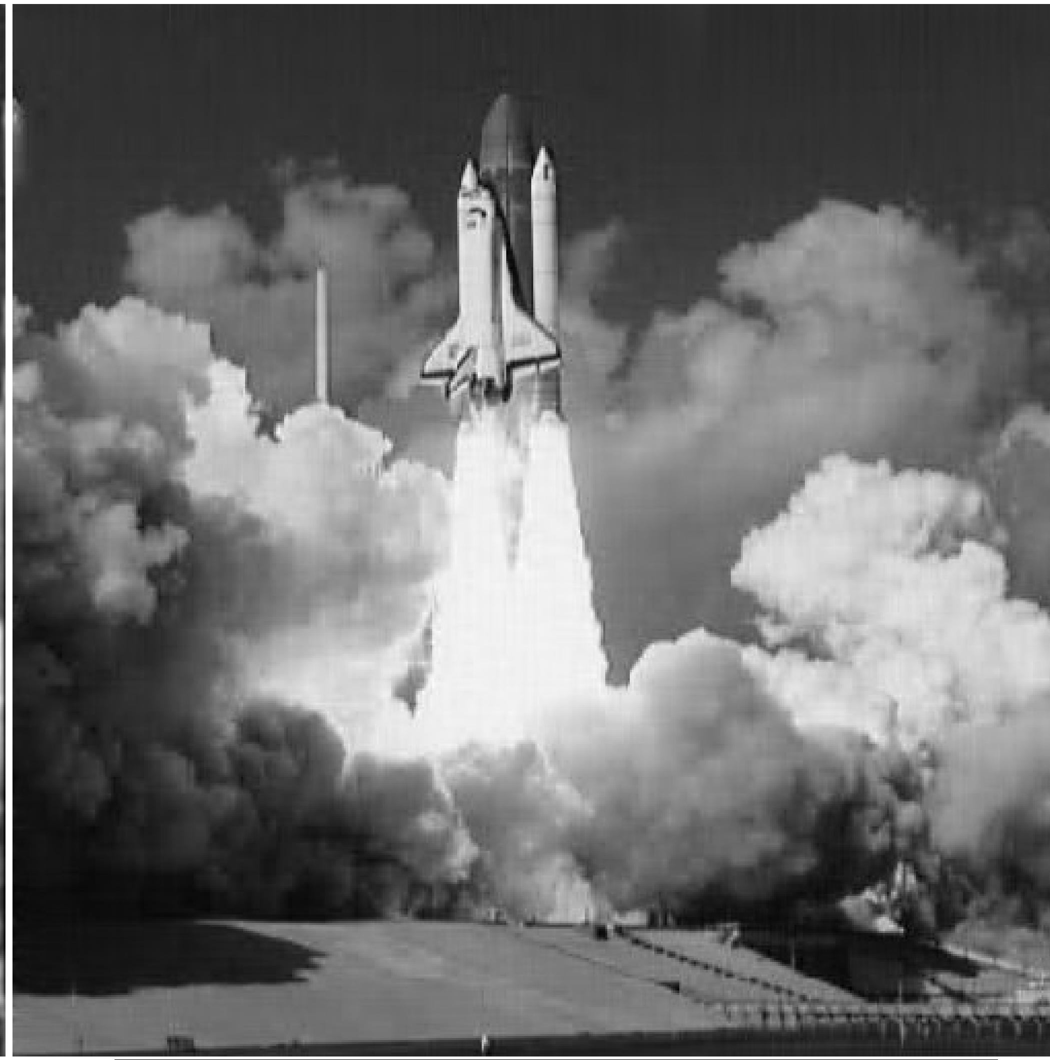
```
I = double(imread('lenna.bmp'));
wtype = 'db2';
[C S] = wavedec2(I, wmaxlev(I,wtype), wtype);
[B IX] = sort(abs(C));
B = fliplr(B);
IX = fliplr(IX);
CF = C;
CF(IX(15000:length(IX))) = 0;

X = waverec2(CF,S,wtype);
figure;
imshow(I,[]);
figure;
imshow(X,[]);
```



Vergleich FFT - Wavelet

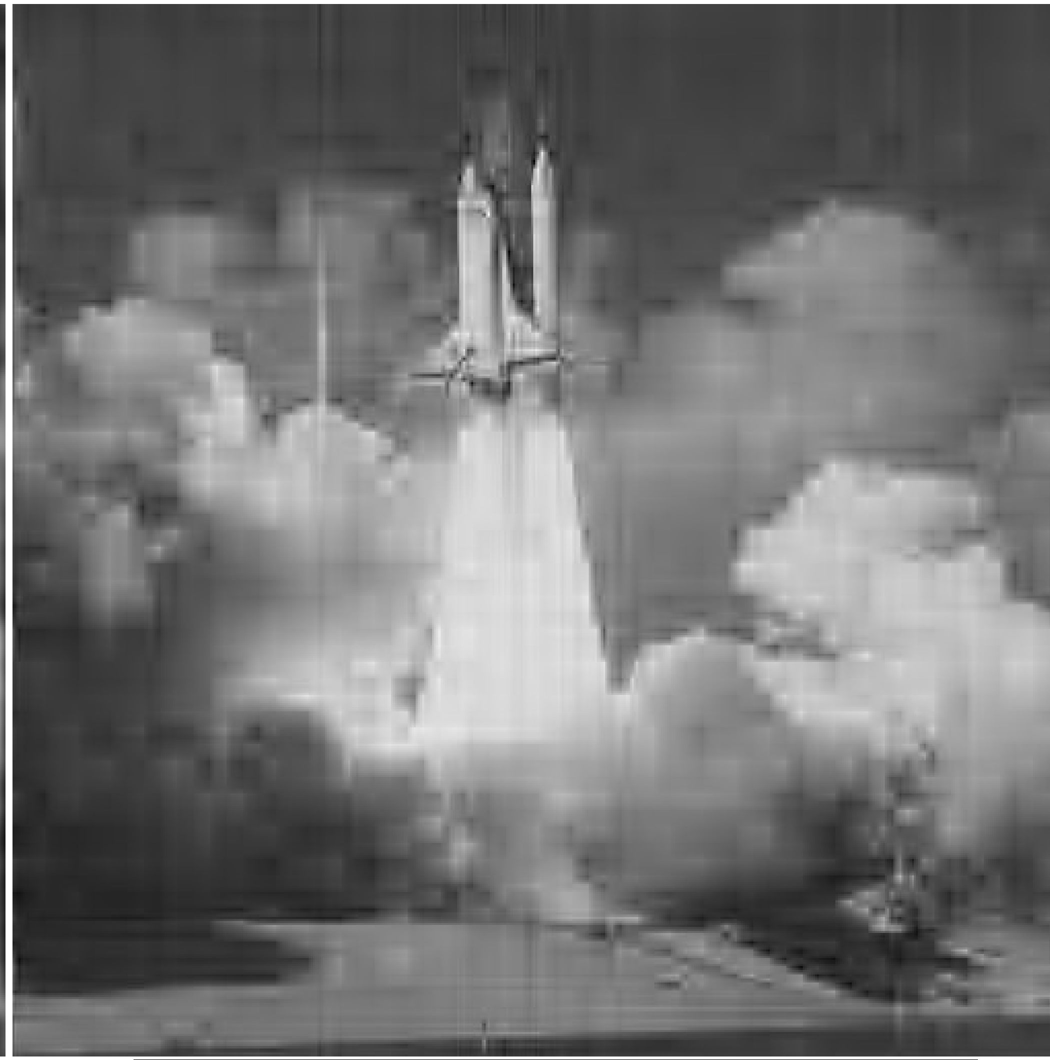
13359 FFT-Koeffizienten



13678 WVLT-Koeffizienten

Vergleich FFT - Wavelet

1363 FFT-Koeffizienten



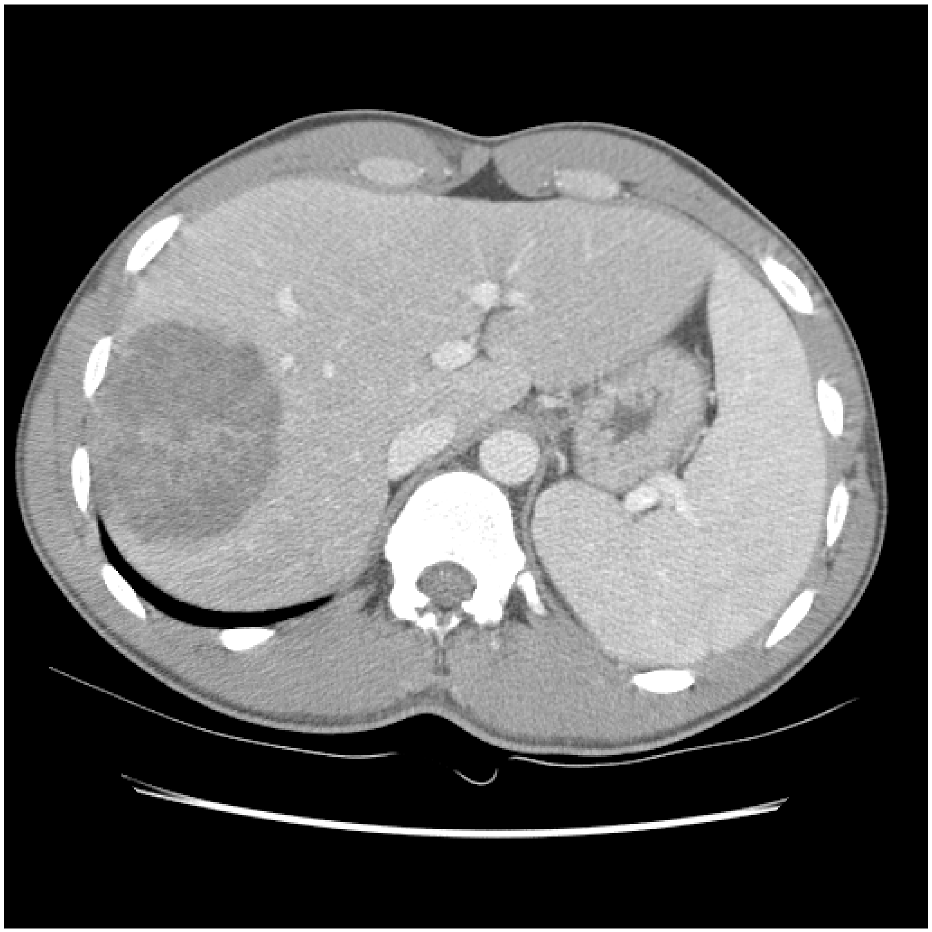
1366 WVLT-Koeffizienten

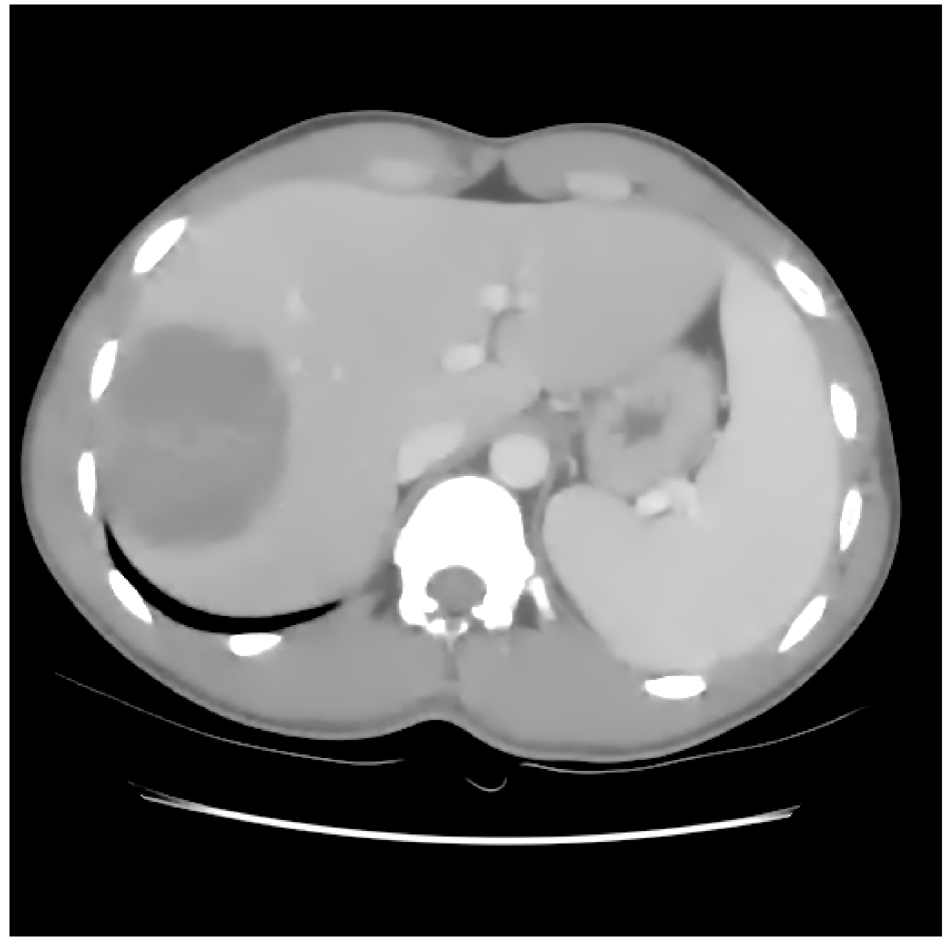
Vergleich FFT - Wavelet

519 FFT-Koeffizienten

532 WVLT-Koeffizienten

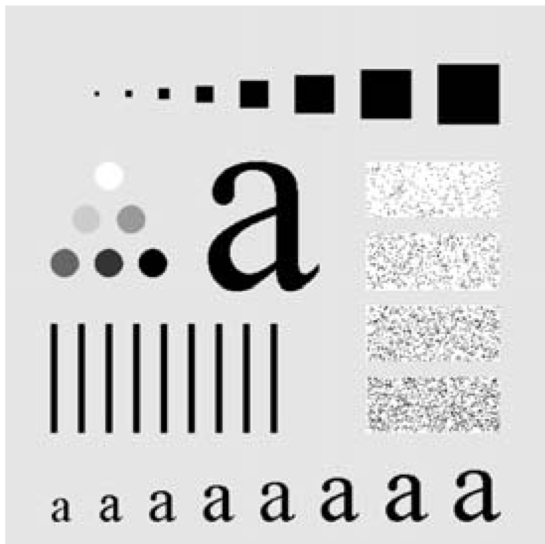
Beispiel: Kanten bleiben gut erhalten

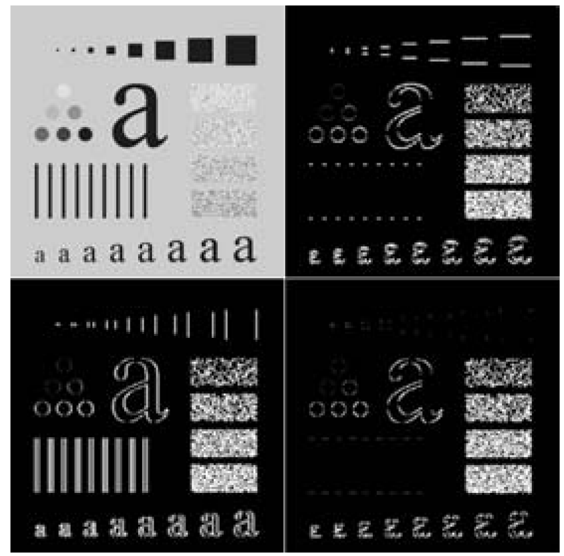




Kantendetektion

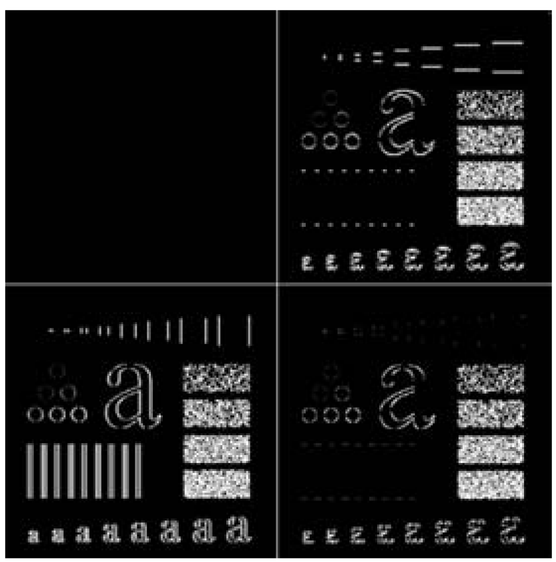
Original





Wavelet-
Decomposition

Approximation
Teil wird auf 0
gesetzt



Wavelet-
Reconstruction

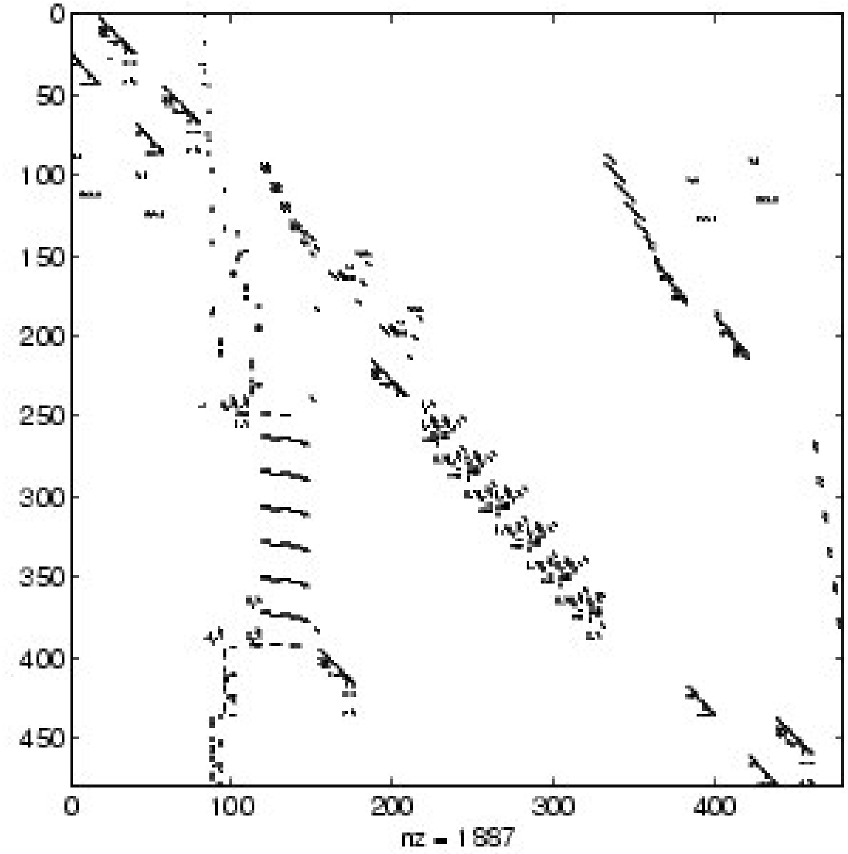
Nur mehr
Kanteninformation
enthalten

Lösen von Gleichungssystemen

- $Ax=b$
- Große Matrizen A führen zu langen Rechenzeiten
- Abhängig von der Gestalt der Matrix A können unterschiedliche Algorithmen schnellere Laufzeiten ermöglichen

Sparse Matrices

- Sparse Matrices = Schwach besetzte Matrizen
- Bei vielen Anwendungen entstehen Gleichungssysteme bei denen nur wenige Elemente Werte ungleich 0 enthalten.
- Es gibt Methoden die solche Gleichungssysteme schneller lösen



Sparse Matrices durch Wavelet-Decomposition

$$Ax = b$$

$$(\psi A \psi^{-1})(\psi x) = (\psi b)$$

Erweitern des ursprünglichen Gleichungssystems mit ψ, was eine Wavelettransformation darstellt.

Das neue Gleichungssystem ist dann sparse und kann schneller gelöst werden.

Der Lösungsvektor (ψx) kann durch die inverse Wavelettransformation (ψ^{-1}x) in den gesuchten Lösungsvektor x zurücktransformiert werden.

Sparse Matrices durch Wavelet-Decomposition

Bsp: A … 4x4 Matrix

x,b … 4 Vektor

ψ = (Haar-Wavelet)

1	1	1	1
1	-1	0	0
1	1	-1	-1
0	0	1	-1

A =

1.9225	0.6175	-0.0200	-0.0200
0.6125	1.9075	-0.0100	-0.0100
-0.0075	-0.0125	4.6600	-2.1400
-0.0175	-0.0225	-2.1300	4.6700

$\psi A \psi^{-1}$ =

2.5000	0.0100	0	0
0	1.3000	0.0100	0
-0.0000	0	2.5600	0
0.0000	0	0.0100	6.8000

Zusammenfassung

- Haar Wavelets
- Standard/Non-Standard decomposition
- Bildkompression, Effizientes Lösen von Gleichungssystemen, Kantendetektion