Digitálne spracovanie zvuku, obrazu a biosignálov - DSZOB

Wavelets and JPEG 2000

Wavelets &
Wavelet Transform

What is a wavelet?

A basis function that is isolated with respect to

- time or spatial location
- frequency or wave number

- Each wavelet has a characteristic location and scale
- Recomended videos:

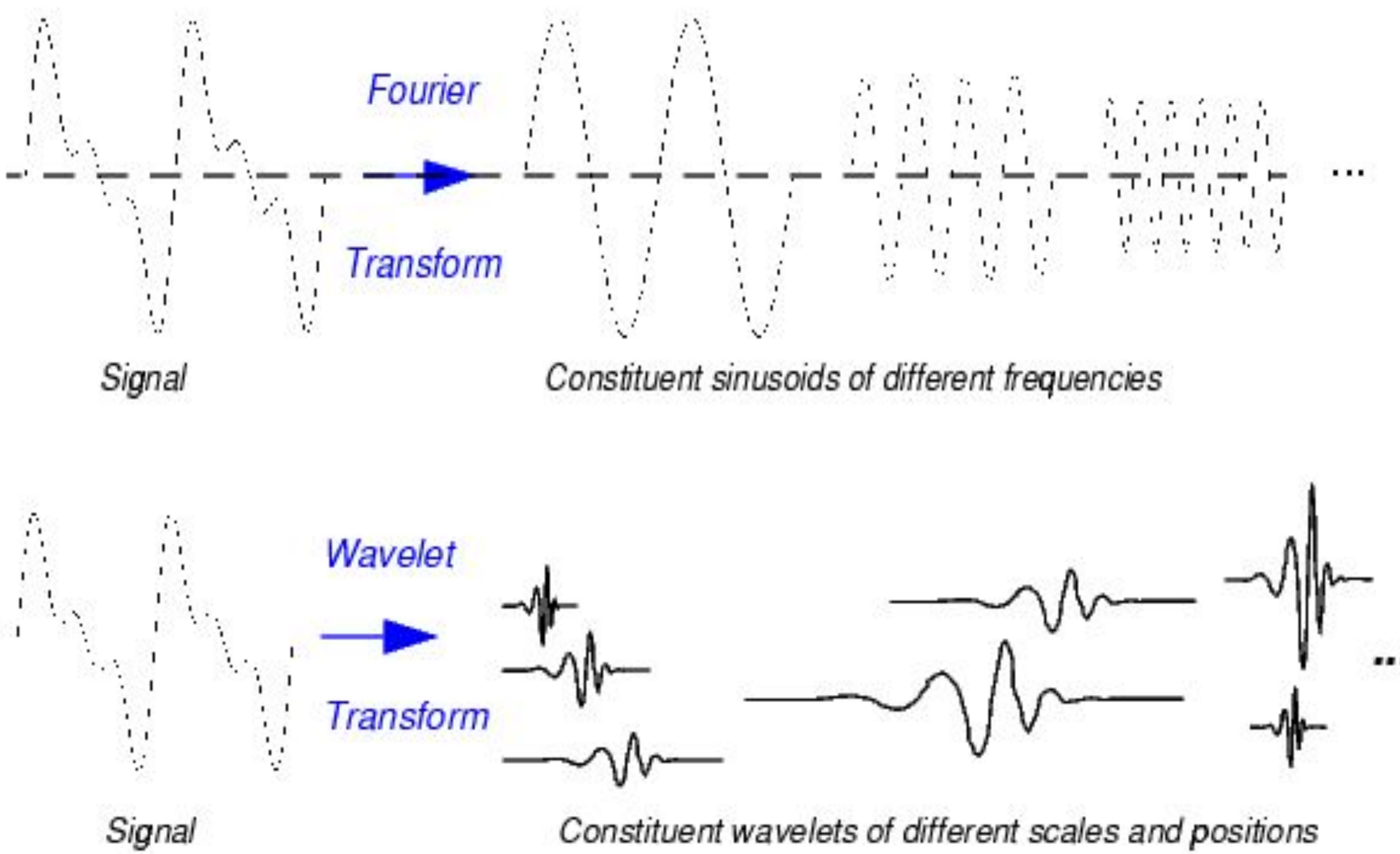
https://de.mathworks.com/videos/series/understanding-wavelets-121287.html

Why?

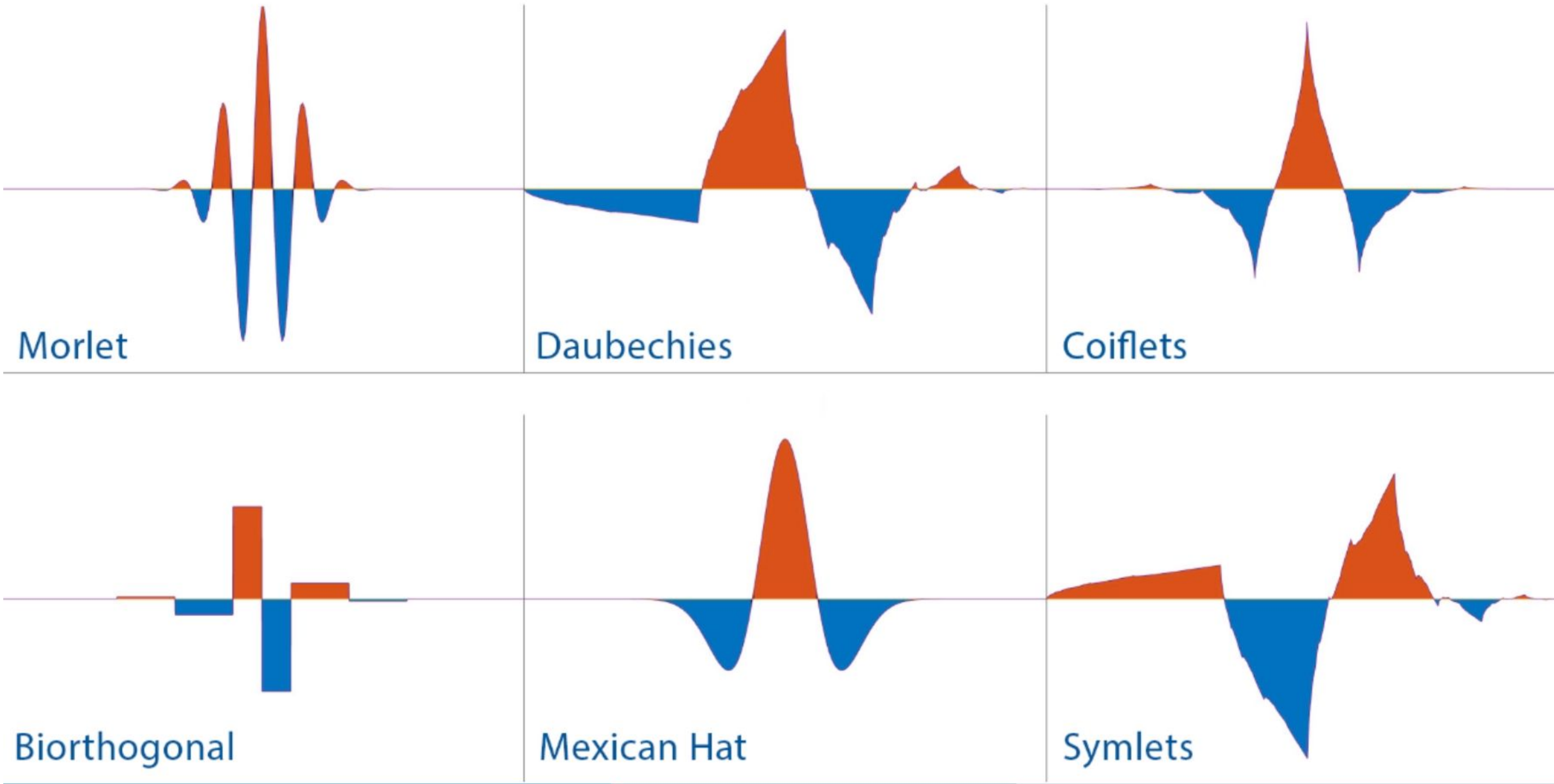
Stationary periodical signal -> Fourier analysis

Nonstationary transient signal -> Wavelet analysis

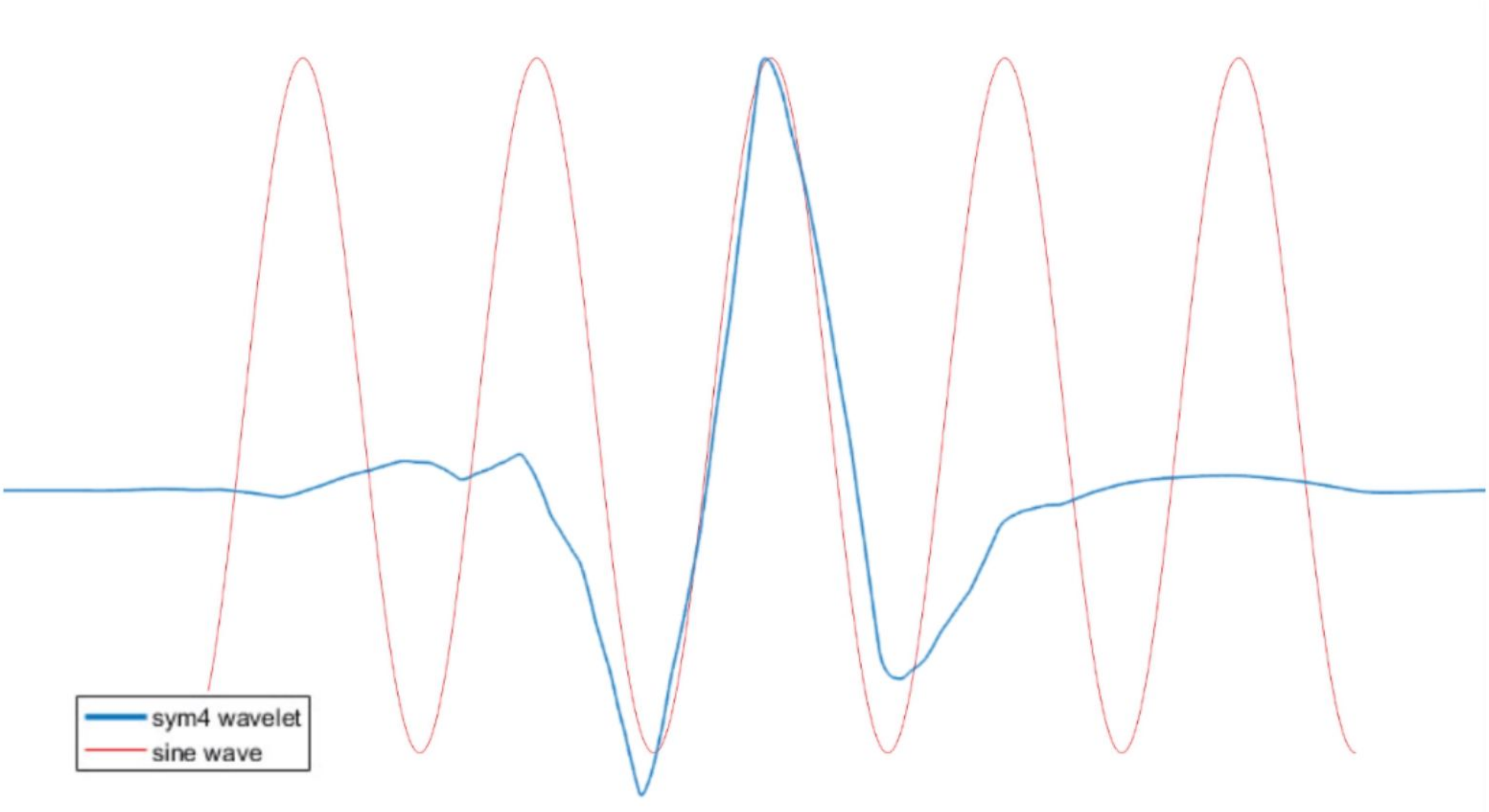
Fourier transform vs. wavelet transform



Different vawelets types



Comparison of a Wavelet with Sine Wave of Same Frequency



Two important wavelet transform concepts

- scaling
- shifting

Scaling

Scaling refers to the process of stretching or shrinking the signal in time, which can be expressed using this equation:

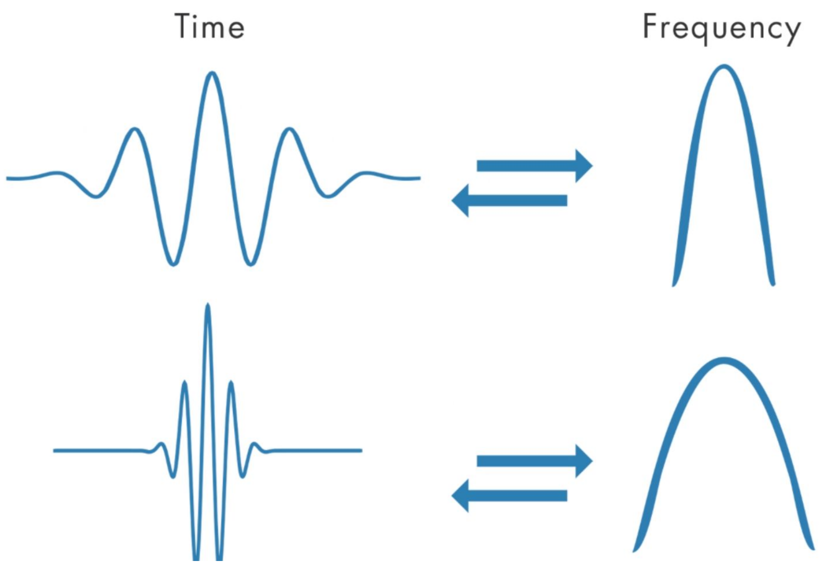
$$\Psi\left(\frac{t}{s}\right) \qquad s>0$$

S is the scaling factor, which is a positive value and corresponds to how much a signal is scaled in time.

scale factor -> frequency

The scale factor is inversely proportional to frequency. For example, scaling a sine wave by 2 results in reducing its original frequency by half .

A larger scale factor results in a stretched wavelet, which corresponds to a lower frequency.



Equivalent frequency

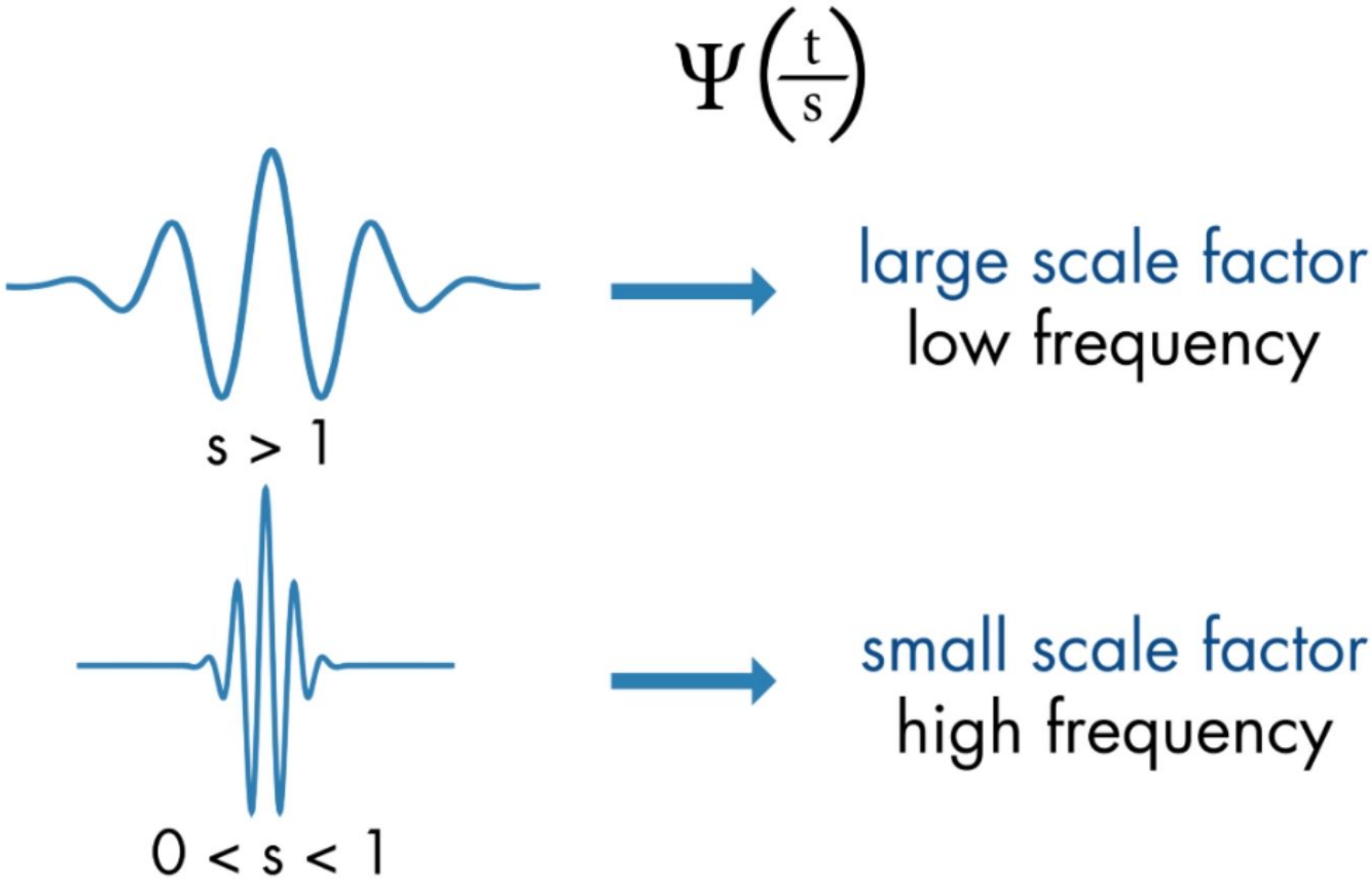
$$F_{eq} = \frac{C_f}{s\delta t}$$

C_f is center frequency of the wavelet,

s is the wavelet scale,

and delta t is the sampling interval.

Scale -> Equivalent frequency



Shifting

Shifting a wavelet simply means delaying or advancing the onset of the wavelet along the length of the signal.

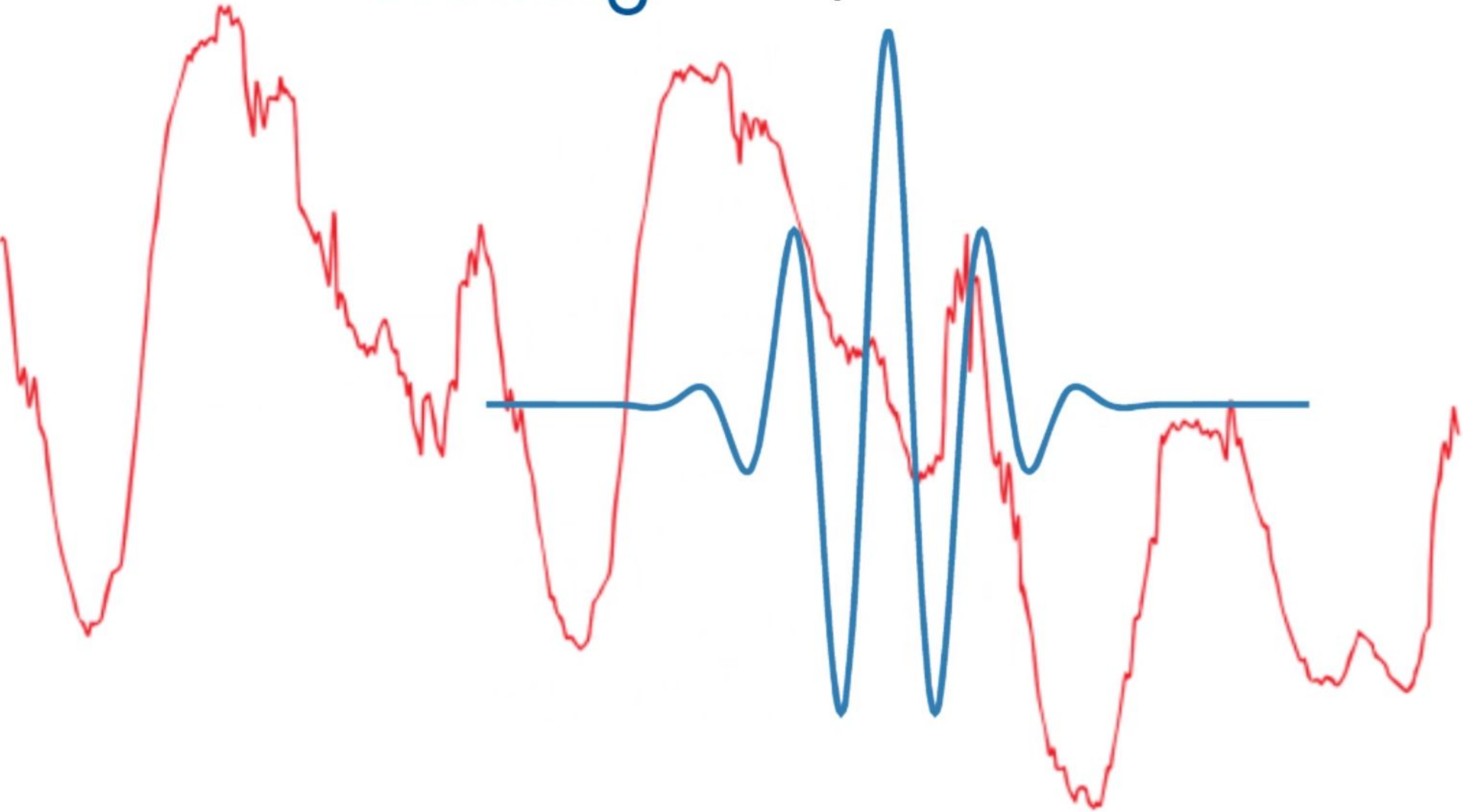
A shifted wavelet represented using this notation

$$\phi(t-k)$$

means that the wavelet is shifted and centered at k.

We need to shift the wavelet to align with the feature we are looking for in a signal.

Shifting $\phi(t-k)$



What is a wavelet transform?

Representation of a function in real space as a linear combination of wavelet basis functions

Two types of wavelet transforms

- Continuous Wavelet Transform

Key applications of the continuous wavelet analysis are:

- time frequency analysis,
- and filtering of time localized frequency components.

- Discrete Wavelet Transform.

The key application for Discrete Wavelet Analysis are:

- denoising and
- compression of signals and images.

these two transforms differ based on how they discretize the scale and the translation parameters.

Continuous Wavelet Transform (CWT)

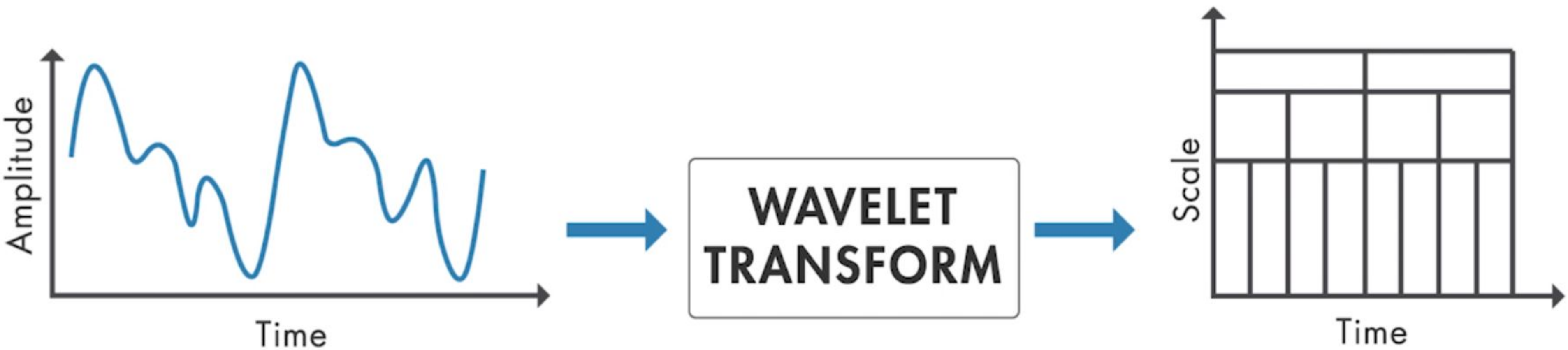
The continuous wavelet transform (CWT) is defined as the sum over all time of the signal multiplied by scaled, shifted versions of the wavelet function :

$$C(scale, position) = \int_{-\infty}^{\infty} f(t)\psi(scale, position, t)dt$$

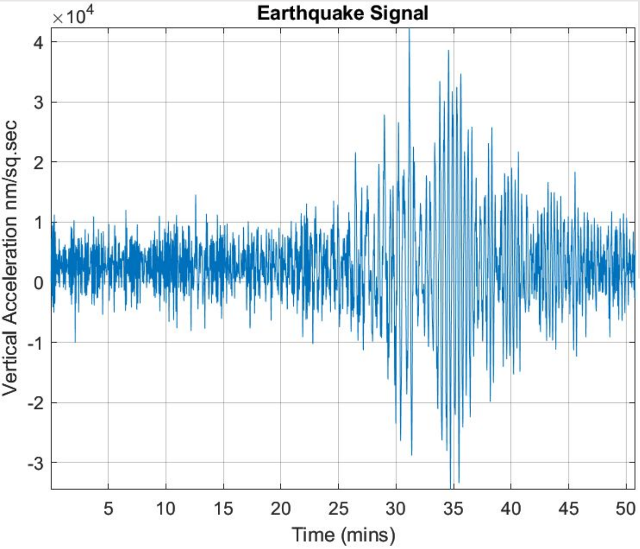
Scale

S>1: dilate the signal

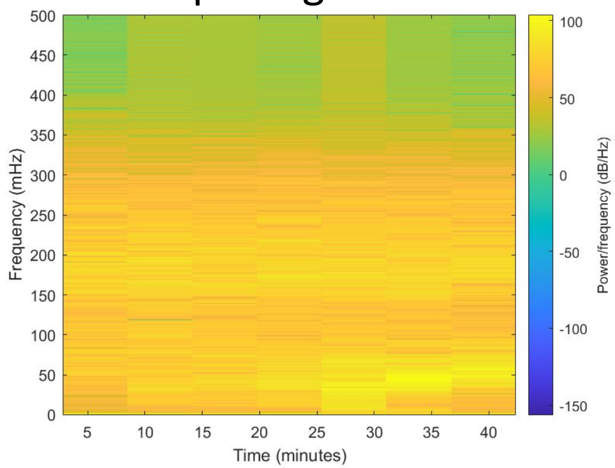
S<1: compress the signal

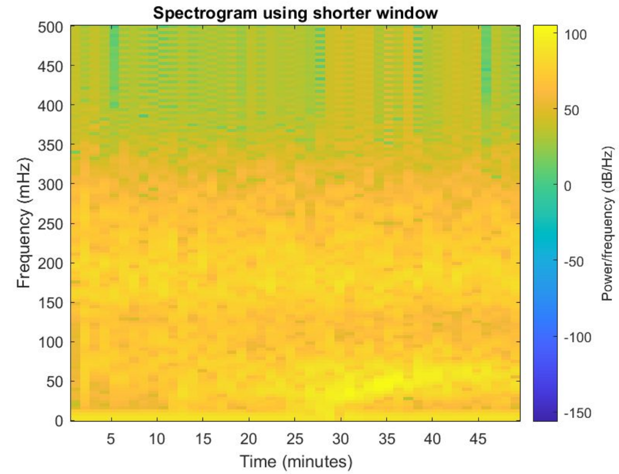

Amplitude
Time
WAVELET
TRANSFORM
Scale
Time

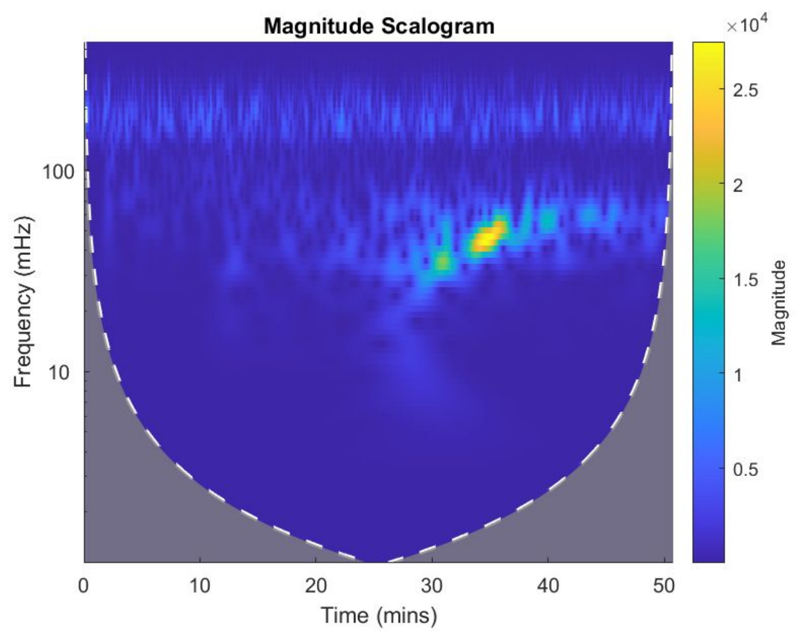
CWT example



Spectogram







Continuous Wavelet Transform (CWT)

Direct transform

$$\left(T^{\text{wave}} f\right)(s,\tau) = |a|^{-1/2} \int f(t)\varphi\left(\frac{t-b}{a}\right)dt$$

Inverse transform

$$f(t) = \iint \left(T^{\text{wave}} f\right)(s,\tau)\varphi\left(\frac{t-b}{a}\right)d\tau\, ds$$

Discrete wavelet transform (DWT)

Discrete wavelet transforms(1D)

- Forward

$$W_\varphi(j_0,k)=\frac{1}{\sqrt{M}}\sum_n f(n)\varphi_{j_0,k}(n)$$

$$W_\psi(j,k)=\frac{1}{\sqrt{M}}\sum_n f(n)\psi_{j,k}(n) for, j\geq j_0$$

- Inverse

$$f(n)=\frac{1}{\sqrt{M}}\sum_k W_\varphi(j_0,k)\varphi_{j_0,k}(n)+\frac{1}{\sqrt{M}}\sum_{j=j_0}^{\infty}\sum_k W_\psi(j,k)\psi_{j,k}(n)$$

Wavelets

Parent wavelets

Father wavelet (φ) or scaling function

Characterizes basic wavelet scale

Mother wavelet (ψ) or wavelet function

Characterizes basic wavelet shape

Daughter wavelets

All other wavelets are called daughter wavelets
- defined in terms of the parent wavelets

$$\psi^{\mu}_{j,i_x,i_y,i_z}(x, y, z)$$

Notation :

μ : directionality of wavelet functions

j : characteristic scale of wavelet

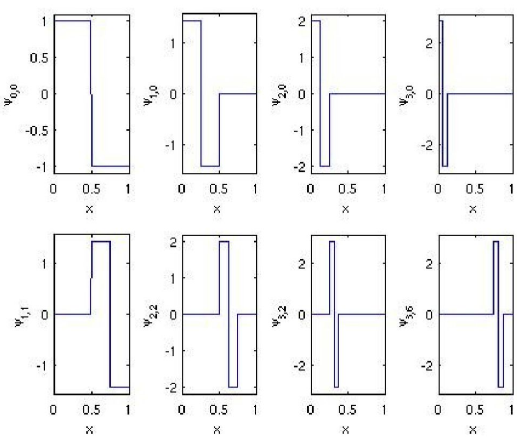
i_s : horizontal and vertical shifts of wavelet functions

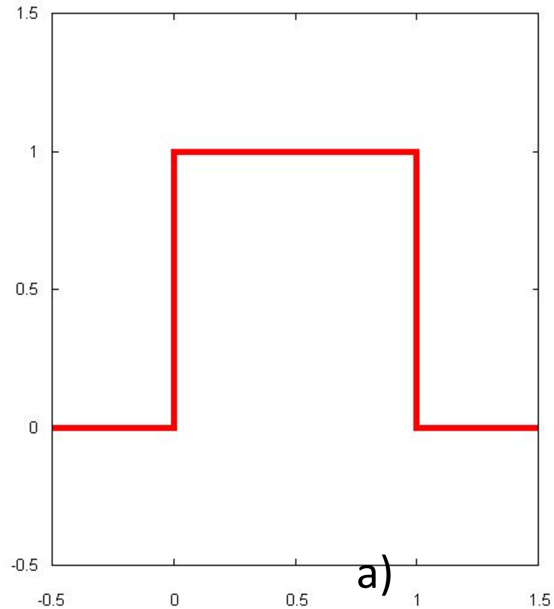
Types Of Wavelets
Haar Wavelets

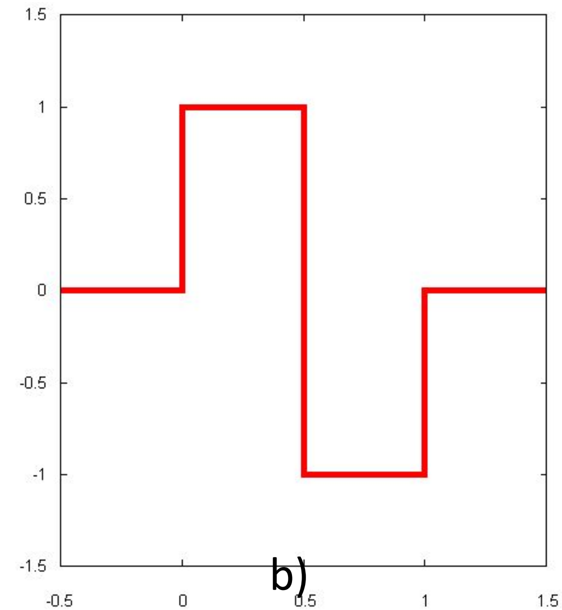
Haar

a)Father -scale function and

b)Mother -Wavelet function



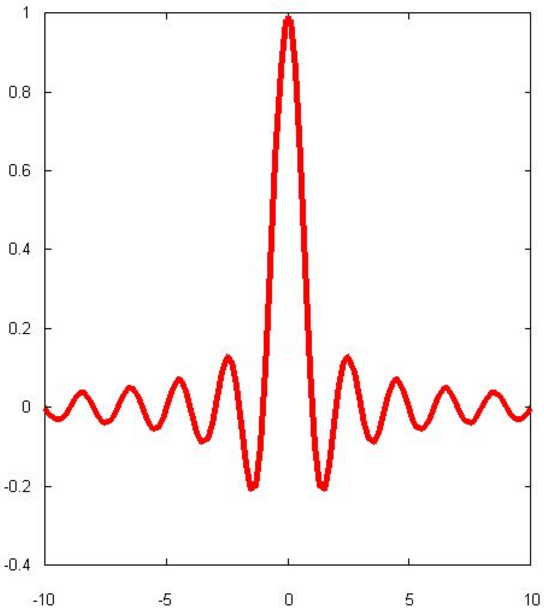




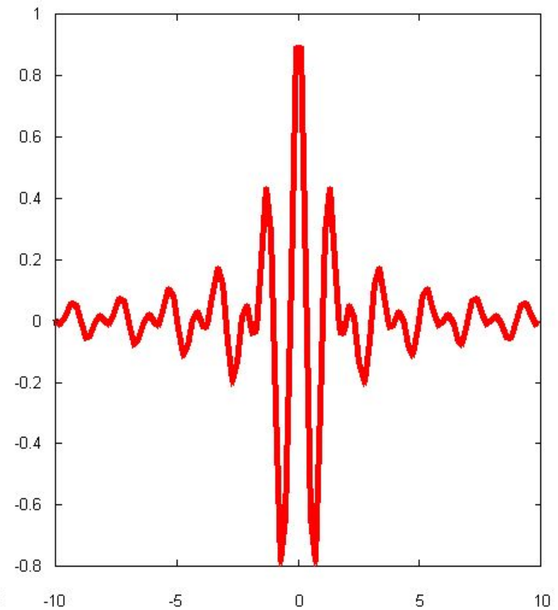
Types Of Wavelets
Shannon Wavelet

Father function $$\varphi(x) = \mathrm{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$$

Wavelet function $$\psi = \frac{\sin(2\pi x) - \sin(\pi x)}{\pi x}$$



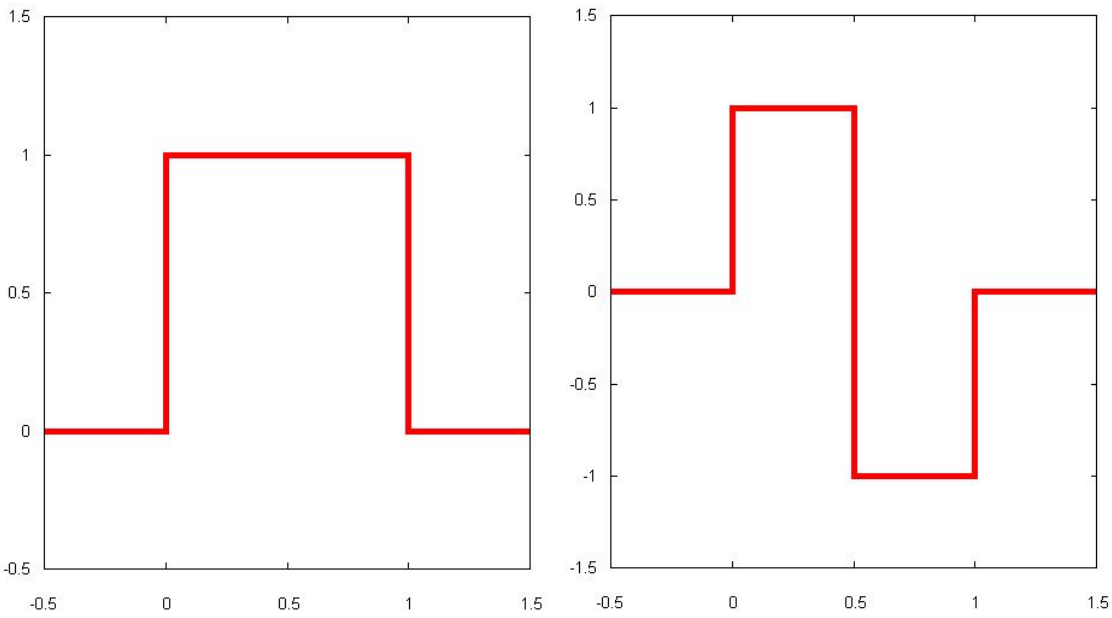
a)Father function



b)Wavelet function

Types Of Wavelets
Daubechies Wavelets

Note: Daubechhies D1 wavelet is Haar Wavelet



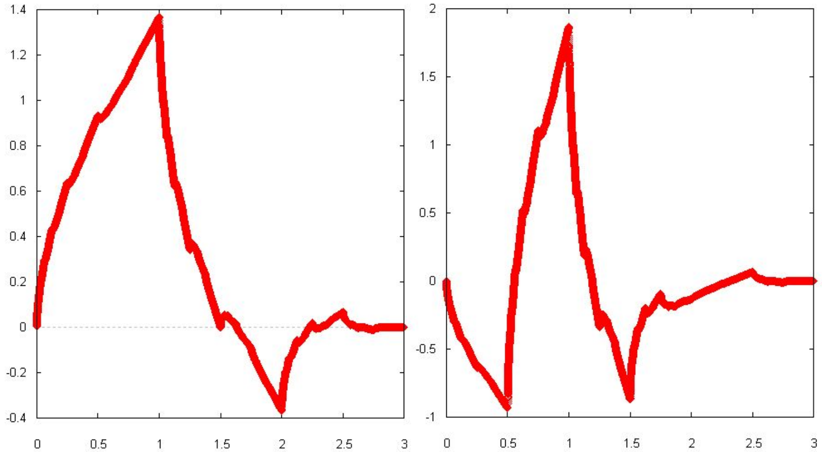
Types Of Wavelets
Daubechies Wavelets

Daubechhies D2

a)Father -scale function and

b)Mother -Wavelet function

The Daubechies wavelets, based on the work of Ingrid Daubechies, are a family of orthogonal wavelets defining a discrete wavelet transform and characterized by a maximal number of vanishing moments for some given support. With each wavelet type of this class, there is a scaling function (called the father wavelet) which generates an orthogonal multiresolution analysis.



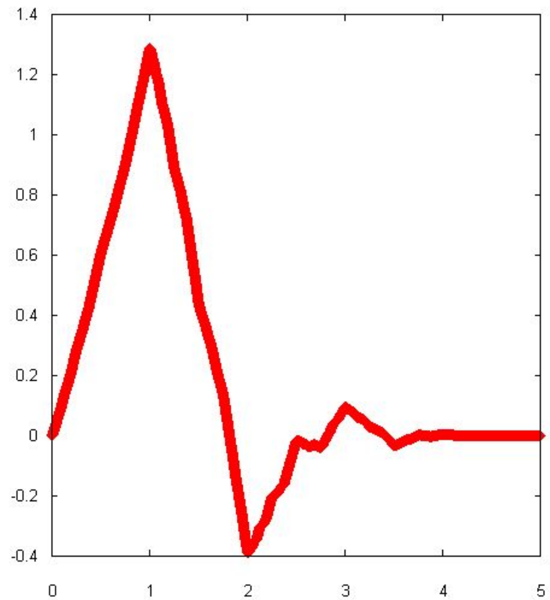
a) b)

Types Of Wavelets
Daubechies Wavelets

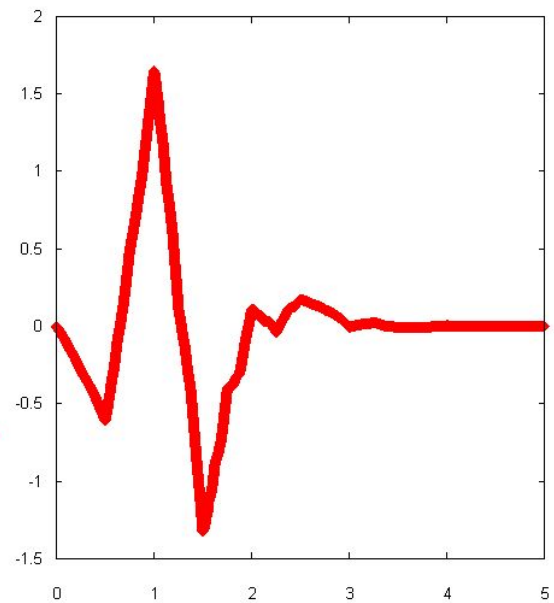
Daubechhies D3

a)Father -scale function and

b)Mother -Wavelet function



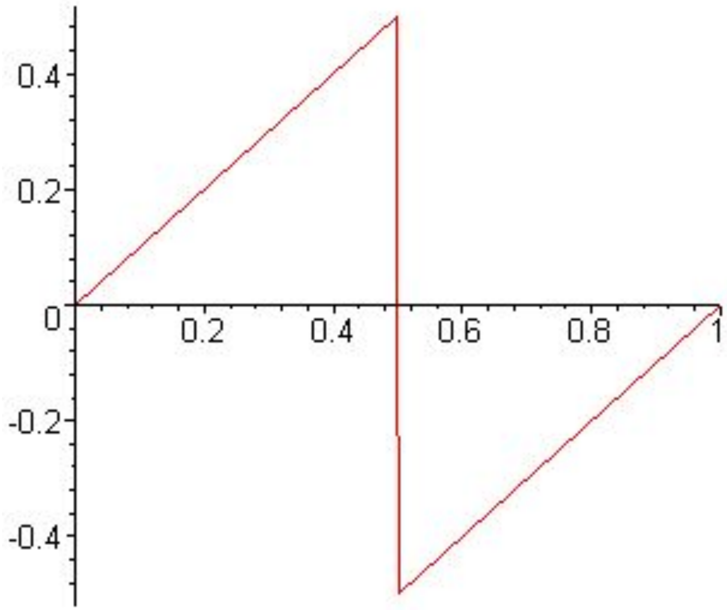
a)



b)

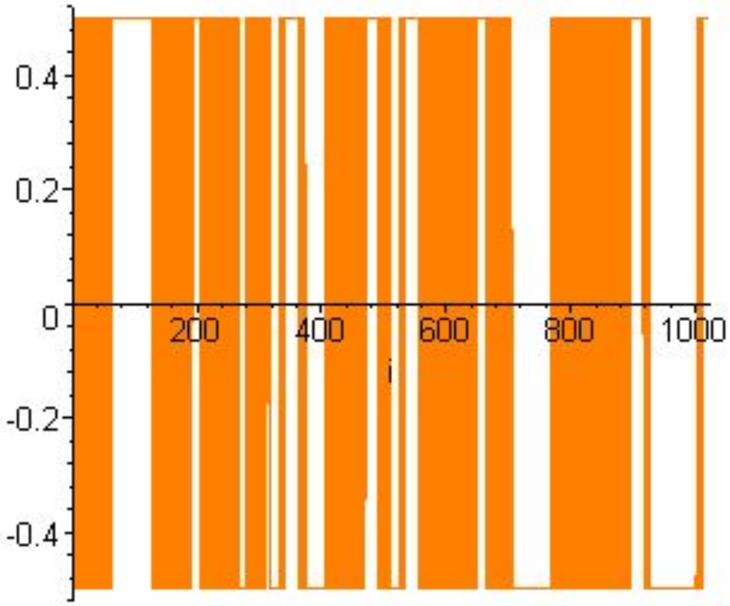
Example - signal

Suppose we have a signal:

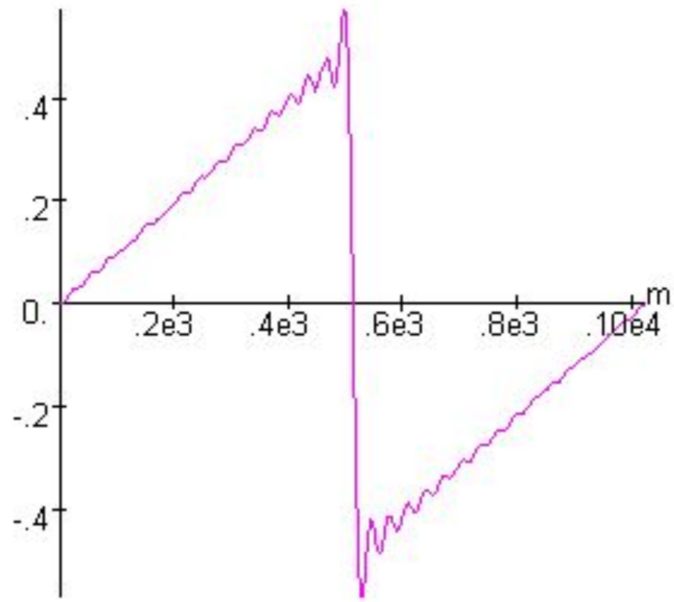


Example – signal Fourier method

Fourier spectrum

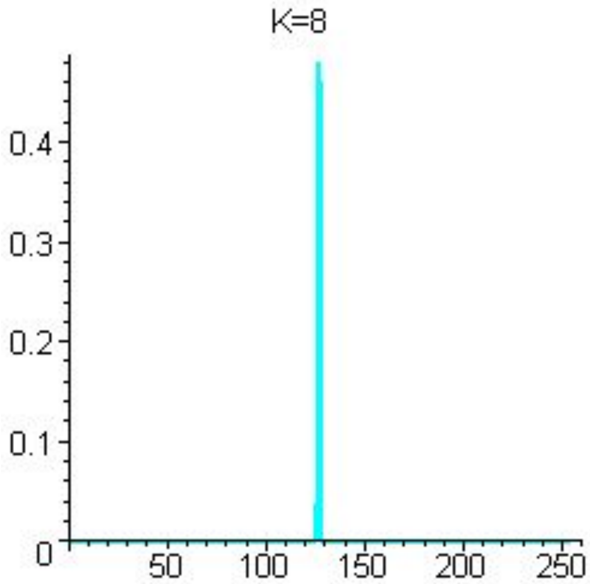


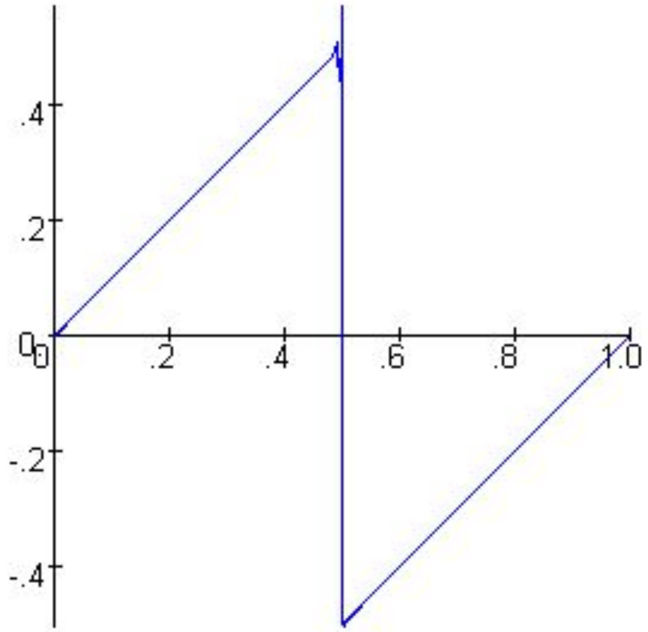
Reconstruction



Example – signal
Wavelet Method

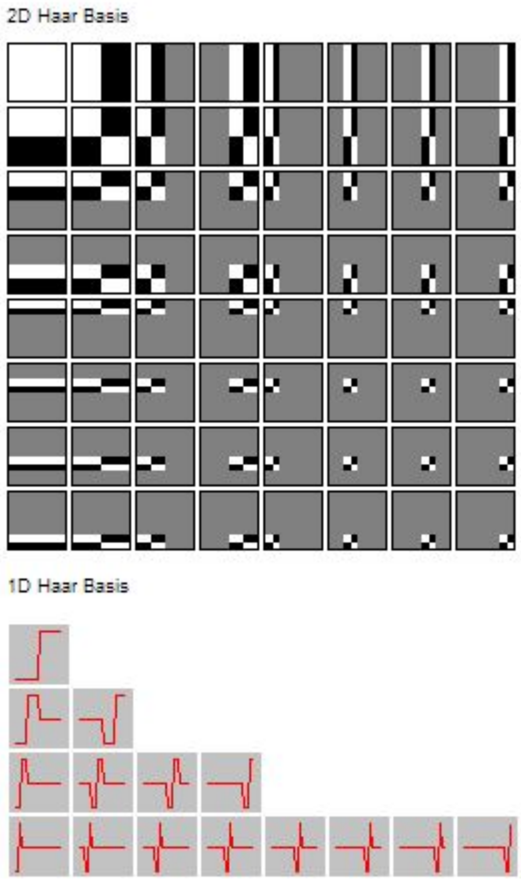
8th Level Coefficients Reconstruction



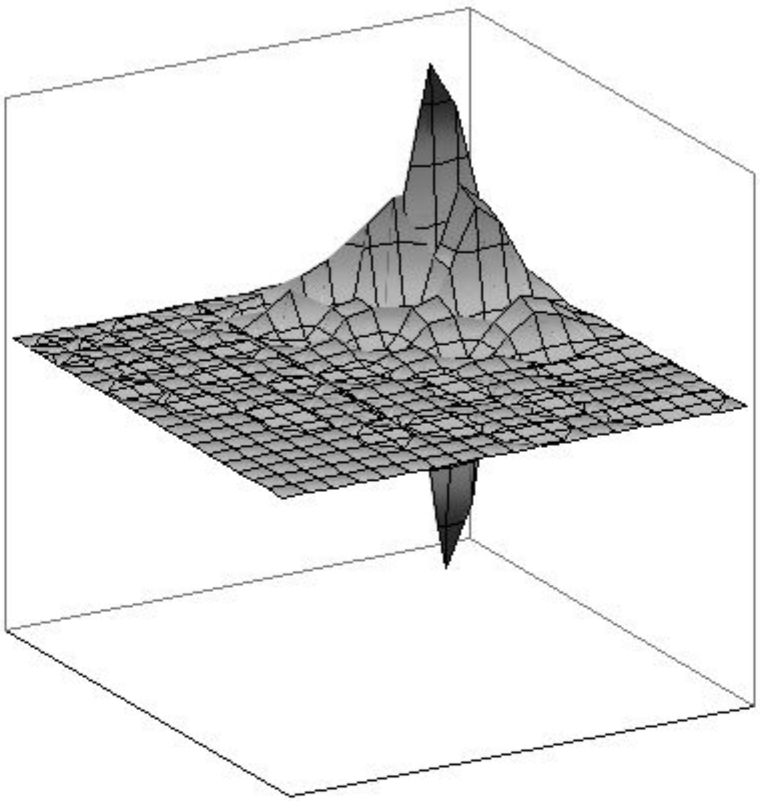


Discrete wavelet transform – examples 2D

Haar wavelet



Daubechies wavelet

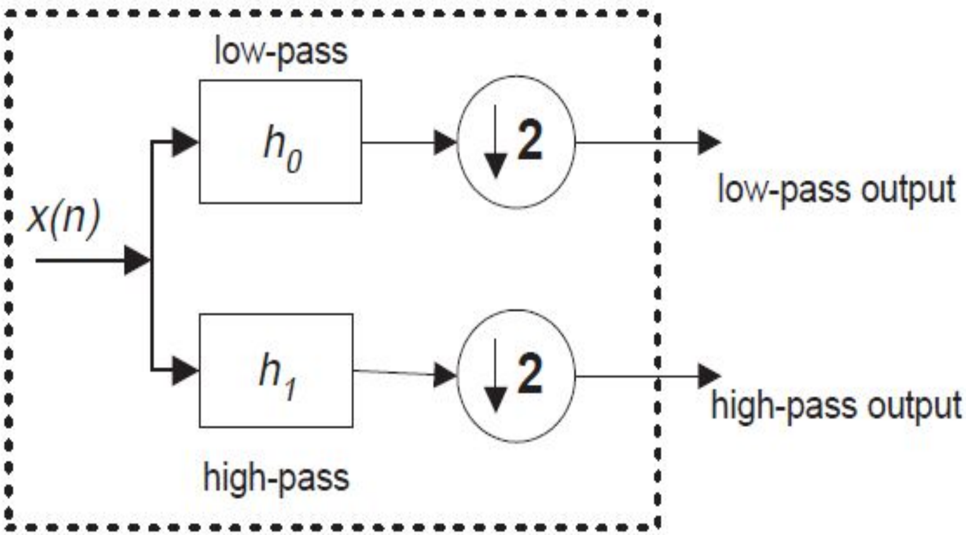


Decomposition

Wavelet decomposition

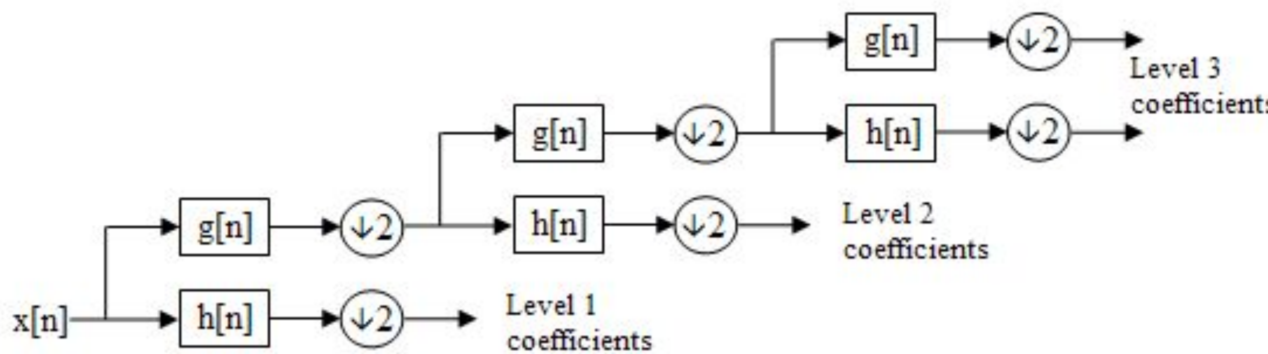
The discrete wavelet transform process is equivalent to comparing a signal with discrete multirate filter banks.

the signal S is first filtered with special lowpass and high pass filter to yield lowpass and highpass sub-bands.

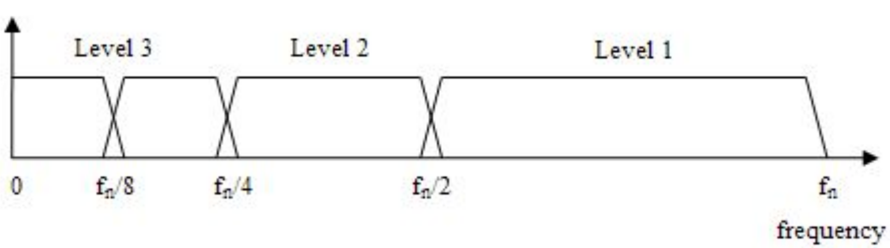


Wavelet decomposition

At each level in the above diagram the signal is decomposed into low and high frequencies.
Due to the decomposition process the input signal must be a multiple of 2^n where n is the number of levels.

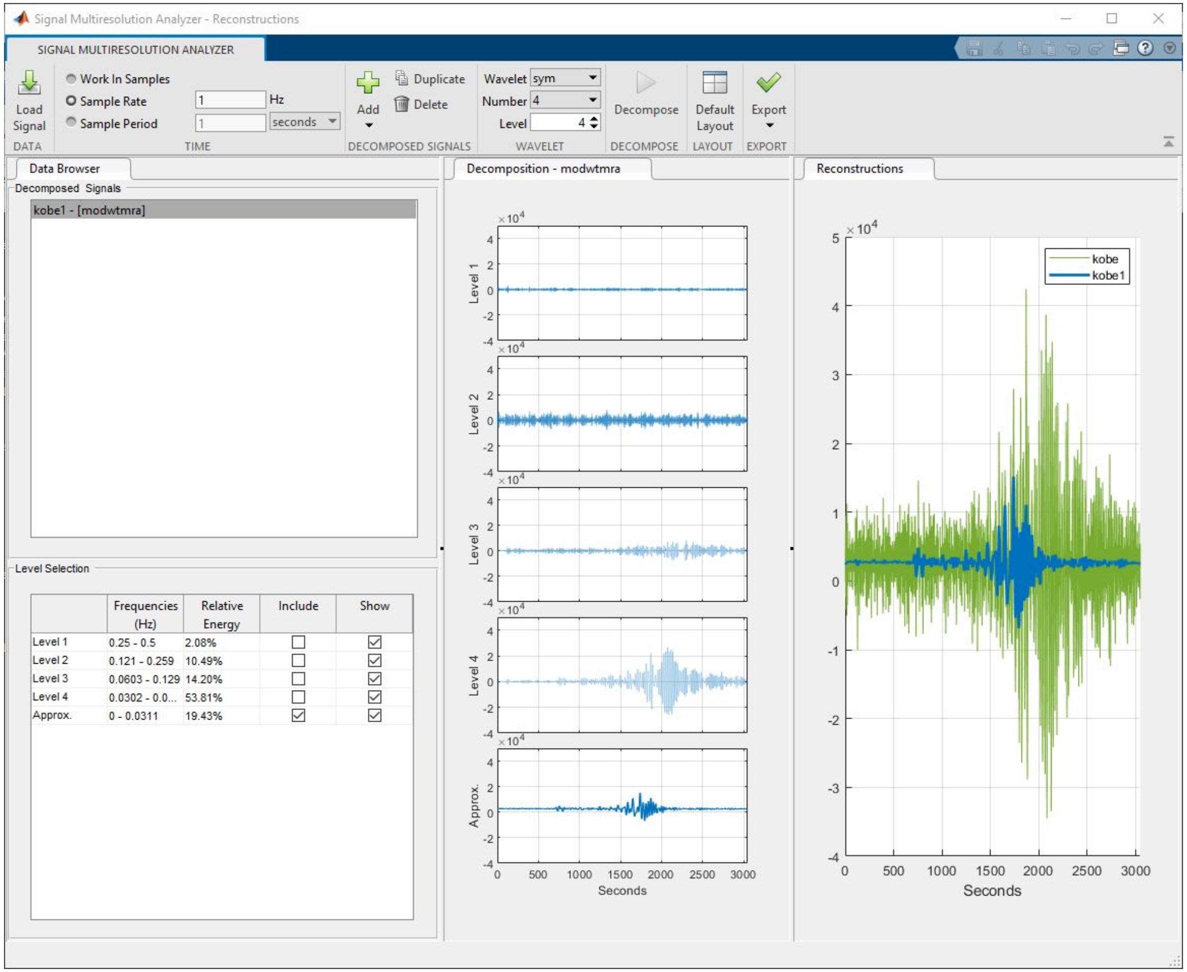


For example, a signal with 32 samples, frequency range 0 to f_n and 3 levels of decomposition, 4 output scales are produced:



Level	Frequencies	Samples
3	$0 to f_n/8$	4
	$f_n/8 to f_n/4$	4
2	$f_n/4 to f_n/2$	8
1	$f_n/2 to f_n$	16

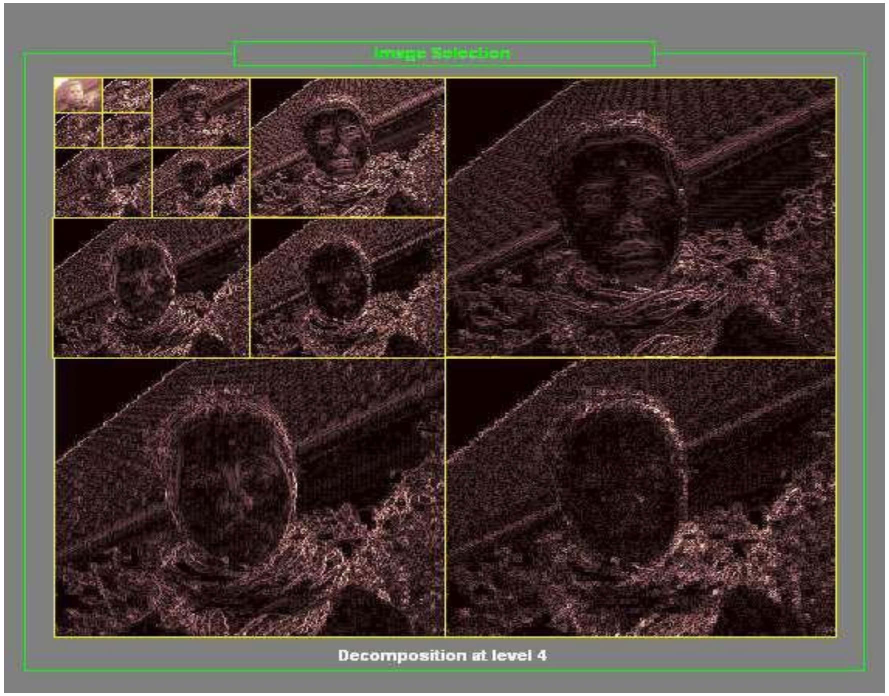
Matlab – Signal Multiresolution Analyser



Wavelet decomposition - image

Wavelet decomposition (haar wavelets)

4 level:



JPEG 2000

JPEG - Why another standard?

Low bit-rate compression: JPEG offers excellent rate-distortion performance in the mid and high bit-rates, but at low bit-rates the subjective distortion becomes unacceptable.

Lossless and lossy compression: Need for standard, which provide lossless and lossy compression in one codestream.

Large images: JPEG doesn't compress images greater then 64x64K without tiling.

JPEG - Why another standard?

Single decompression architecture: JPEG has 44 modes, many of them are application specific and not used by the majority of the JPEG decoders.

Transmission in noisy environments: in JPEG quality suffers dramatically, when bit errors are encountered.

Computer generated images and binary iamges:

JPEG is optimized for natural images and performs badly on computer generated images

JPEG fails to compress binary - binary imagery (text) .

JPEG2000
Standard ISO/IEC 15444

Coding standard for:

different types of still images (gray-level, color,...)

different characteristics (natural, scientific, ...)

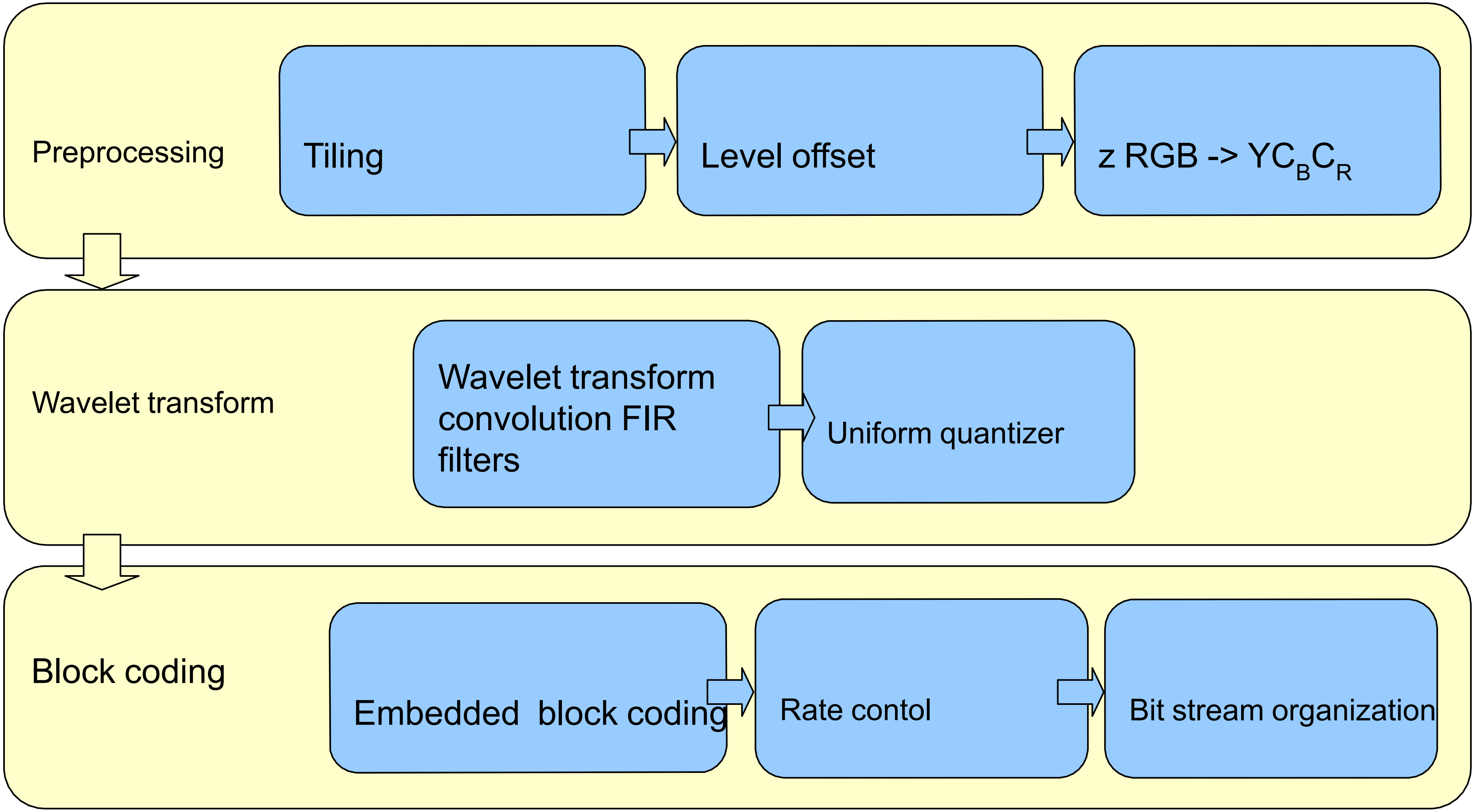
different imaging models (client/server, real-time,.)
within a unified and integrated system.

This coding system is intended for:

low bit-rate applications,

exhibiting rate-distortion and subjective image quality performance superior to existing standards.

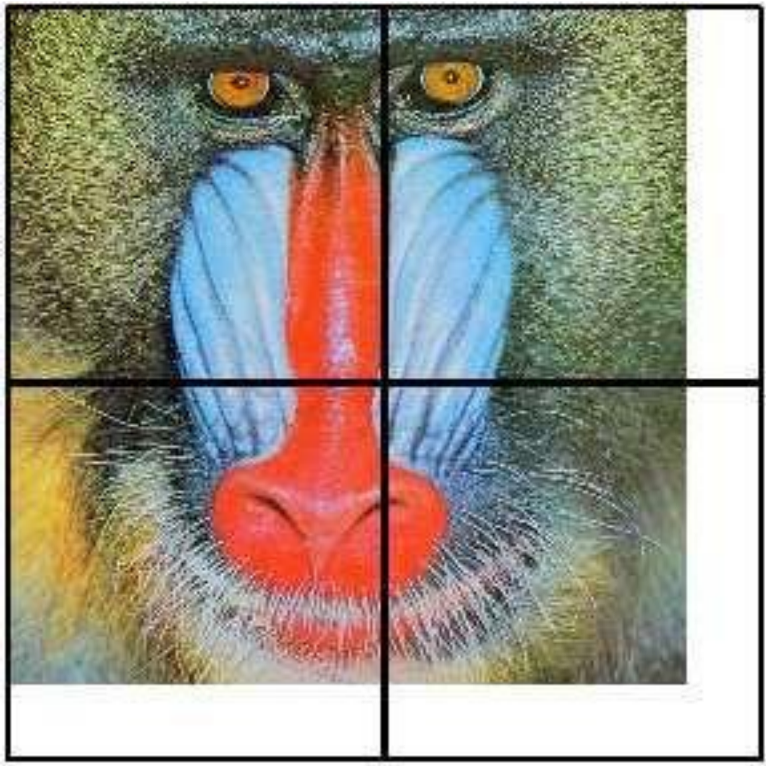
JPEG2000


Preprocessing
Tiling
Level offset
z RGB -> YC_BC_R
Wavelet transform
Wavelet transform convolution FIR filters
Uniform quantizer
Block coding
Embedded block coding
Rate contol
Bit stream organization

Pre-Processing

Image tiling:

- optional
- image is partitioned into rectangular and non-overlapping tiles of equal size



Pre-Processing

1. Image tiling:

Image may be quite large in comparison to the amount of memory available to the codec.

Partition of the original image into rectangular non-overlapping blocks (tiles), to be compressed independently

All operations, including component mixing, wavelet transform, quantization and entropy coding are performed independently on the image tiles.

Tiling affects the image quality both subjectively and objectively → Smaller tiles create more tiling artifacts

Pre-Processing

2. DC-level shifting:

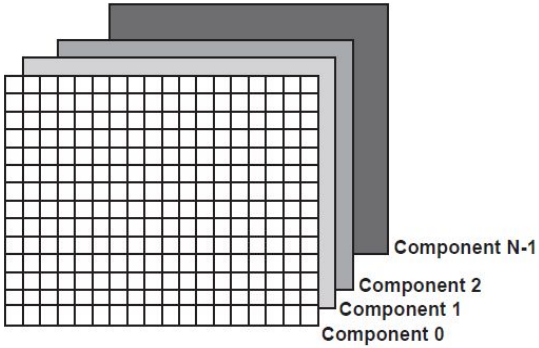
The codec expects its input sample data to have a nominal dynamic range that is approximately

centered about zero (0 – 255 -> -128 to 127)

If the sample values are unsigned, the nominal dynamic range of the samples is adjusted by subtracting from each of the sample values (2^{P-1})

Pre-Processing

3. Color transformation:



The image model that JPEG2000 uses:
an image is composed of one or more
Components - channels (up to 214),

each component consists of a matrix of samples

the sample values are integer valued, can be either signed or unsigned, and can have between 1 and 32 bits/sample

Pre-Processing

3. Color components transformation:

Maps data from RGB to YCrCb

Y, Cr, Cb - less statistically dependent;

reduce the correlation between components, leading to improved coding efficiency.

There are reversible and irreversible transforms.

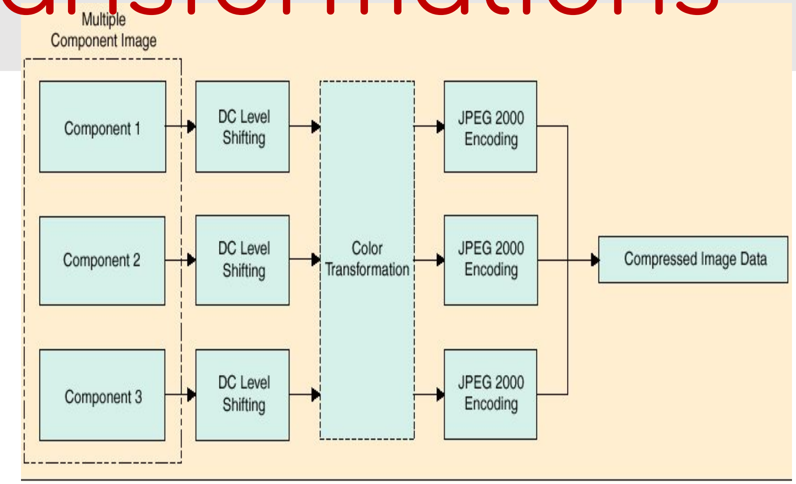
Pre-Processing
Component Color Transformations

Component transformations

Irreversible component transformation (ICT):

Floating point

For use with irreversible (floating point) wavelet



The JPEG 2000 multiple component encoder. Color transformation is optional. If employed, it can be irreversible or reversible.

$$\begin{pmatrix} Y \\ C_b \\ C_r \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.16875 & -0.33126 & 0.5 \\ 0.5 & -0.41869 & -0.08131 \end{pmatrix} \cdot \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

Reversible component transformation (RCT) :

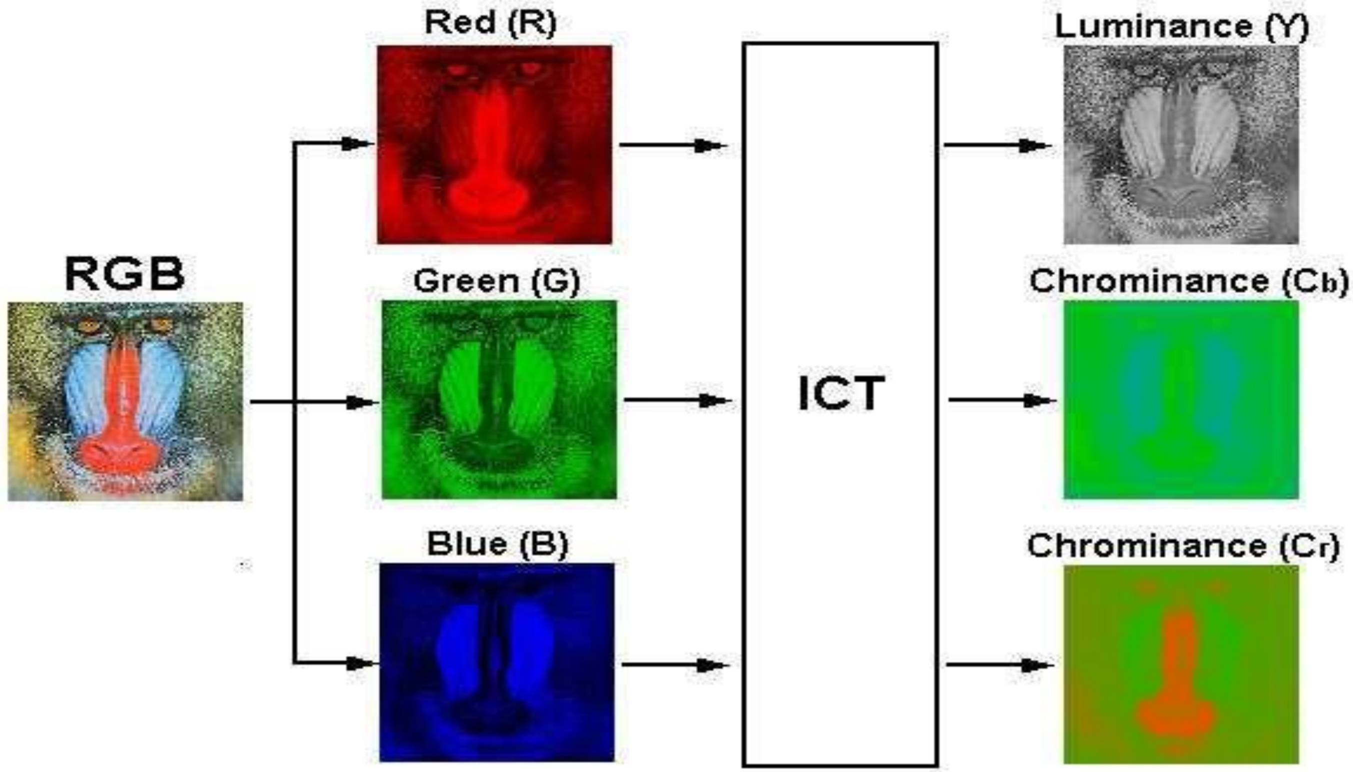
Integer approximation

For use with reversible (integer) wavelet

$$\begin{pmatrix} Y_r \\ U_r \\ V_r \end{pmatrix} = \begin{pmatrix} \left\lfloor \dfrac{R+2G+B}{4} \right\rfloor \\ R-G \\ B-G \end{pmatrix}$$

Pre-Processing
Component Color Transformations

$$\begin{bmatrix} Y \\ C_r \\ C_b \end{bmatrix} = \begin{bmatrix} 0.299 & 0.586 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$



Wavelet Transform

Floating point wavelet filter for lossy compression

Best performance at low bit rate

High implementation complexity, especially for hardware

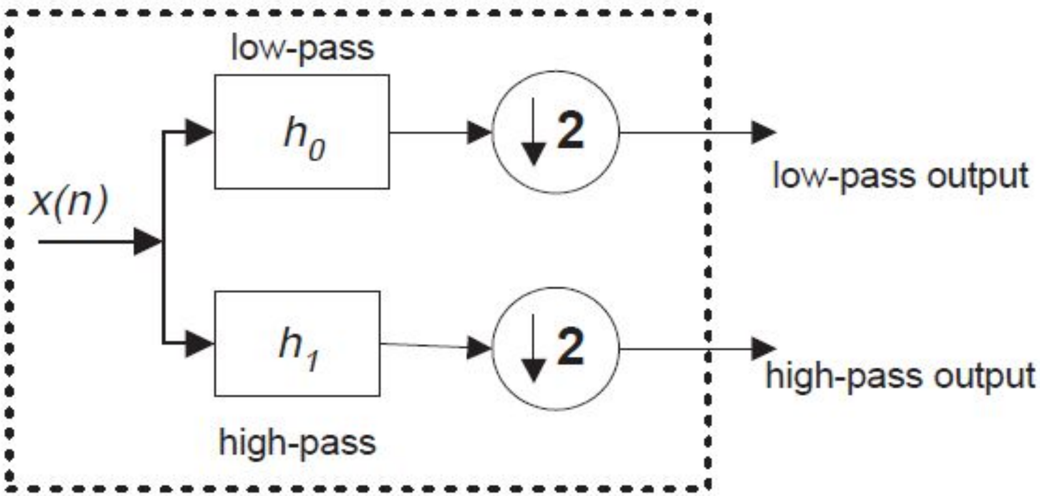
Integer wavelet filter for lossless coding

Integer arithmetic, low implementation complexity

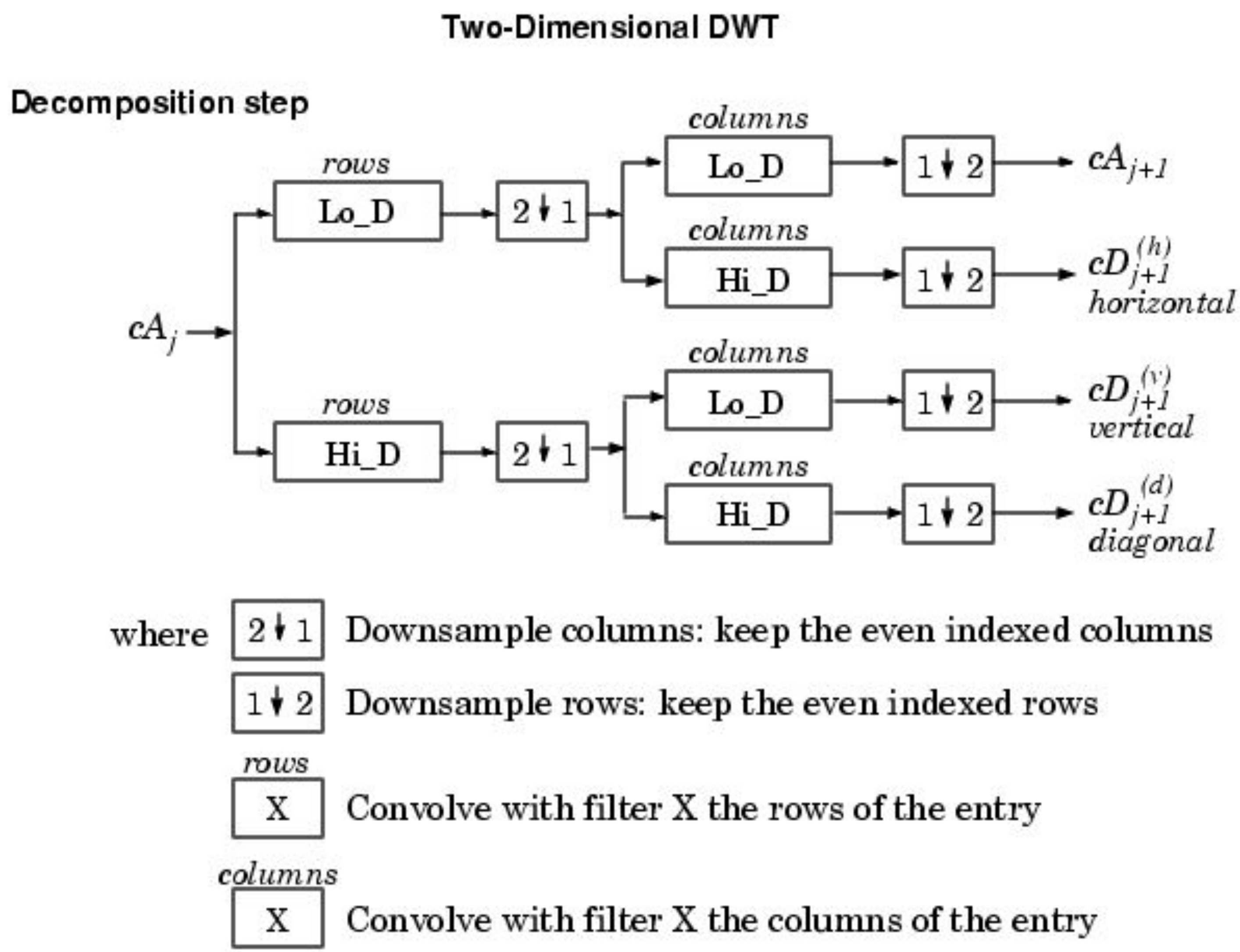
Wavelet decomposition

JPEG2000 uses a discrete wavelet decomposition (DWT) to decompose each image tile into its high and low subbands.

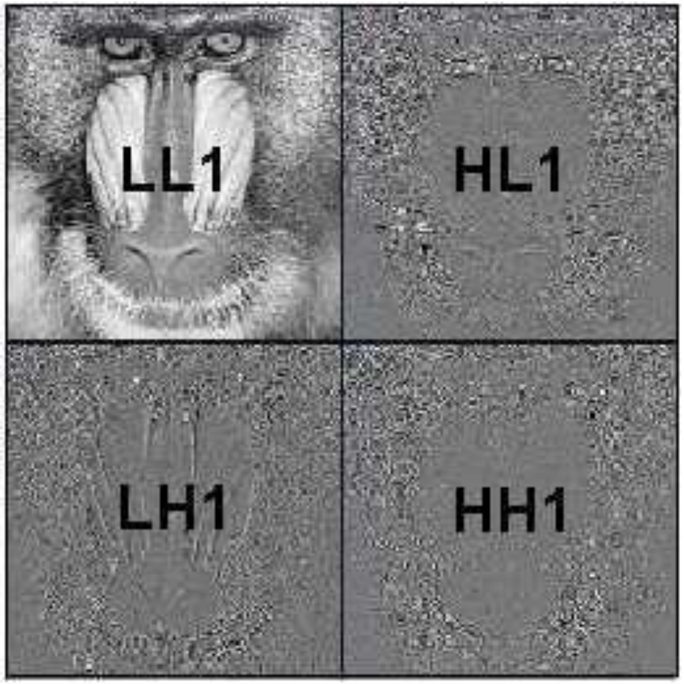
The DWT is performed by filtering each row and column of the pre-processed image tile with a high- pass and low-pass filter.



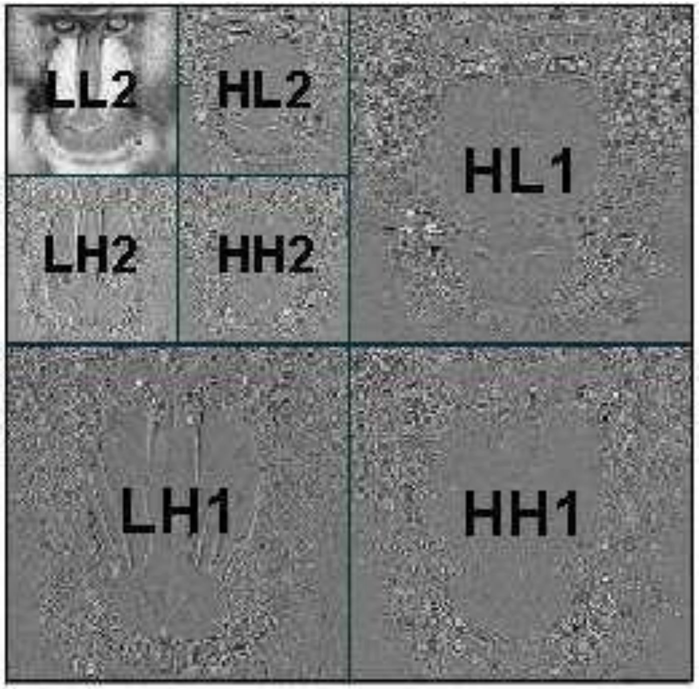
Wavelet decomposition



Wavelet decomposition

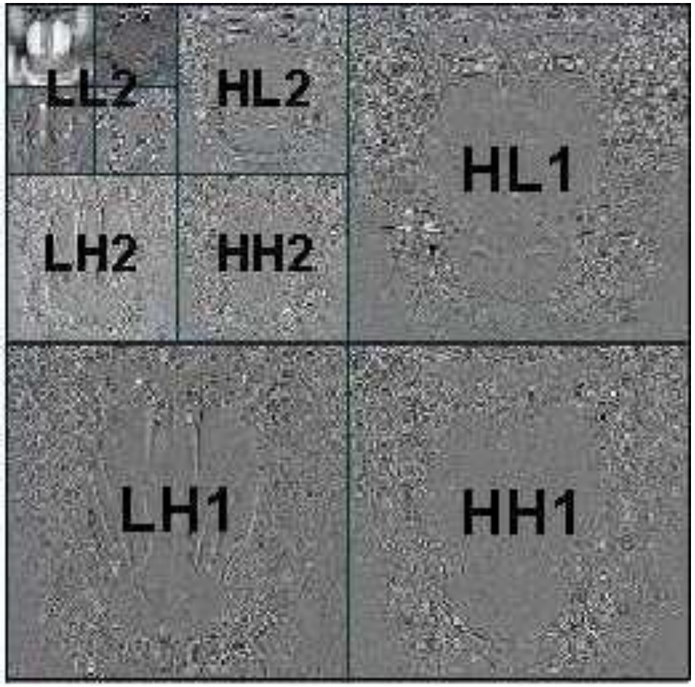


Stage 1 DWT of the 8-bit baboon image tile.



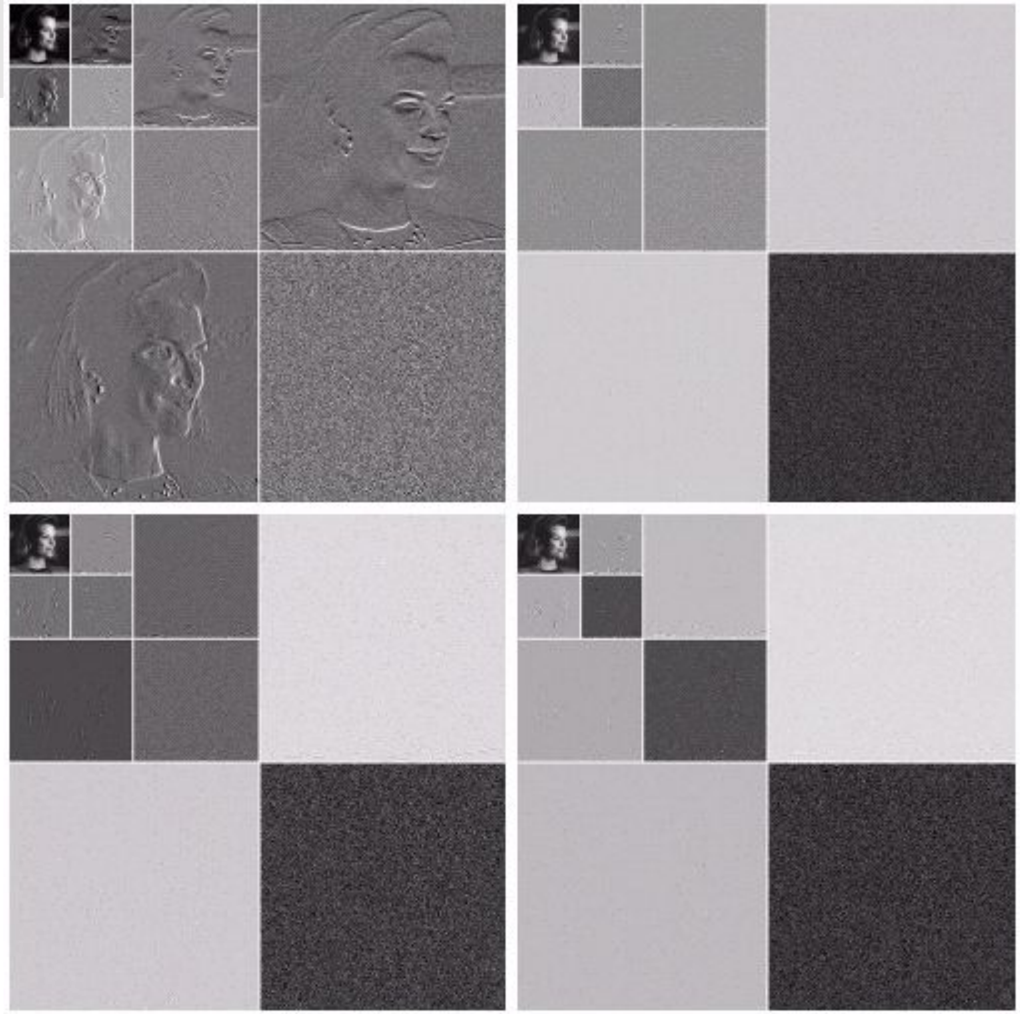
Stage 2 DWT of the 8-bit baboon image tile.

Wavelet decomposition



Stage 3 DWT of the 8-bit baboon image tile.

Comparison of Different Filters



a b
c d

FIGURE 8.42 Wavelet transforms of Fig. 8.23 with respect to (a) Haar wavelets, (b) Daubechies wavelets, (c) symlets, and (d) Cohen-Daubechies-Feauveau biorthogonal wavelets.

DWT decomposition

In JPEG2000 multiple stages (levels) of the DWT are performed.

JPEG2000 supports from 0 to 32 stages.

For natural images, usually between 4 to 8 stages are used.

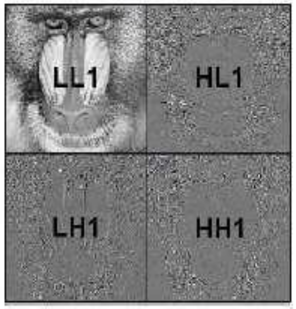


Figure 10: Stage 1 DWT of the 8-bit baboon image tile.

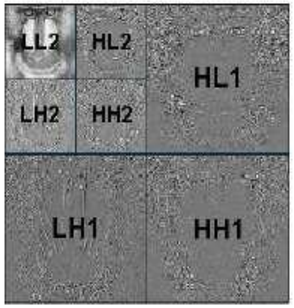


Figure 11: Stage 2 DWT of the 8-bit baboon image tile.

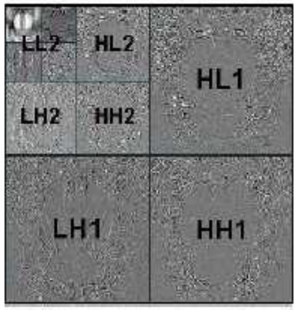


Figure 12: Stage 3 DWT of the 8-bit baboon image tile.

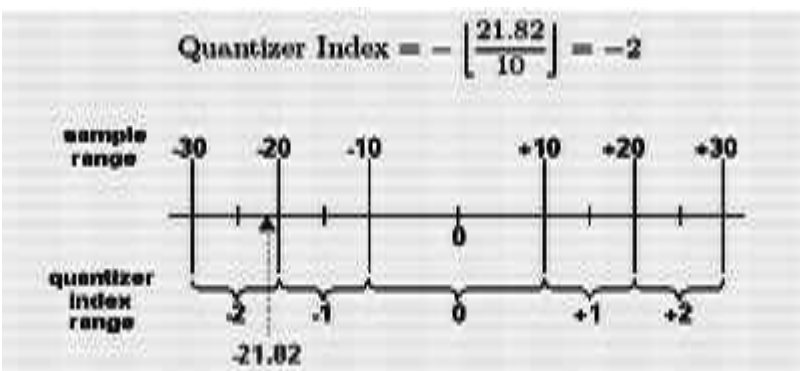
Quantization

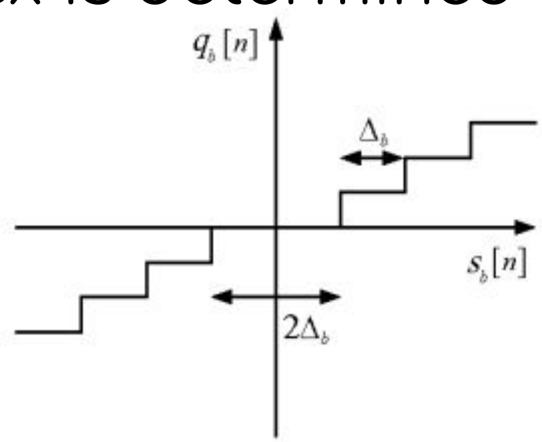
The wavelet coefficients are quantized using a uniform quantizer with deadzone. For each subband b, a basic quantizer step size Δb is used to quantize all the coefficients in that subband according to:

$$q = \mathrm{sign}(y)\left\lfloor \frac{|y|}{\Delta_b} \right\rfloor$$

Example:

Given a quantizer step of 10 and an encoder input value of 21.82, the quantizer index is determined as shown:





Coefficient Bit Modeling – Code blocks.

Wavelet coefficients are associated with different sub-bands arising from the 2D separable transform applied.

These coefficients are then arranged into rectangular blocks within each sub-band, called code blocks.

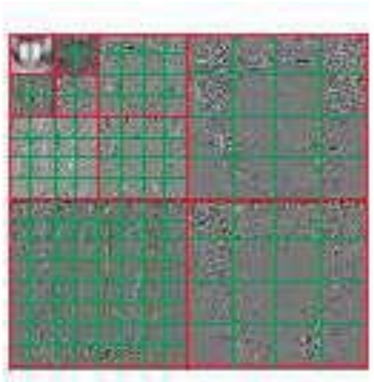


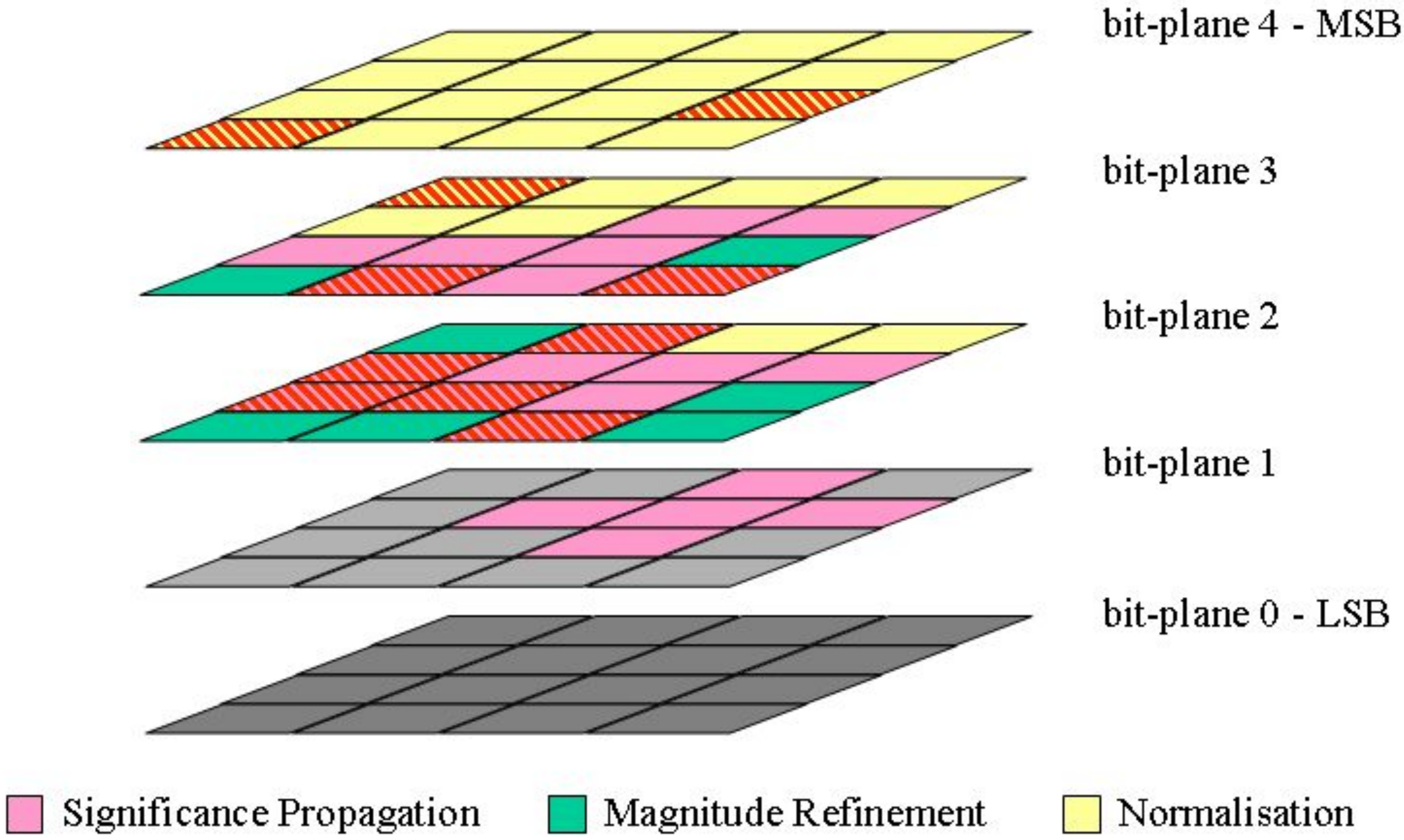
Figure 15: Example division of subbands into code-blocks.

Rate Control

Rate control is the process by which the code-stream is altered so that a target bit rate can be reached.

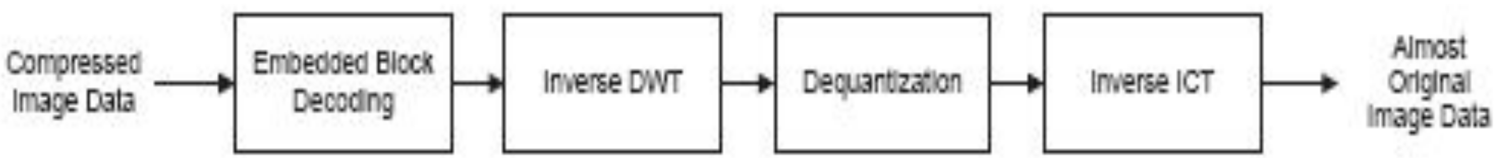
Once the entire image has been compressed, a post-processing operation passes over all the compressed blocks and determines the extent to which each block's embedded bit stream should be truncated in order to achieve the target bit rate.

Bit planes coded separatly



Decoding - The decoder basically performs the opposite of the encoder:

The code-stream is received by the decoder according to the progression order stated in the header. The coefficients in the packets are then decoded and dequantized, and the reverse-ICT is performed:



In the case of irreversible compression, the decompression results in loss of data. The resulting image is not exactly like the original

.

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.0 & 0.0 & 1.4021 \\ 1.0 & -0.3441 & -0.7142 \\ 1.0 & 1.7718 & 0.0 \end{bmatrix} \begin{bmatrix} Y \\ C_r \\ C_b \end{bmatrix}$$

JPEG vs. JPEG 2000

JPEG vs. JPEG 2000

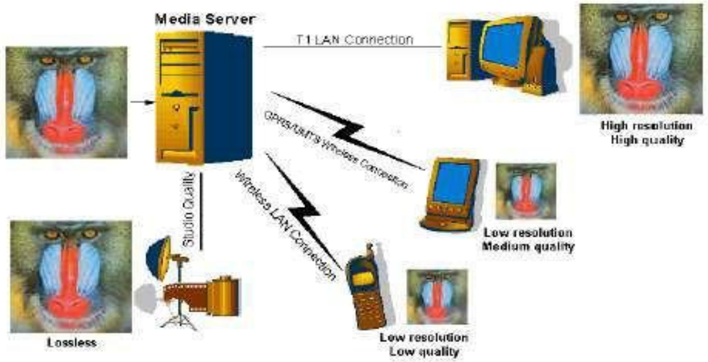
1. Compress once - decompress many ways
2. Region-Of-Interest encoding
3. Progression is native
4. Error resilience (resistance)
5. Quality of synthetized images

Compress once, decompress many ways

In JPEG2000, the compressor decides the maximum resolution and maximum image quality to be used.

It is also possible to perform random access by decompressing only a certain region (ROI) of the image or a specific component of the image (e.g. the grayscale component of a color image). Both can be performed with varying qualities and resolutions.

In each case it is possible to locate, extract, and decode the bytes required for the desired image product without decoding the entire code-stream.

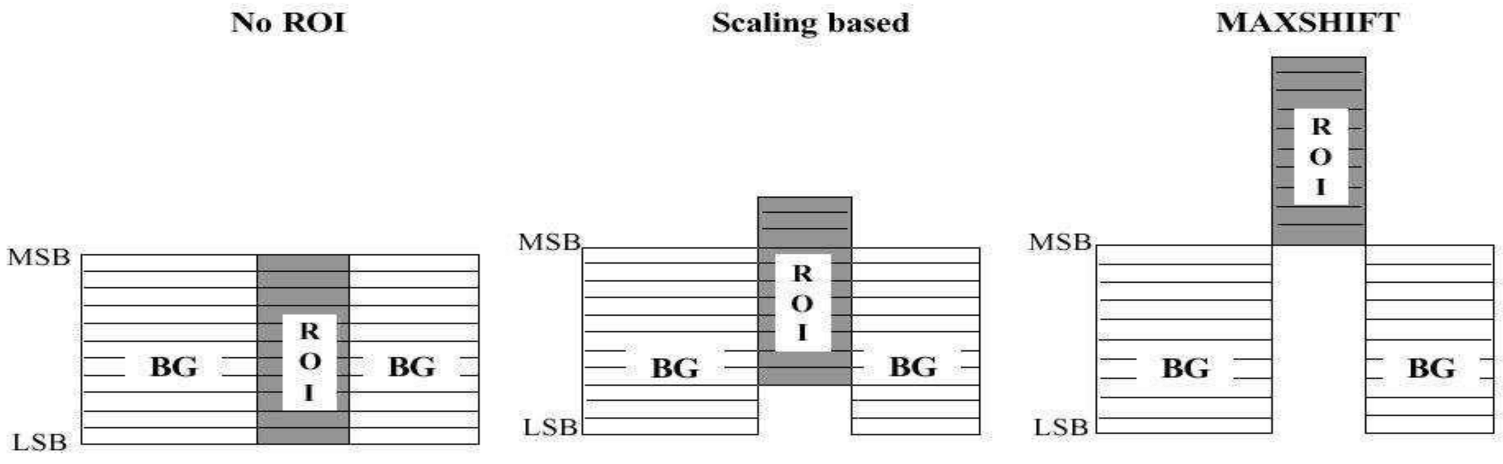


Region-of-interest (ROI)

A ROI is a part of an image that is encoded with higher quality than the rest of the image (the background).

The encoding is done in such a way that the information associated with the ROI precedes the information associated with the background.

Two methods: Scaling based and Maxshift



Scaling of ROI coefficients

ROI - example

Original Image

with ROI Defined

Decoded Image

with ROI Intact

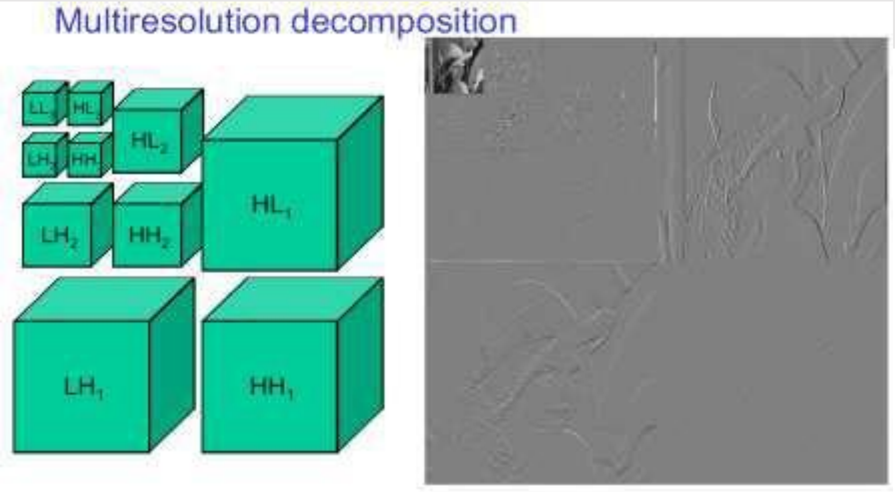
Scalability and bit-stream parsing

2 important modes of scalability:

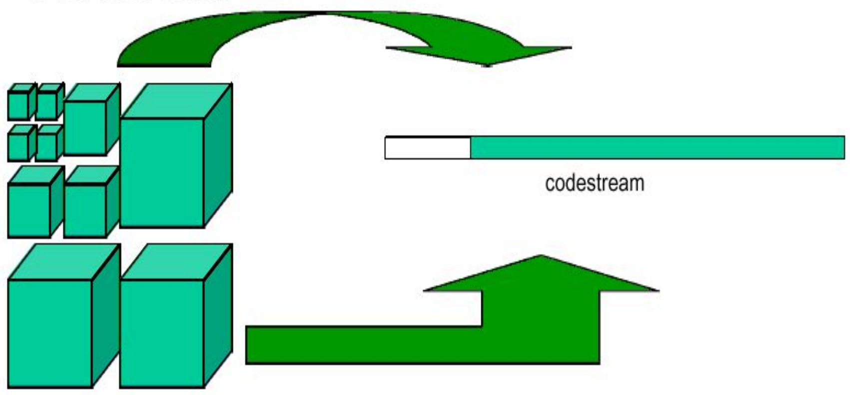
1. Resolution/Spatial Quality (SNR)
2. Bit-stream parsing

A combination of spatial and quality scalability.

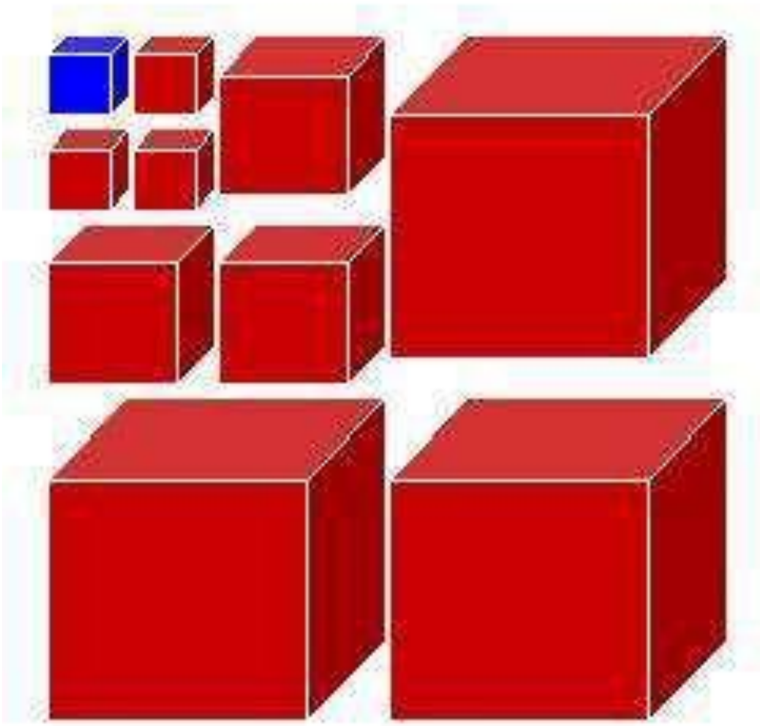
It is possible to progress by spatial scalability to a given (resolution) level and then change the progression by SNR at a higher level.



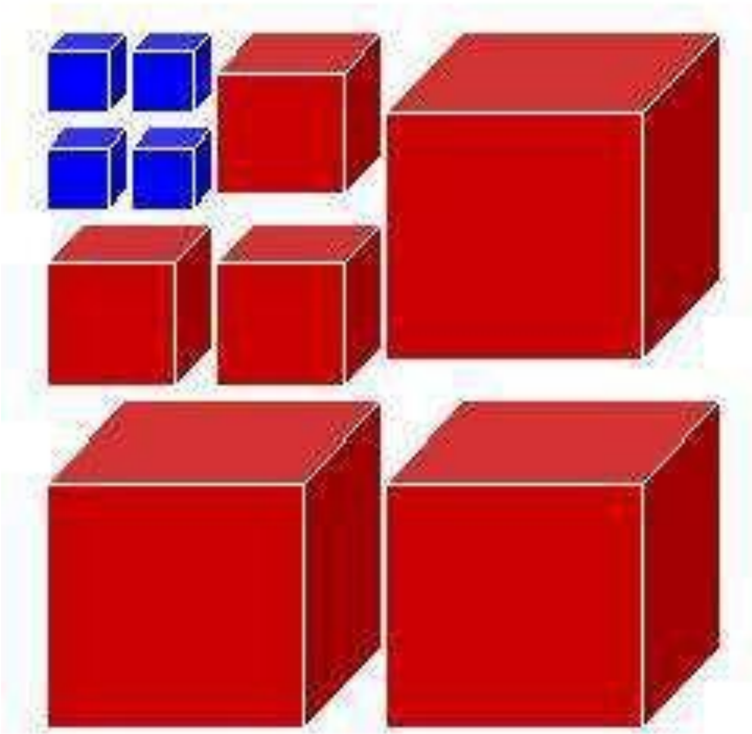
– Different modes are realized depending on the way information is written into the codestream



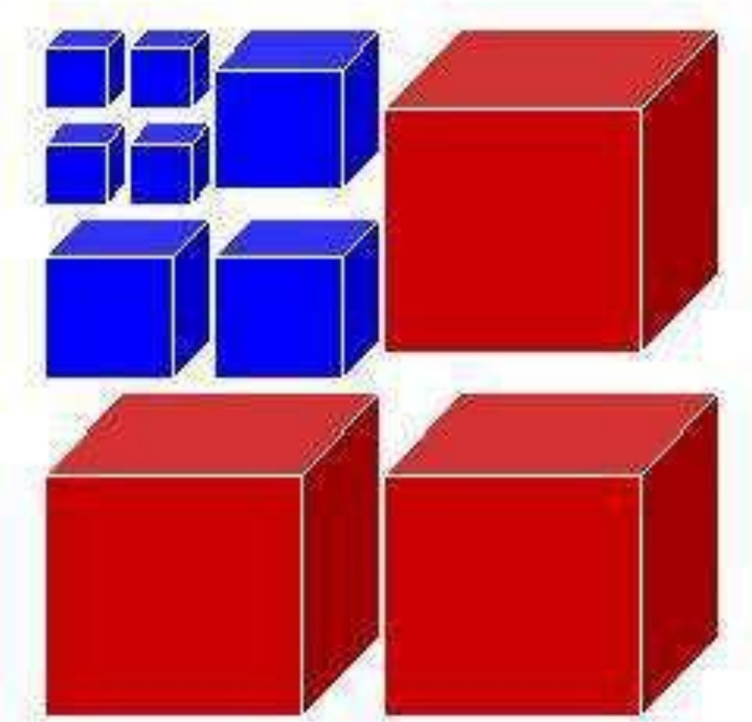
Resolution scalability



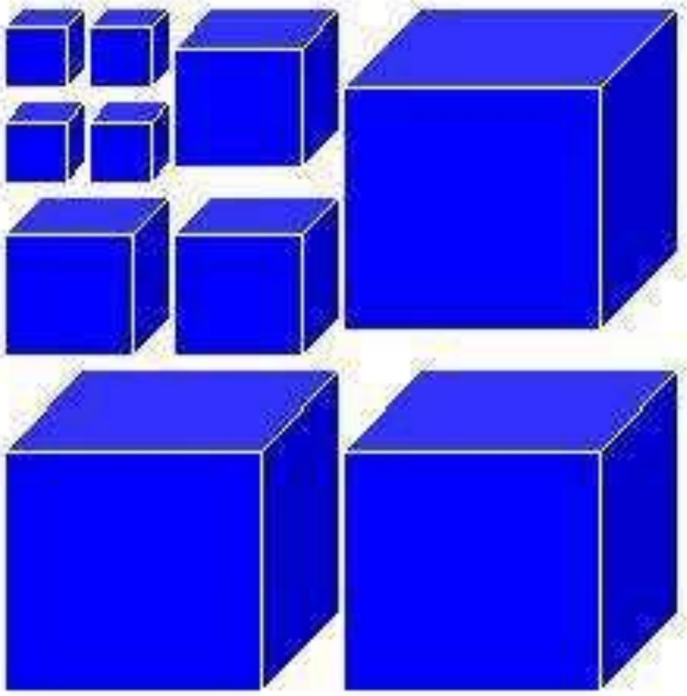
Resolution scalability



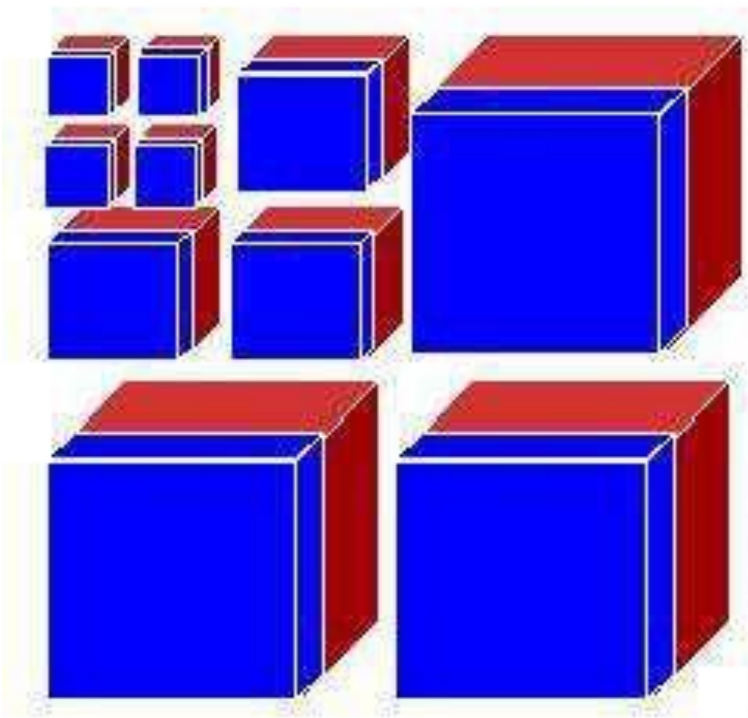
Resolution scalability



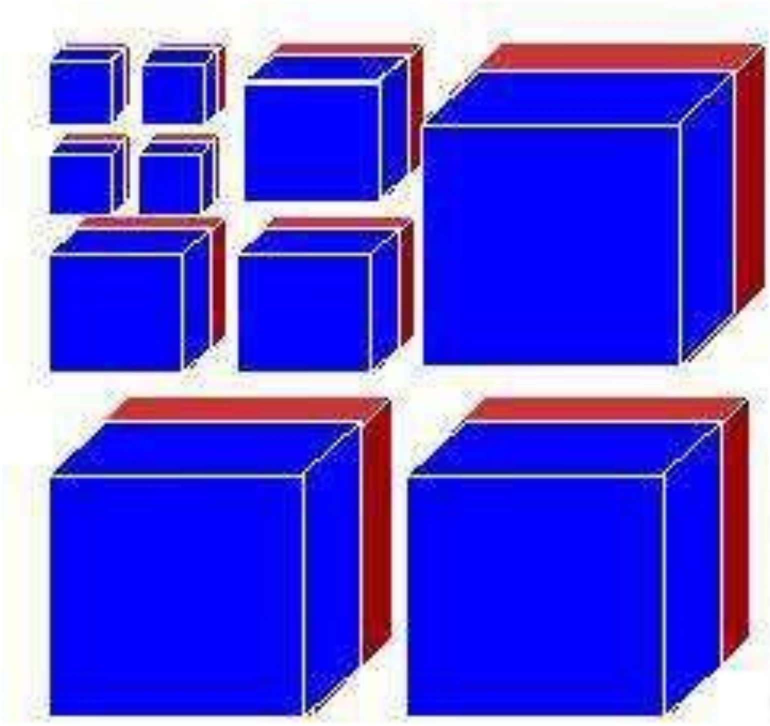
Resolution scalability



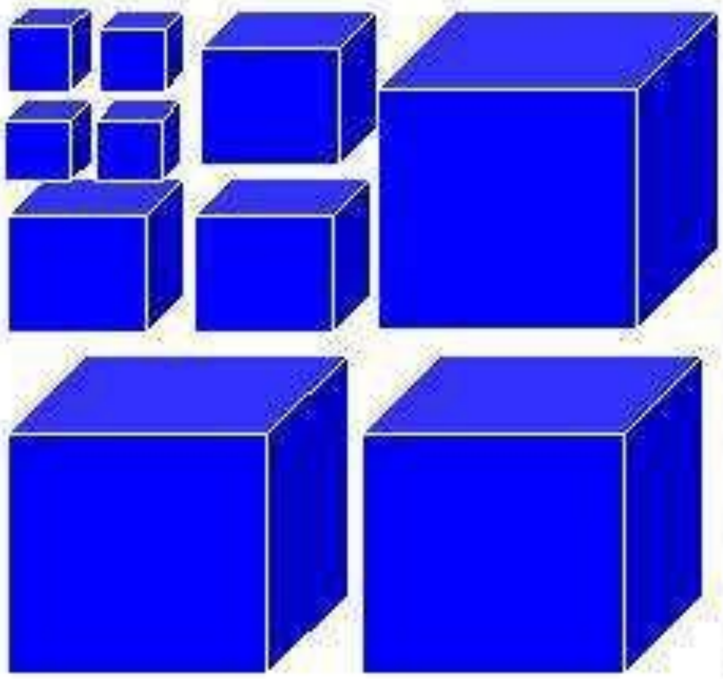
Quality scalability



Quality scalability



Quality scalability



Protecting code-block data

Segmentation symbols: special symbol sequence is coded at the end of each bit-plane. If wrong sequence is decoded, an error has occurred and the last bit-plane is corrupted (at least).

Regular predictable termination: the arithmetic coder is terminated at the end of each coding pass using a special algorithm (predictable termination). The decoder reproduces the termination and if it does not find the same unused bits at the end, an error has occurred in the last coding pass (at least).

Both mechanism can be freely mixed, but slightly decrease the compression efficiency.

Error resilience

Error effects:

In a packet body: corrupted arithmetically coded data for some code-block => severe distortion.

In a packet head: wrong body length can be decoded, code block data can be assigned to wrong code-blocks => total synchronization loss.

Bytes missing (i.e. network packet loss): combined effects of error in packet head and body

-> Protecting code-block data

Segmentation symbols: special symbol sequence is coded at the end of each bit-plane.

Regular predictable termination: the arithmetic coder is terminated at the end of each coding pass using a special algorithm (predictable termination).

Error resilience example

16:1 compression ratio. Transmission error rate 10^{-5}. No errors in codestream header.
Magnified portion shown.

No transmission errors

No error resilience

Full error resilience

Examples

Reconstructed images compressed at 0.25 bpp by means of (a) JPEG and (b) JPEG2000

Examples

(a) (b)

Reconstructed images compressed at 0.125 bpp by means of
(a) JPEG and (b) JPEG2000

Examples

JPEG 2000 (1.83 KB)

Original (979 KB)

JPEG (6.21 KB)

Conclusion - Benefits:

1. lossless and lossy compression,
2. higher image quality and compression ratios,
3. view the file at multiple resolutions,
4. one area of the image to be examined more closely using its Region Of Interest capability.

Support

Safari is the only web browser that supports JPEG 2000 natively. To view JPEG 2000 files on other browsers, you need a plug-in. The following browsers support JPEG 2000 via plug-ins:

- Chrome
- Edge
- Firefox
- Opera
- IE
- Chrome for Android

You can also open JPEG 2000 files in the latest version of Photoshop.

Library support for JPEG 2000

Program	Basic		Advanced		Language	License
	Read	Write	Read	Write		
ERDAS ECW JPEG2000 SDK	Yes	Yes	?	?	C, C++	Proprietary
FFmpeg	Yes	Yes	?	?	C	LGPL
Grok	Yes	Yes	Yes	Yes	C, C++	AGPL
J2K-Codec	Yes	No	Yes	No	C++	Proprietary
JasPer	Yes [Note 1]	Yes	No	No	C	MIT License-style
Kakadu	Yes	Yes	Yes	Yes	C++	Proprietary
OpenJPEG	Yes	Yes	Yes	Yes	C	BSD

Application support for JPEG 2000

Program	Basic [Note 1]		Advanced [Note 1]		License
	Read	Write	Read	Write	
ACDSee	Yes	Yes	?	?	Proprietary
Adobe Lightroom	No	No	No	No	Proprietary
Adobe Photoshop [Note 2]	Yes	Yes	Yes	Yes	Proprietary
Apple iPhoto	Yes	No	Yes	No	Proprietary
Apple Preview [Note 3]	Yes	Yes	Yes	Yes	Proprietary
Autodesk AutoCAD	Yes	Yes	Yes	?	Proprietary
BAE Systems CoMPASS	Yes	No	Yes	No	Proprietary
Blender[50]	Yes	Yes	?	?	GPL
Chasys Draw IES	Yes	Yes	Yes	Yes	Freeware
CineAsset	Yes	Yes	Yes	Yes	Proprietary
CompuPic Pro	Yes	Yes	?	?	Proprietary
Corel Photo-Paint	Yes	Yes	Yes	Yes	Proprietary
Daminion[51]	Yes	No	Yes	No	Proprietary
darktable[52]	?	Yes	?	?	GPL
DBGallery	Yes	No	Yes	No	Proprietary
digiKam[53][54] (KDE[55])	Yes	Yes	?	?	GPL
ECognition	Yes	Yes	?	?	Proprietary
ENVI	Yes	Yes	?	?	Proprietary
ERDAS IMAGINE	Yes	Yes	?	?	Proprietary
evince (PDF 1.5 embedding)	Yes	No	No	No	GPL v2
FastStone Image Viewer	Yes	Yes	Yes	Yes	Freeware
FastStone MaxView	Yes	No	Yes	No	Proprietary
FotoGrafix 2.0	No	No	No	No	Proprietary
FotoSketcher 2.70	No	No	No	No	Proprietary
GIMP 2.10	Yes[56]	No	?	No	GPL
Global Mapper	Yes	Yes	No	No	Proprietary
GNOME Web	Yes	–	?	–	GPL
Google Chrome	No	No	No	No	Proprietary
GraphicConverter	Yes	Yes	Yes	?	Shareware
Gwenview (KDE[55])	Yes	Yes	?	?	GPL
IDL	Yes	Yes	?	?	Proprietary
ImageMagick	Yes	Yes	Yes	Yes	ImageMagick License
Imagine (with a plugin)[57]	Yes	No	No	No	Freeware
IrfanView	Yes	Yes	No	No	Freeware
KolourPaint (KDE[55])	Yes	Yes	?	?	2-clause BSD
Mathematica	Yes	Yes	No	No	Proprietary
Matlab	via toolbox	via toolbox	via toolbox	via toolbox	Proprietary
Mozilla Firefox	No [Note 4]	No	No	No	MPL
Opera	via QuickTime	–	?	–	Proprietary
Paint Shop Pro	Yes	Yes	Yes	Yes	Proprietary
Phase One Capture One	Yes	Yes	Yes	Yes	Proprietary
PhotoFiltre 7.1	No	No	No	No	Proprietary
PhotoLine	Yes	Yes	?	?	Proprietary
Pixel image editor	Yes	Yes	?	?	Proprietary
Preview	Yes	Yes	?	?	Proprietary
QGIS (with a plugin)	Yes	Yes	?	?	GPL
Safari	Yes	–	?	–	Proprietary
SilverFast	Yes	Yes	Yes	Yes	Proprietary
XnView	Yes	Yes	Yes	Yes	Proprietary
Ziproxy	Yes	Yes	No	No	GPL

WebP

WebP

image format developed by Google and supported in Chrome, Opera and Android

optimized to enable faster and smaller images on the Web. WebP images are about 30% smaller in size compared to PNG and JPEG images at equivalent visual quality.

WebP

- Lossy compression
  - the lossy compression is based on VP8 key frame encoding.
- Lossless compression
- Transparency
  - 8-bit alpha channel is useful for graphical images
- Animation
  - It supports true-color animated images.
- Metadata
  - It may have EXIF and XMP metadata (used by cameras, for example).
- Color Profile
  - set of data that characterizes a color input or output device, or a color space, according to standards promulgated by the International Color Consortium (ICC).

WebP

- https://developers.google.com/speed/webp/

Basic principles:

predictive coding

uses the values in neighboring blocks of pixels to predict the values in a block, and then encodes only the difference.

Block prediction - smaller segments called macroblocks

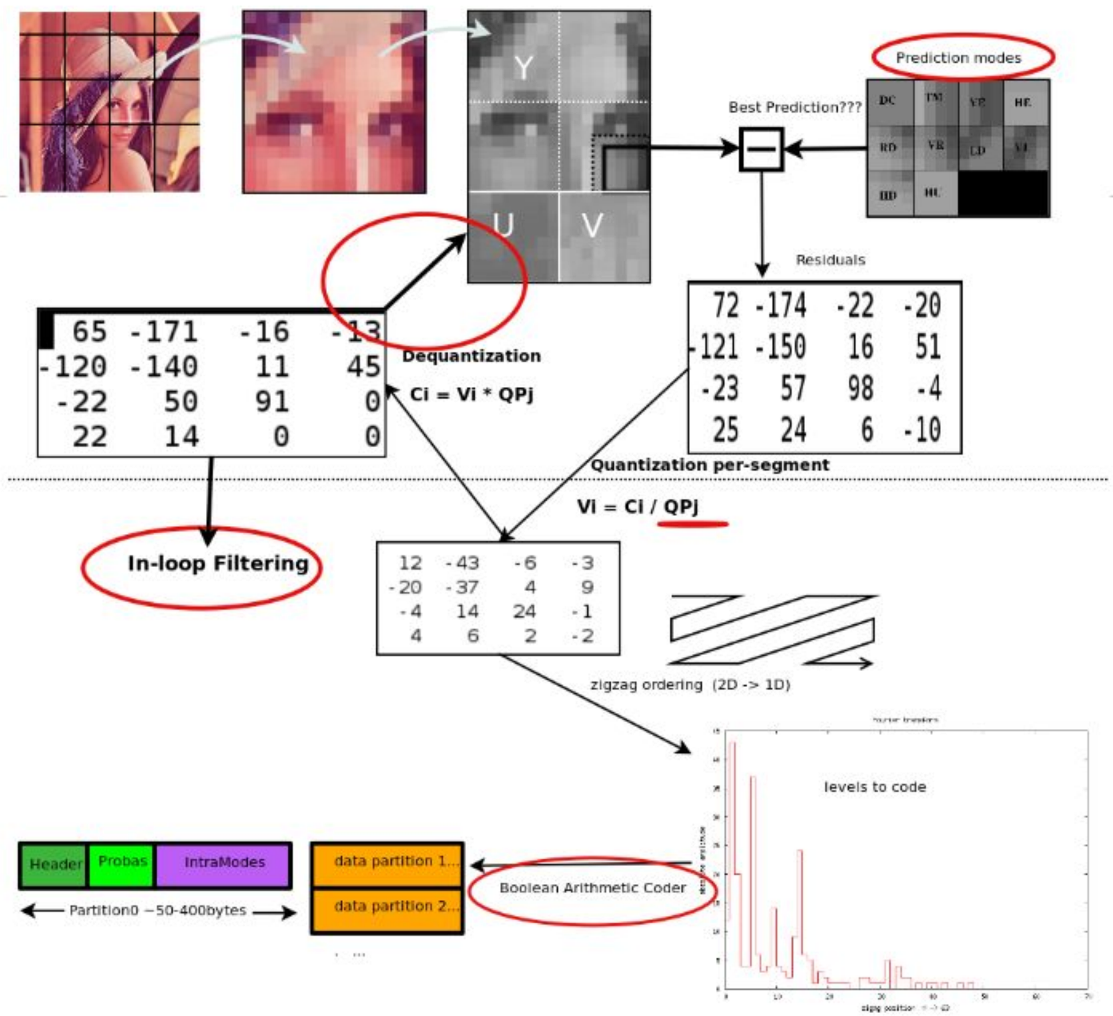
intra-frame coding

The redundant data can then be subtracted from the block, which results in more efficient compression.

WebP lossy compression.

The following diagram shows the steps involved WebP lossy compression.

The differentiating features compared to JPEG are circled in red.



Intra Prediction Modes

intra prediction modes are used with three types of macroblocks:

- 4x4 luma
- 16x16 luma
- 8x8 chroma

Four common intra prediction modes are shared by these macroblocks:

- horizontal prediction.
    - Fills each column of the block with a copy of the left column, L.
- vertical prediction.
    - Fills each row of the block with a copy of the above row, A.
- DC prediction.
    - Fills the block with a single value using the average of the pixels in the row above A and the column to the left of L.
- TrueMotion prediction.
    - In addition to the row A and column L, TrueMotion prediction uses the pixel P above and to the left of the block. Horizontal differences between pixels in A (starting from P) are propagated using the pixels from L to start each row.

WebP Support

WebP is natively supported in

Google Chrome and the Opera browser

`libwebp`