



Berner Fachhochschule
Haute école spécialisée bernoise
Bern University of Applied Sciences

Introduction to Computer Vision:

Global Operations: Wavelet Transform

Marcus Hudritsch (hsm4)



Wavelet Transform

Module 7281, CPVR1: Introduction to Computer Vision

Marcus Hudritsch, Berne University of Applied Sciences

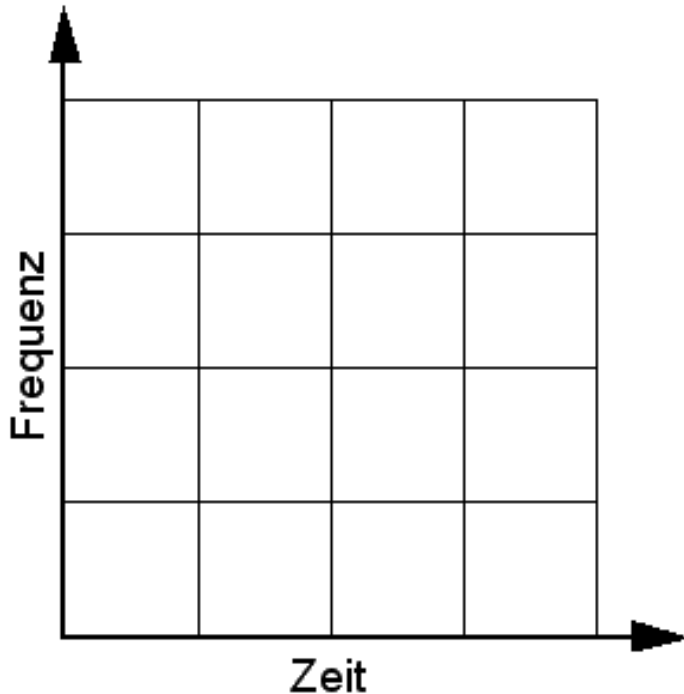
- The foundation of the **Wavelet theory** goes back to the 1930's.
- The today's wavelet transform was invented by the **French physicists Jean Morlet** and **Alex Grossmann** in the 1980's.
- A good introduction to wavelets is by **Barbara Burke**:
„Wavelets: Die Mathematik der kleinen Wellen“ [Burke97].

All images in this presentation are from this book.

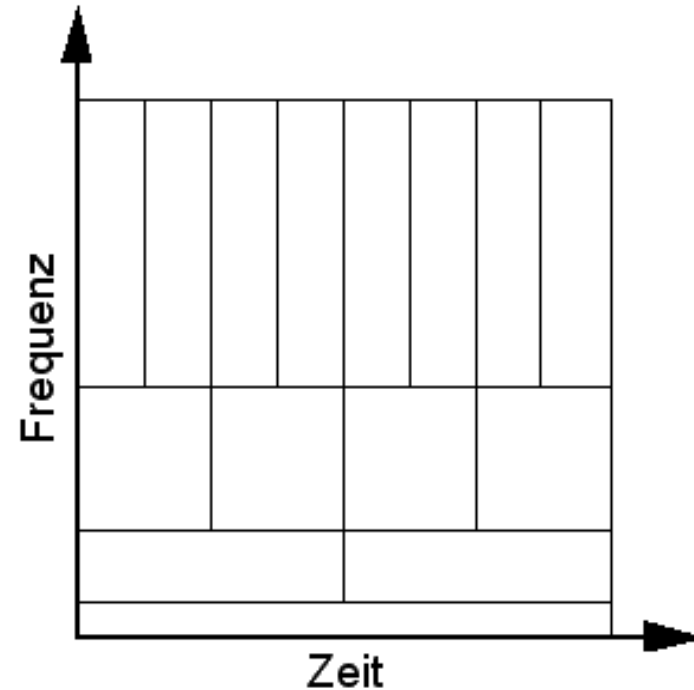
- The motivation behind wavelets where some problems with the FT.

Wavelet Transform vs. Windowed Fourier Transform

- The discrete Windowed Fourier transform (WFT) remains static.
- The wavelet transform analyses the signal in multiple resolutions:

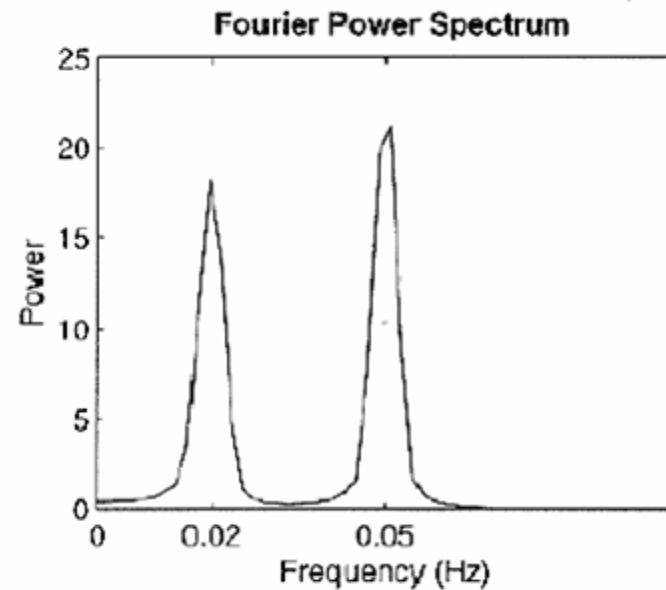
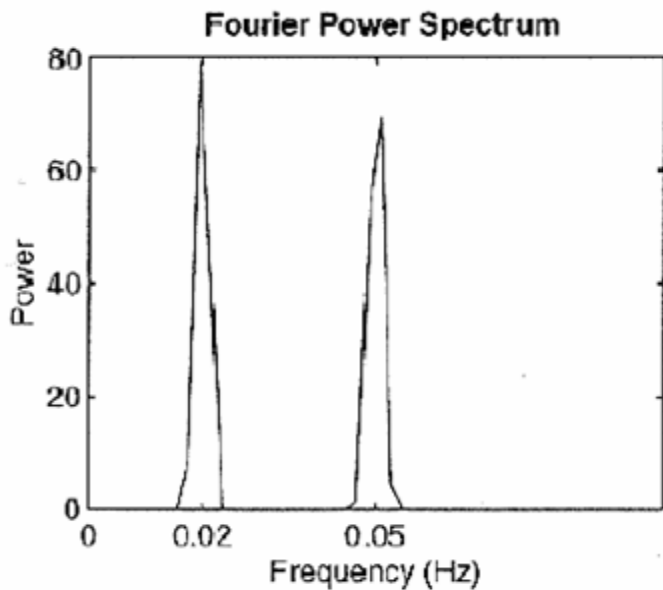
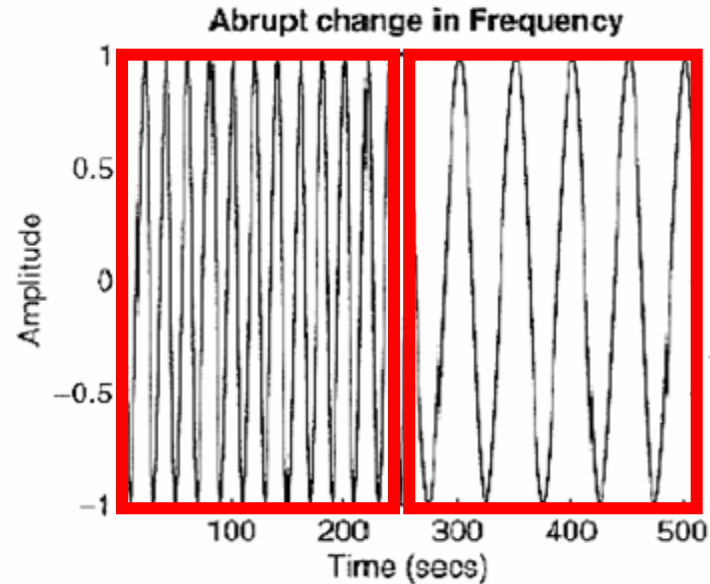
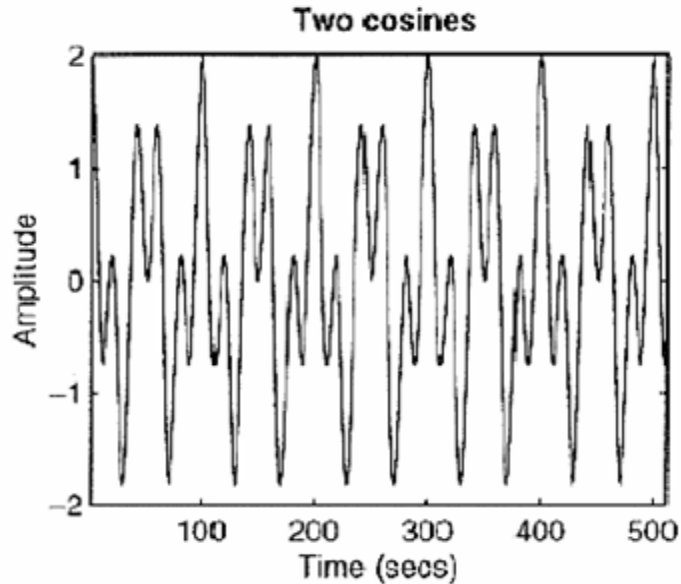


Static time-frequency
resolution in FT



Multiple Resolution in WT

Wavelet Transform vs. Windowed Fourier Transform

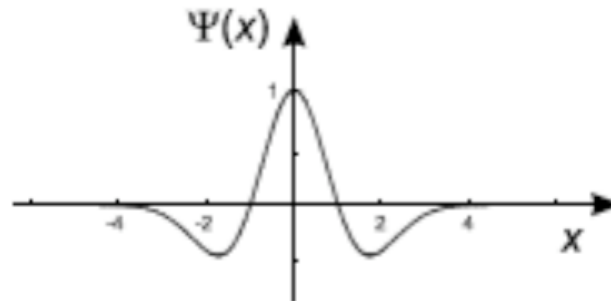


Wavelet Transform: Base or Mother Wavelets

- Wavelets are functions that are used for sampling.
- By sampling we calculate the cover ration with the underlying signal section.
- Wavelets can be scaled and shifted.
- The unscaled and unshifted wavelet is called **base** or **mother wavelet**.

Basiswavelet :

Mexican hat



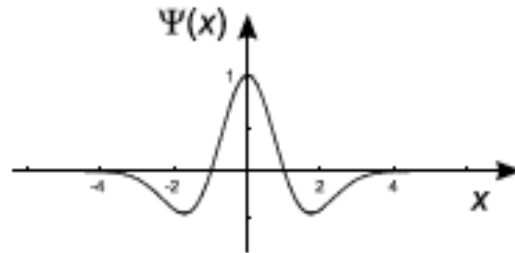
$$\Psi(x) = (1 - x^2) e^{-x^2/2}$$

Wavelet Transform: Base Wavelet

- All versions of the base wavelet together are the **wavelet family**.

Basiswavelet :

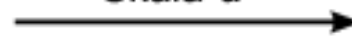
Mexican hat



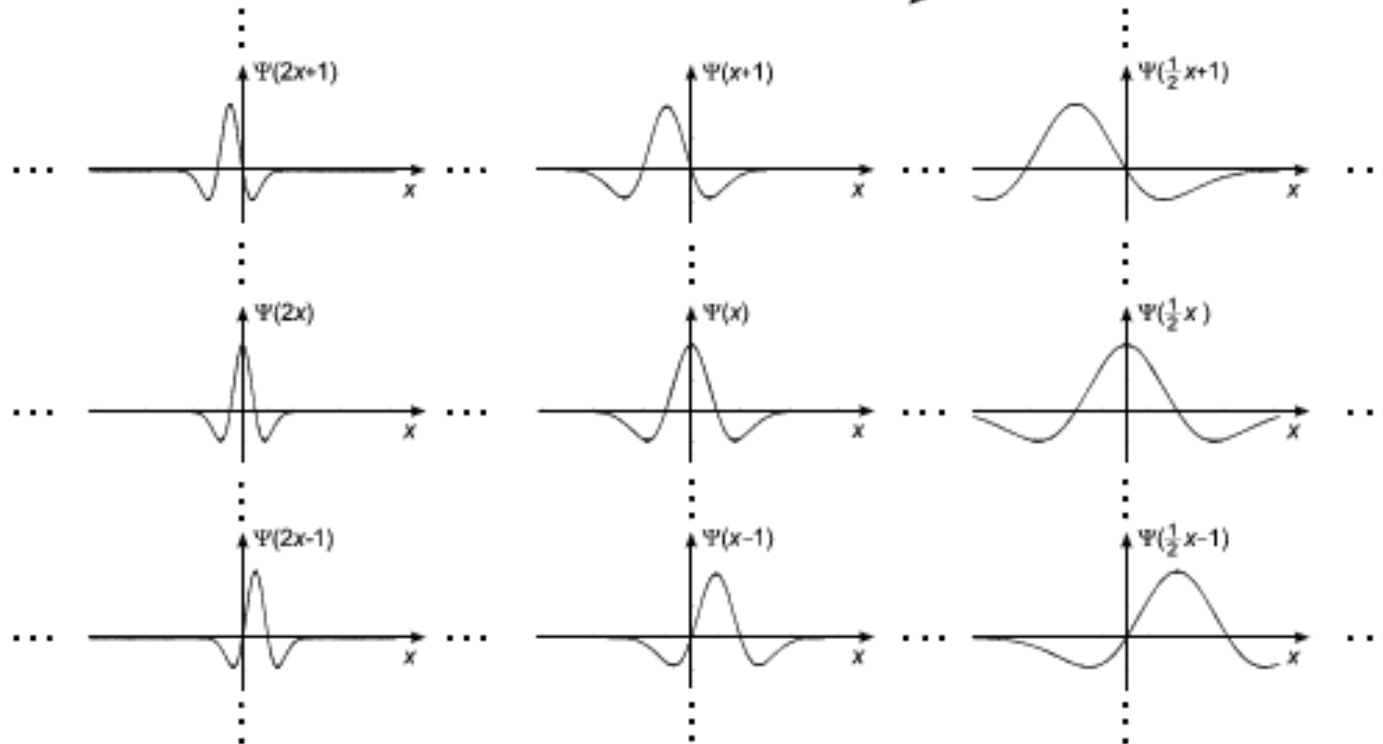
$$\Psi(x) = (1 - x^2) e^{-x^2/2}$$

Waveletfamilie:

Skala a



Ort b



Wavelet Transform: Base Wavelet

Base wavelets Ψ must fulfill the following conditions:

- The area of the wavelet must be zero:

$$\int \psi(x) dx = 0$$

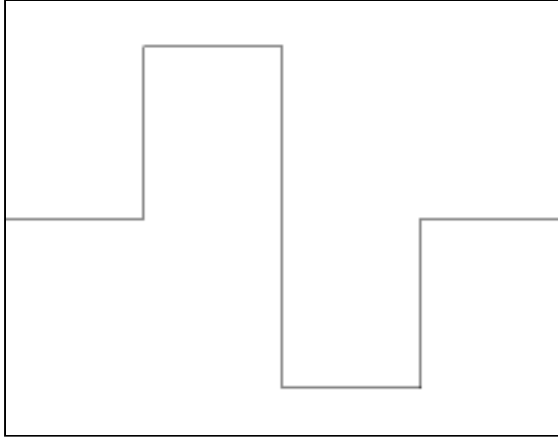
- A wavelet $\Psi(a,b)$ of a wavelet family is defined as follows:

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

where **a** is the scale factor and **b** is the shift factor.

Wavelet Transform: Base Wavelet

Some well known base wavelets:



Haar wavelet



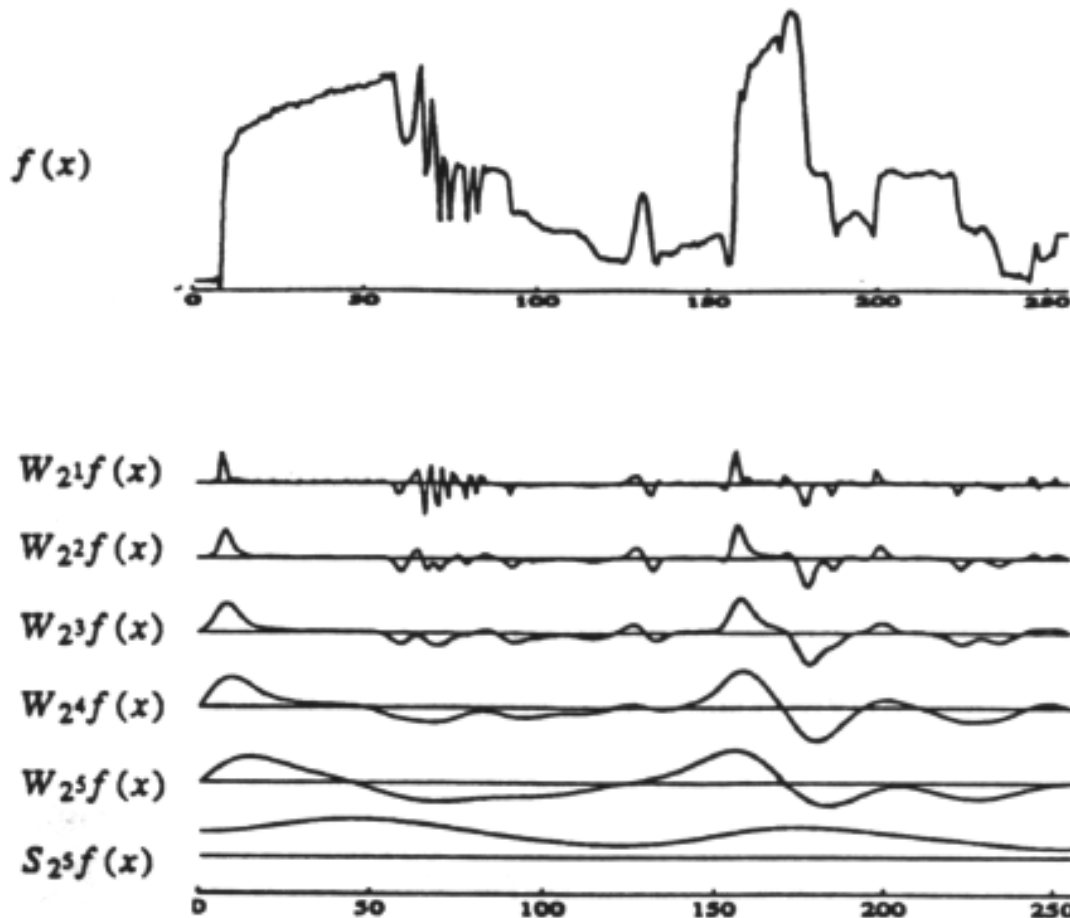
Daubechies 4 wavelet



Daubechies 20 wavelet

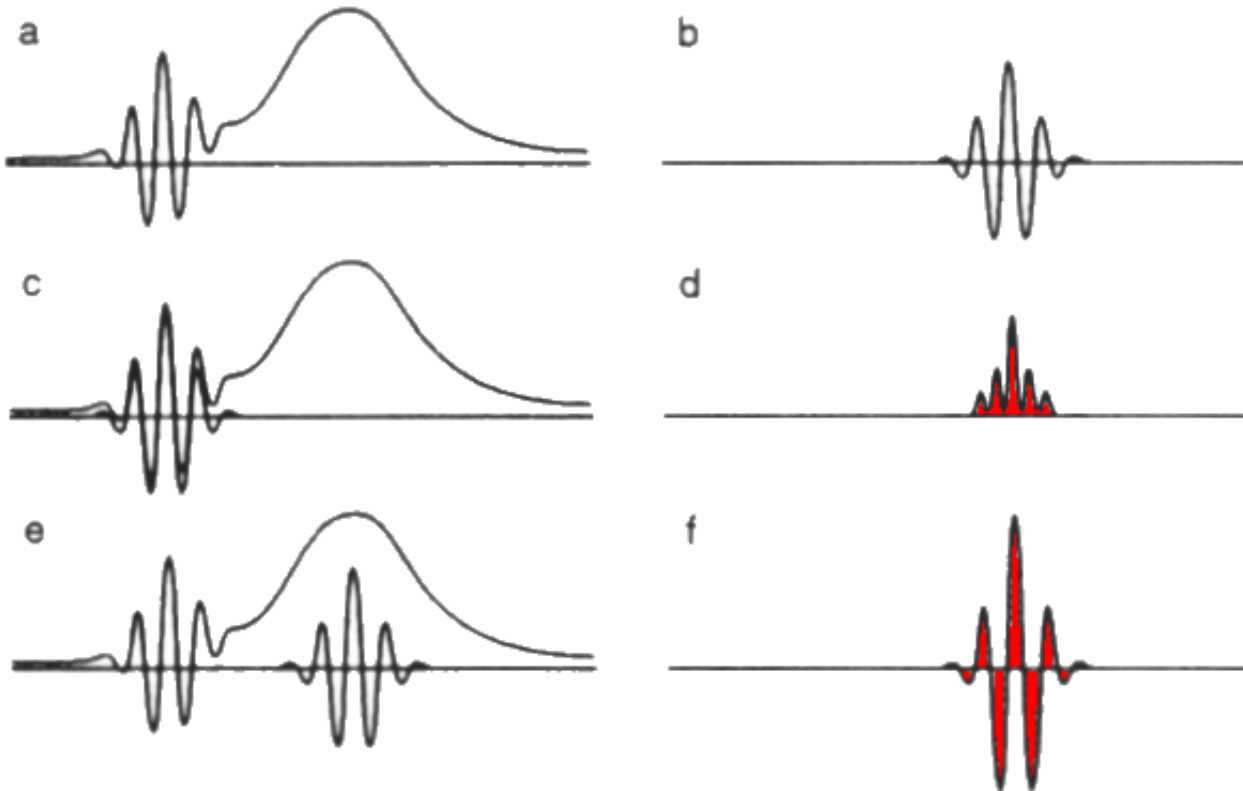
Wavelet Transform: Continuous WT

- By sampling the signal with scaled wavelets we get the wavelet transform.
- It corresponds to the cover ratio of the wavelets with the signal.
- The reconstruction is done as in the FT by summing up all wavelet coefficients.



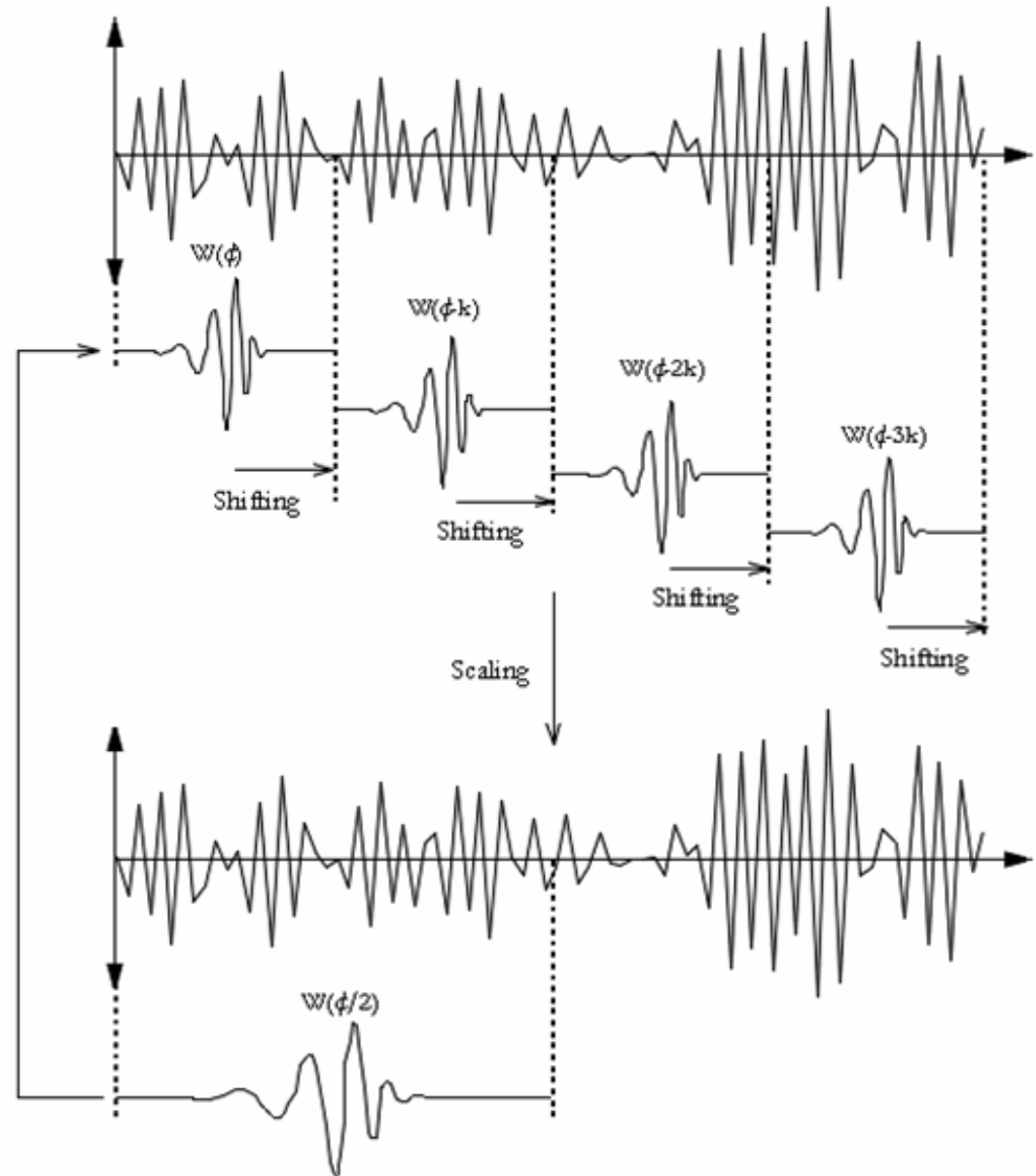
Wavelet Transform: Continuous WT

- How do we calculate the cover ratio of the wavelet with the signal?
- You multiply the wavelet function with the signal section.
- We calculate the area of the resulting function with integration.
- This value is called **Wavelet coefficient**.



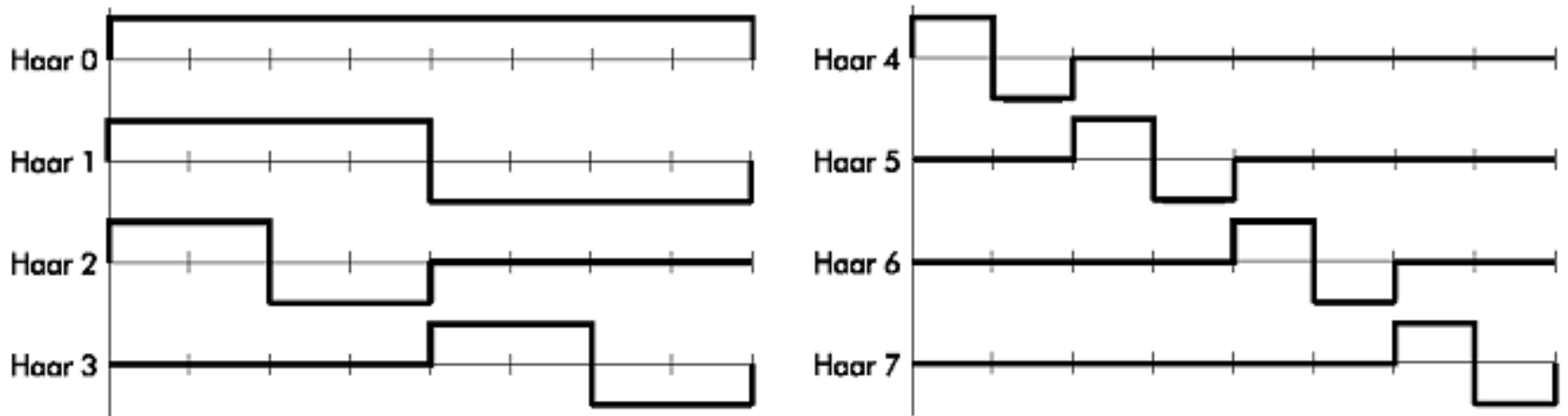
Wavelet Transform: Discrete WT

- The **discrete WT (DWT)** samples the signal in **discrete step distances**.
- The break through came in 1986 with the **Fast Wavelet Transform (FWT)** from *Stéphane Mallat* und *Yves Meyer*



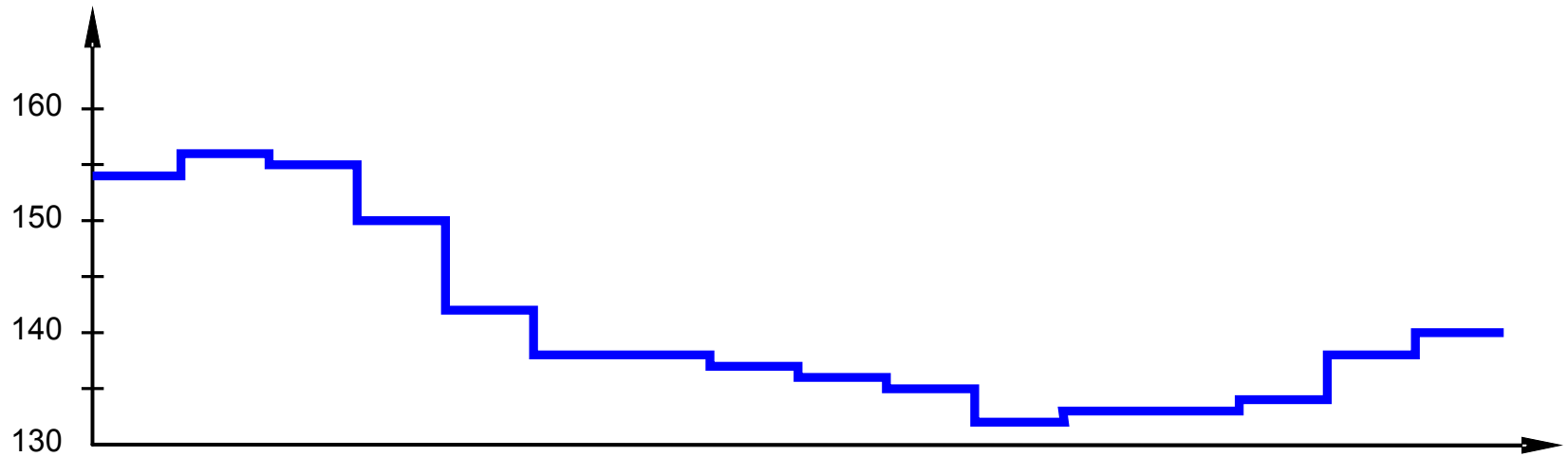
Wavelet Transform: 1D Haar Transform

- The simplest Wavelet transform is the ***Haar transform***.
- It is a simple and hard function.
- Through scaling and shifting we get the Haar Wavelet family:



Wavelet Transform: 1D Haar Transform

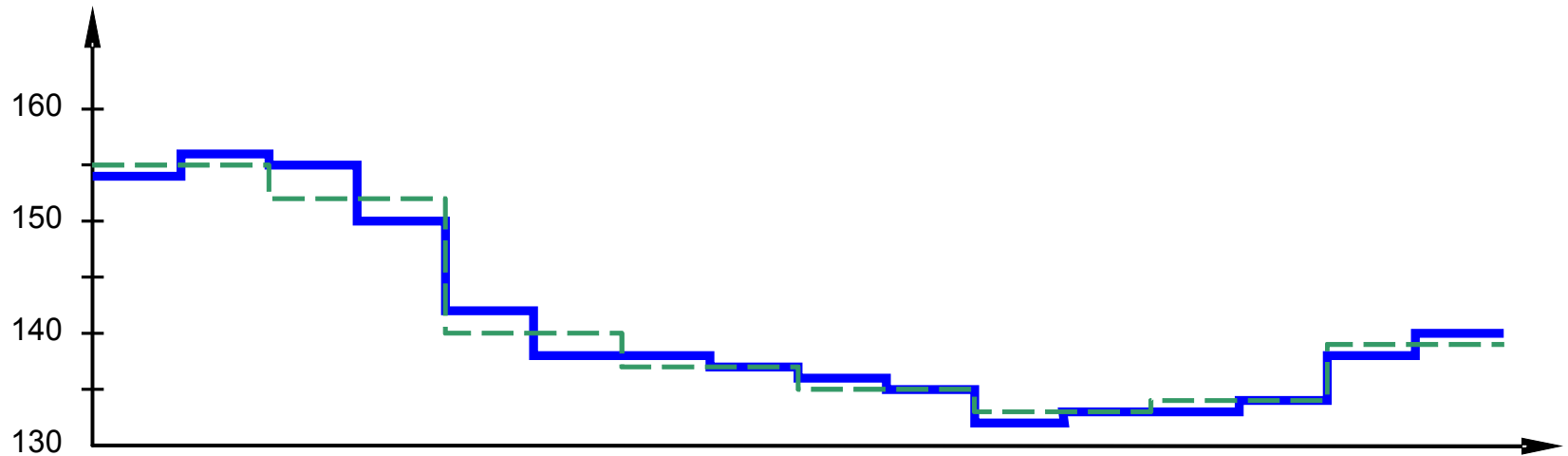
- We build the average of neighbouring cells (approximation).
- In the lines in between we write the difference (detail)



Signal	154	156	155	150	142	138	138	137	136	135	133	134	134	135	138	140
--------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

Wavelet Transform: 1D Haar Transform

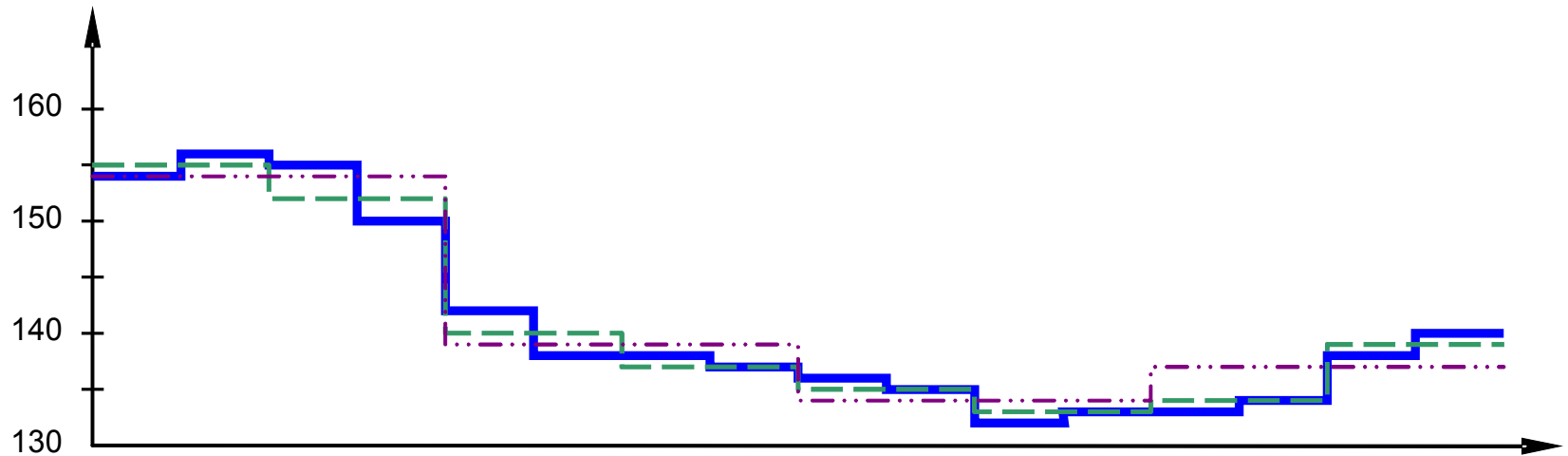
- We build the average of neighbouring cells (approximation).
- In the lines in between we write the difference (detail)



Signal	154	156	155	150	142	138	138	137	136	135	133	134	134	135	138	140
Detail 1	-1	1	2.5	-2.5	2	-2	0.5	-0.5	0.5	-0.5	-0.5	0.5	-0.5	0.5	-1	1
Approximation 1	155		152.5		140		137.5		135.5		133.5		134.5		139	

Wavelet Transform: 1D Haar Transform

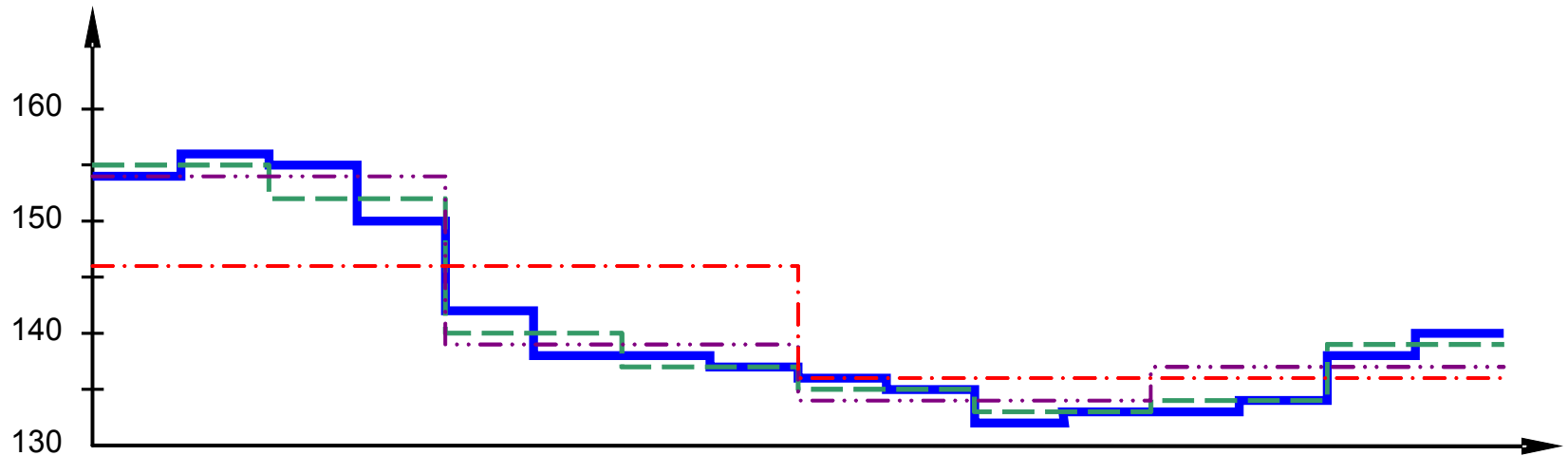
- We build the average of neighbouring cells (approximation).
- In the lines in between we write the difference (detail)



Signal	154	156	155	150	142	138	138	137	136	135	133	134	134	135	138	140
Detail 1	-1	1	2.5	-2.5	2	-2	0.5	-0.5	0.5	-0.5	-0.5	0.5	-0.5	0.5	-1	1
Approximation 1	155		152.5		140		137.5		135.5		133.5		134.5		139	
Detail 2	1.25		-1.25		1.25		-1.25		1		-1		-2.25		2.25	
Approximation 2	153.75				138.75				134.5				136.75			

Wavelet Transform: 1D Haar Transform

- We build the average of neighbouring cells (approximation).
- In the lines in between we write the difference (detail)



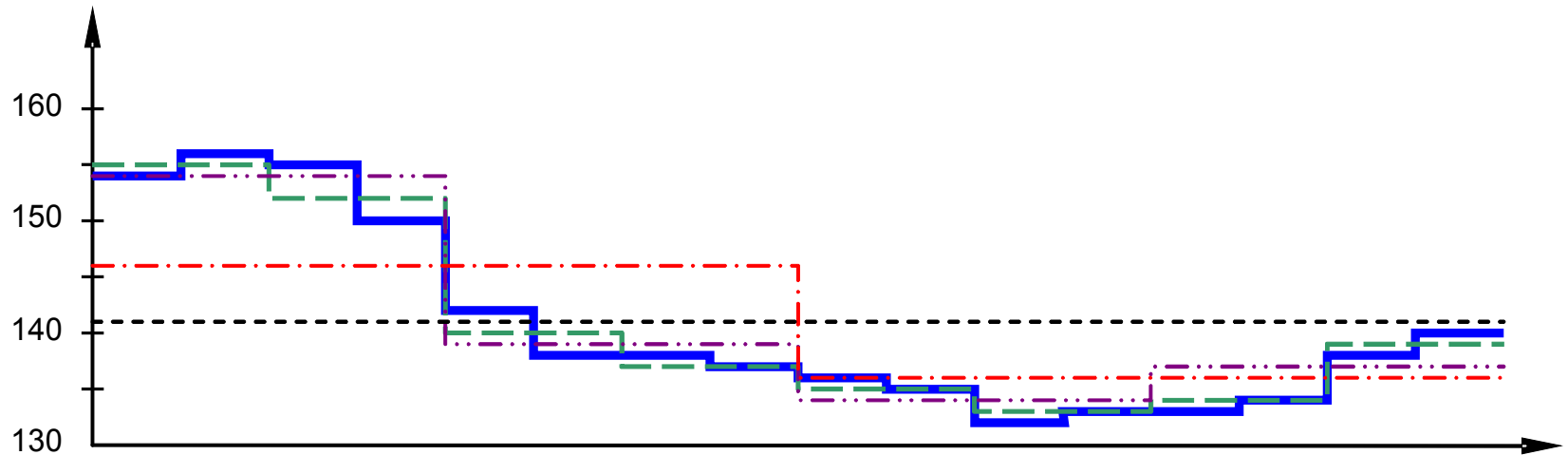
Signal	154	156	155	150	142	138	138	137	136	135	133	134	134	135	138	140
Detail 1	-1	1	2.5	-2.5	2	-2	0.5	-0.5	0.5	-0.5	-0.5	0.5	-0.5	0.5	-1	1
Approximation 1	155		152.5		140		137.5		135.5		133.5		134.5		139	
Detail 2	1.25		-1.25		1.25		-1.25		1		-1		-2.25		2.25	
Approximation 2	153.75				138.75				134.5				136.75			
Detail 3	7.5				-7.5				-1.125				1.125			
Approximation 3	146.25								135.625							

Wavelet Transform: 1D Haar Transform

- We 1 average value and 15 detail coefficients:

140.9375, 5.3125, 7.5, -1.125, 1.25, 1.25, 1, -2.25, -1, 2.5, 2, 0.5, 0.5, -0.5, -0.5, -1

- Because the details are symmetric, we store only one detail value:



Signal	154	156	155	150	142	138	138	137	136	135	133	134	134	135	138	140
Detail 1	-1	1	2.5	-2.5	2	-2	0.5	-0.5	0.5	-0.5	-0.5	0.5	-0.5	0.5	-1	1
Approximation 1	155		152.5		140		137.5		135.5		133.5		134.5		139	
Detail 2	1.25		-1.25		1.25		-1.25		1		-1		-2.25		2.25	
Approximation 2	153.75				138.75				134.5				136.75			
Detail 3	7.5				-7.5				-1.125				1.125			
Approximation 3	146.25								135.625							
Detail 4	5.3125								-5.3125							
Approximation 4	140.9375															

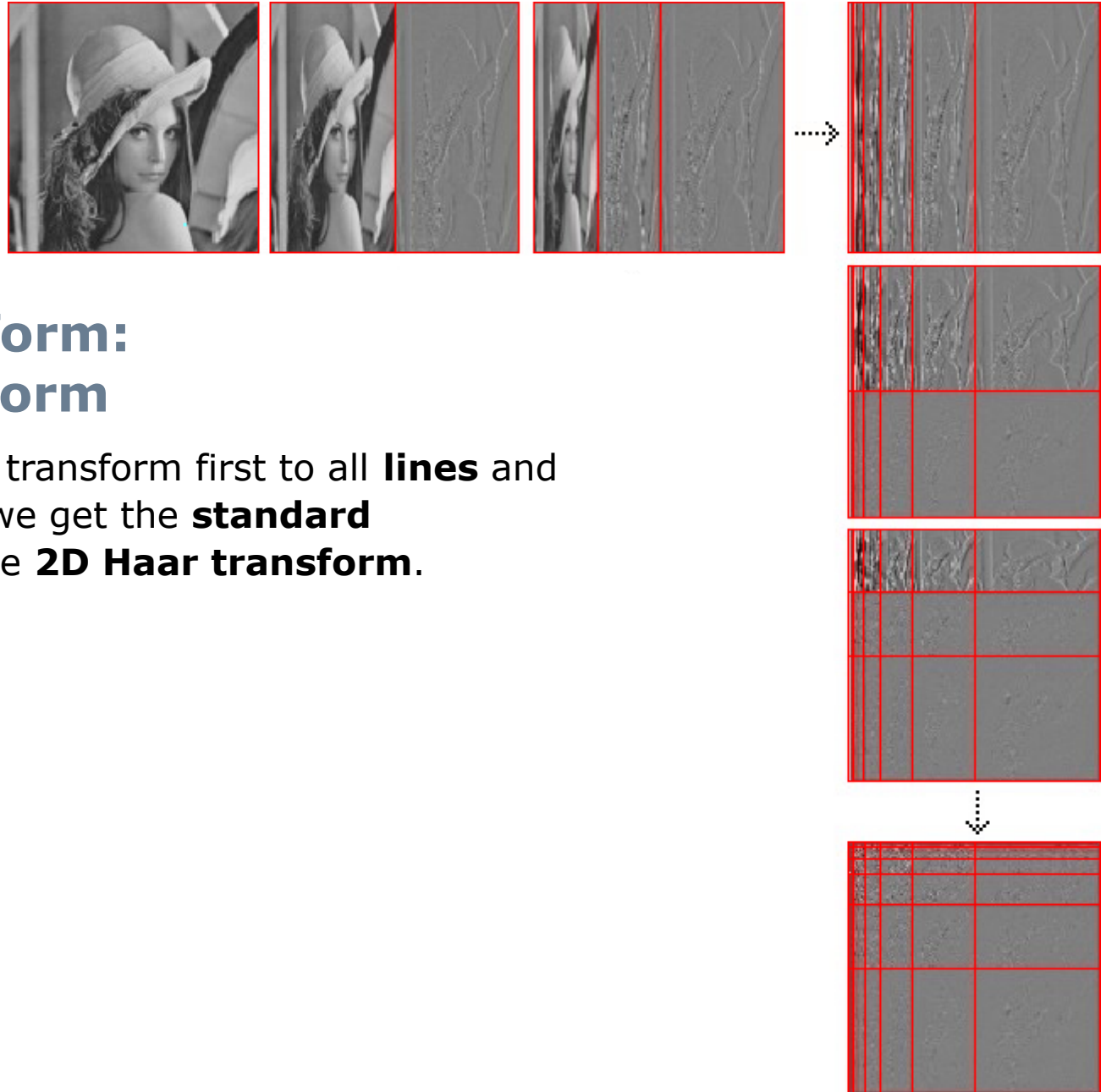
Wavelet Transform: 1D Haar Transform

- The forward transform is called **analysis**.
- The **approximation** and **details** can also be built with **filter masks**:
- This is a **1D convolution** (2 multiplies & 1 add):

$$h_0 = \left[\frac{1}{2}, \frac{1}{2} \right] \text{ low pass filter (approximation)} \quad h_1 = \left[\frac{1}{2}, -\frac{1}{2} \right] \text{ high pass filter (detail)}$$

- The **reconstruction** called **synthesis** has the following filter masks:

$$f_0 = [1, 1] \text{ left synthesis filter} \quad f_1 = [1, -1] \text{ right synthesis filter}$$



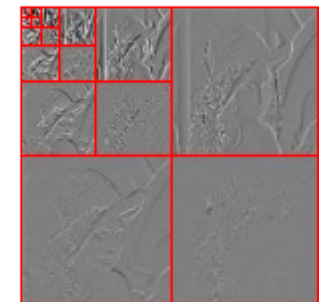
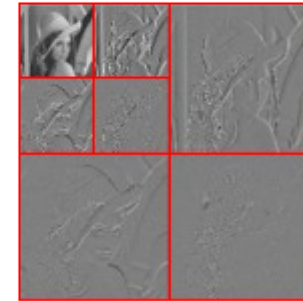
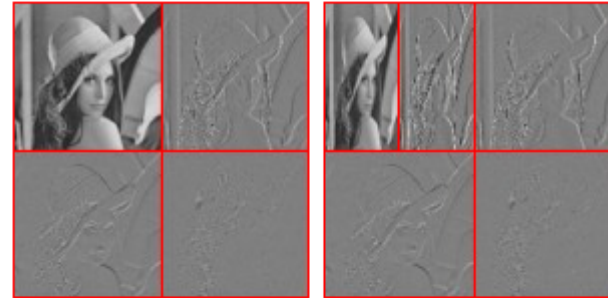
Wavelet Transform: 2D Haar Transform

- If you apply the Haar transform first to all **lines** and then to all **columns** we get the **standard decomposition** of the **2D Haar transform**.



Wavelet Transform: 2D Haar Transform

- If you apply the Haar transform alternately to all **lines** and **columns** we get the **non-standard decomposition** of the **2D Haar transform**.



Wavelet Transform: 2D Haar Transform

- Image transformed with this hard Haar transform tend to get block artefacts and to be noisy.
- The normalized Haar transform is better.

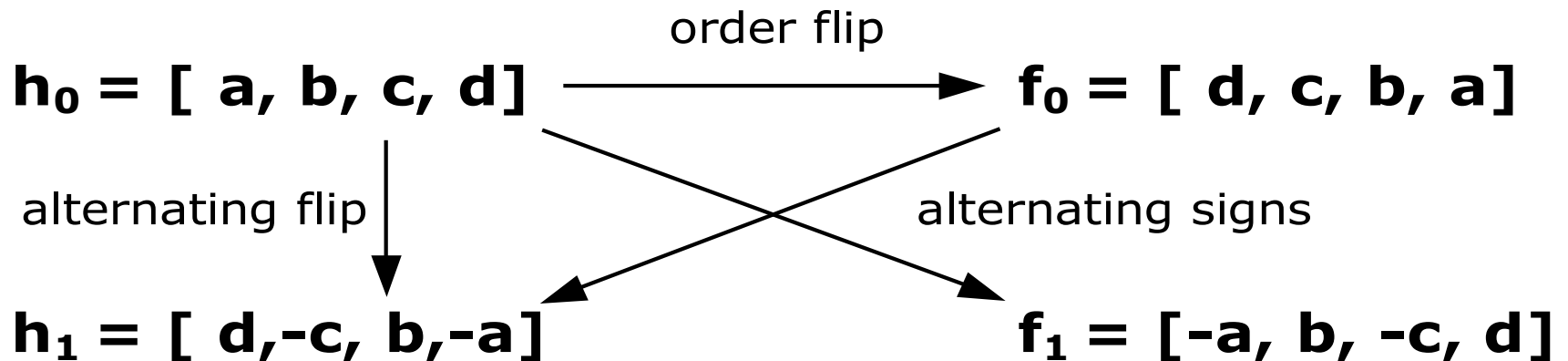
$$h_0 = \left[\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right] \text{ low pass filter (approximation)} \quad h_1 = \left[\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}} \right] \text{ high pass filter (detail)}$$

The reconstruction has the following filter masks:

$$f_0 = \left[\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right] \text{ left synthesis filter} \quad f_1 = \left[\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}} \right] \text{ right synthesis filter}$$

Wavelet Transform: Filter bank

- Other wavelet transforms have wider filter masks.
- We always need a low pass, a high pass and two synthesis filters.
- This set of filters is also called a **filter bank**.
- There is a specific relation between the filters.
- We therefore only need to define the low pass filter:



Wavelet Transform: Filter bank

- **Normalized Haar (filter length 2)**

`h0 = [0.70710678118655, 0.70710678118655]`

`h1 = [0.70710678118655, -0.70710678118655]`

`f0 = [0.70710678118655, 0.70710678118655]`

`f1 = [0.70710678118655, 0.70710678118655]`

- **Daubechies 4: (filter length 8)**

`h0 = [0.23037781330886, 0.71484657055254, 0.63088076792959, -0.02798376941698,
-0.18703481171888, 0.03084138183599, 0.03288301166698, -0.010597401785]`

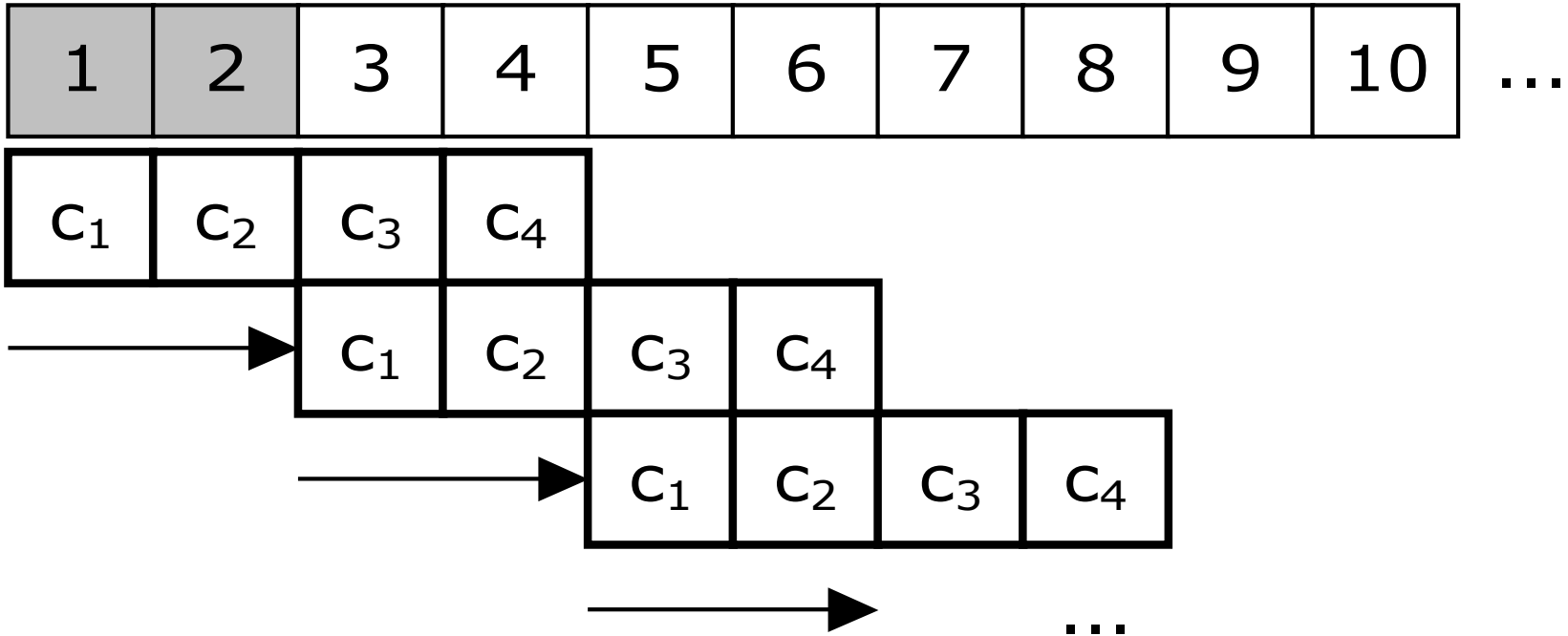
`h1 = [-0.010597401785, -0.03288301166698, 0.03084138183599, 0.18703481171888,
-0.02798376941698, -0.63088076792959, 0.71484657055254, -0.23037781330886]`

`f0 = [-0.010597401785, 0.03288301166698, 0.03084138183599, -0.18703481171888,
-0.02798376941698, 0.63088076792959, 0.71484657055254, 0.23037781330886]`

`f1 = [-0.23037781330886, 0.71484657055254, -0.63088076792959, -0.02798376941698,
0.18703481171888, 0.03084138183599, -0.03288301166698, -0.010597401785]`

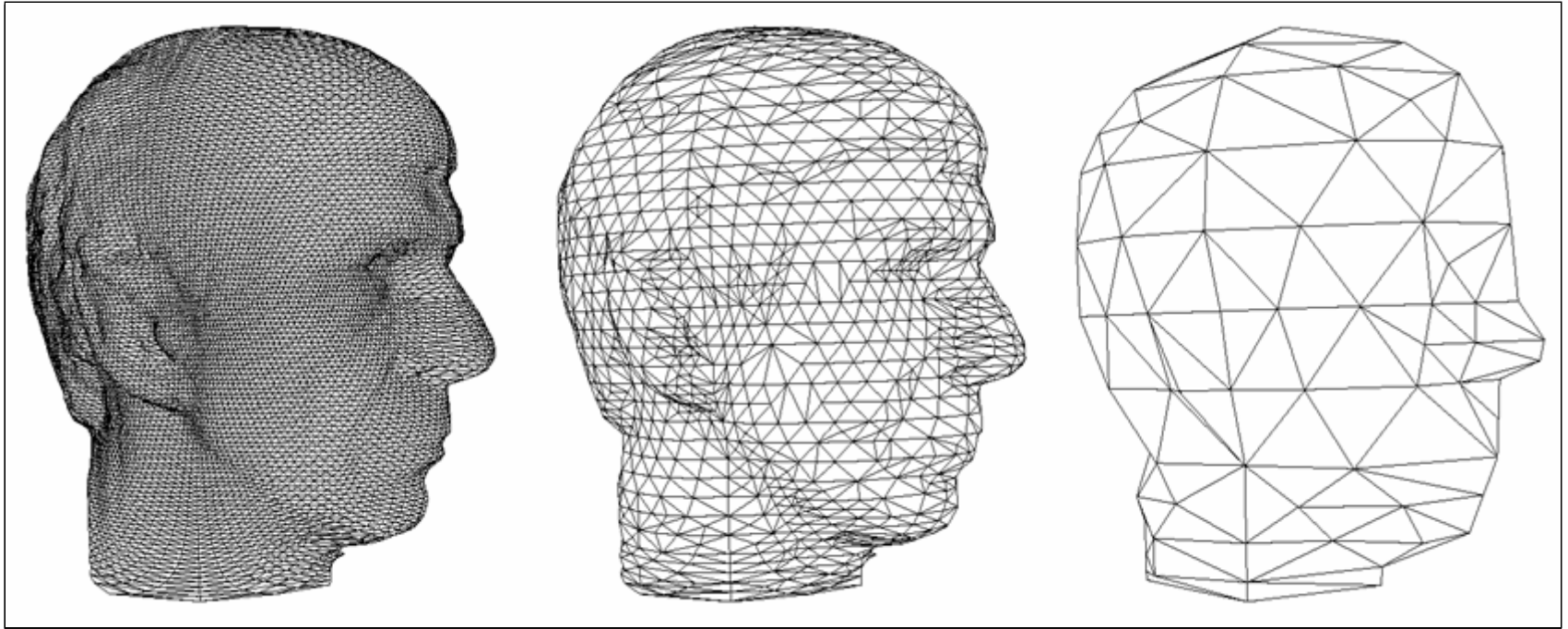
Wavelet Transform: Filter bank

- No matter how wide the filters are, they are move allways 2 cells.
- For filters wider than 2 we get a border problem:



Wavelet Transform: Applications

- Image compression
- Multi scalen resolution (level of detail)



- Audio analysis

Wavelet Transform: Applications

- Reconstruct the following 1D Haar transform:

89.6875,

1.6875, -5.875, 3.25, -4.5, -0.25, 2.25, 0.75, -1, -4, -1, -0.5, 1.5, 1, 0.5, 0

- The first value is called DC. Why?