2.2 Still Image Compression Techniques

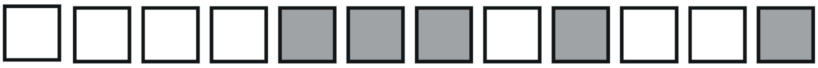
2.2.1 Telefax

New *telecommunication* standards are defined by the International Telecommunications Union (ITU-T) (in former times: CCITT = Commitée Consultatif International de Téléphonie et Télégraphie).

The standard for lossless Telefax compression was one of the early standards for still image compressions.

Images are interpreted by the Group 3 compression standard as two-tone (black-and-white) pictures. As a result a pixel can be represented by one bit. The example shows a part of a line of black-and-white pixels. Obviously, runs will be much larger than 1 in most cases, and thus run-length encoding is efficient.

Example:



run-length encoding: 4w 3b 1w 1b 2w 1b

Fax Standards of ITU-T

Standard T.4

First passed in 1980, revised in 1984 and 1988 (Fax Group 3) for error-prone lines, especially telephone lines

Two-tone (black-and-white) images of size A4 (similar to US letter size)

Resolution: 100 dots per inch (dpi) or 3,85 lines/mm vertical, 1728 samples per line

Objective:

Transmission at 4800 bits/s over the telephone line (one A4 page per minute)

Standard T.6

First passed in 1984 (Fax Group 4) for error-free lines or digital storage.

Compression Standards for Telefax (1)

Telefax Group 3, ITU-T Recommendation T.4

First approach: Modified Huffman Code (MH)

- Every image is interpreted as consisting of lines of pixels
- For every line the run-length encoding is calculated.
- The values of the run-length encoding are Huffman coded with a standard table.
- Black an white runs are encoded using different Huffman codes because the run length distributions are quite different.
- For error detection an EOL (end-of-line) code is inserted in the end of every line. This enables re-synchronization in case of bit transmission errors.

Compression Standards for Telefax (2)

Second approach : Modified Read (MR) Code

- The pixel values of the previous line are used to predict the values of the current line
- Then run-length encoding and a static Huffman code are used (same as for MH).
- The EOL code is also used.

The MH and MR coding alternates in order to avoid error propagation.

Huffman-Table for Telefax Group 3 (excerpt)

White run length	Code word	Black run length	Code word
0	00110101	0	0000110111
1	000111	1	010
2	0111	2	11
3	1000	3	10
4	1011	4	011
5	1100	5	0011
6	1110	6	0010
7	1111	7	00011
8	10011	8	000101
9	10100	9	000100
10	00111	10	0000100
11	01000	11	0000101
12	001000	12	0000111
13	000011	13	00000100
14	110100	14	00000111
15	110101	15	000011000
16	101010	16	0000010111
17	101011	17	0000011000
18	0100111	18	0000001000
19	0001100	19	00001100111
20	0001000	20	00001101000

Telefax Group 4

Telefax Group 4, ITU-T Recommendation T.6

Coding techniques: Modified Modified Read Code (MMR)

- Simplification of the MR-Codes; there are no error detection mechanisms in order to improve the compression rate.

Typical compression rates:

	business documents
Group 3:	20:1
Group 4:	50:1

For photos (and the like) the compression rate of T.6 is low because the the length of the runs is very short. Other schemes such as arithmetic coding would be more suitable.

2.2.2 Block Truncation Coding (BTC)

This simple coding algorithm is used in the compression of monochrome images. Every pixel is represented by a gray value between 0 (black) and 255 (white).

The BTC Algorithm

1. Decompose the image into blocks of size n x m pixels.
2. For each block calculate the mean value and the standard deviation as follows:

$$\mu = \frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m} Y_{i,j} \qquad \sigma = \sqrt{\frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m}(Y_{i,j}-\mu)^2}$$

where $Y_{i,j}$ is the brightness of the pixel.

3. Calculate a bit array B of size n x m as follows:

$$B_{i,j} = \begin{cases} 1 \ldots \ \mathit{if} & Y_{i,j} \leq \mu \\ 0 \ldots \ \mathit{else} & \end{cases}$$

The BTC Algorithm (continued))

4. Calculate two gray scale values for the darker and the brighter pixels:

$$a = \mu - \sigma\sqrt{p/q}$$

$$b = \mu + \sigma\sqrt{q/p}$$

p is the number of pixels having a larger brightness than the mean value of the block, q is the number of pixels having a smaller brightness.

5. Output: (Bit matrix, *a, b*) for every block.

Decompression with BTC

For every block the gray value of each pixel will be calculated as follows:

$$Y'_{i,j} = \begin{cases} a \ldots \ \ if & B_{i,j} = 1 \\ b \ldots \ else & \end{cases}$$

Compression rate example

Block size :	4 x 4
Original (gray values)	1 byte per pixel
Encoded representation:	bit matrix with 16 bits + 2 x 8 bits for *a* and *b*

=> reduction from 16 bytes to 4 bytes, I.e., the compression rate is 4:1.

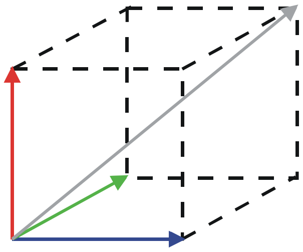
2.2.3 Color Cell Compression

Color cell compression (CCC) is a algorithm for the compression of color images. In principle, BTC can be used for color images rather than for gray scale images by compressing the three color components separately. However, the Color Cell Compression technique leads to a better compression rate.

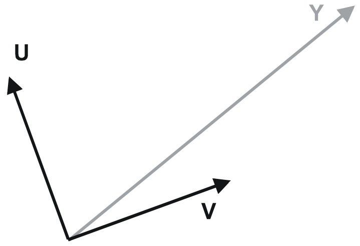
Color Models

The classical color model for the computer is the **RGB model**. The color value of a pixel is the sum of the intensities of the color components red, green and blue. The maximum intensity of all three components results in white.

In the **YUV model, Y** represents the value of the luminance (brightness) of the pixel, U and V are two vertical color vectors. The color value of an pixel can be easily converted from model to model.



RGB-Model



YVU-Model

A advantage of the YUV model is that the value of the luminance is directly available. That means that a gray scale version of the image can be created very fast. Another point is that the compression of the luminance component can differ from the compression of the chrominance components.

The CCC Algorithm

1. Decompose the image into blocks of size n x m pixels.
2. The brightness of a pixel is computed as follows:

$$Y = 0.3P_{red} + 0.59P_{green} + 0.11P_{blue}$$

Y=0 is equivalent to black, Y=1 is equivalent to white

3. For c = red, green, blue calculate the mean color value of the pixel as follows:

$$a_c = \frac{1}{q} \sum_{Y_{i,j} \le \mu} P_{c,i,j} , \qquad b_c = \frac{1}{p} \sum_{Y_{i,j} \succ \mu} P_{c,i,j}$$

Again, *q* and *p* are the numbers of pixels with a brightness larger or smaller than the mean value, respectively.

The CCC Algorithm (continued)

4. Calculate a bit array B of size n x m as follows:

$$B_{i,j} = \begin{cases} 1 \ldots & if \quad Y_{i,j} \le \mu \\ 0 \ldots & else \end{cases}$$

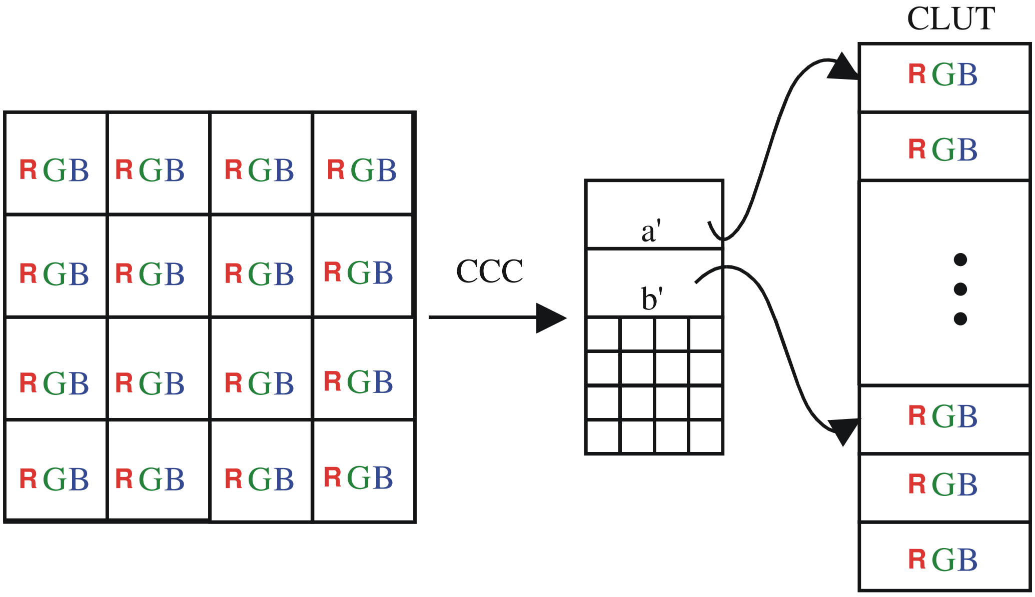
5. The color values *a* = (a_{red}, a_{green}, a_{blue}) and *b* = (b_{red}, b_{green}, b_{blue}) are now quantized onto a color lookup table. We get the values *a'* and *b'* as an index for the Color Lookup Table (CLUT).
6. Output: (bit matrix, *a'*, *b'*) for every block

Decompression of CCC Images

For every block the decompression algorithm works as follows:

$$P'_{i,j} = \begin{cases} CLUT\ [a'] \ldots & if \quad B_{i,j} = 1 \\ CLUT\ [b'] \ldots & else \end{cases}$$

Usage of the Color Lookup Table in CCC

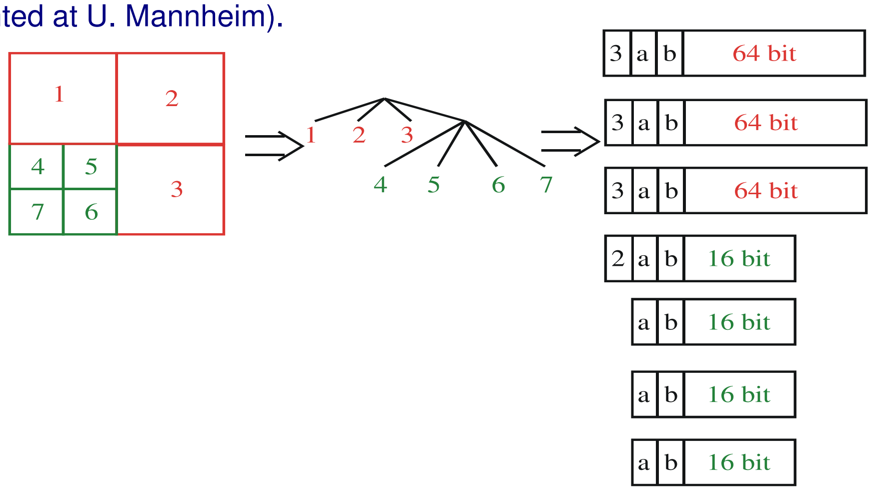

CLUT
RGB
CCC
a'
b'

Extended Color Cell Compression (XCCC)

This method is an extension of CCC for further improvement of the compression rate.

Idea

Use a hierarchy of block sizes. In the first step the algorithm tries to code a large block with CCC. If the difference to the true color values is greater than a given threshold the block is divided into four parts. The algorithm works recursively (invented at U. Mannheim).


1
2
3
4
5
7
6
1 2 3
4 5 6 7
3 a b 64 bit
3 a b 64 bit
3 a b 64 bit
2 a b 16 bit
a b 16 bit
a b 16 bit
a b 16 bit

2.2.4 A Brief Introduction to Transformations

Motivation for Transformations

Improvement of the compression ratio while maintaining a good image quality.

What is a transformation?

- Mathematically: a change of the base of the representation
- Informally: representation of the same data in a different way.

Motivation for the use of transformations in compression algorithms: **In the frequency domain, leaving out data is often less disturbing to the human visual (or auditive) system than leaving our data in the original domain.**

The Frequency Domain

In the frequency space the signal (one-dimensional or two-dimensional) is represented as an overlay of base frequencies. The coefficients of the frequencies specify the amplitudes with which the frequencies occur in the signal.

The Fourier Transform

The Fourier transform of a function *f* is defined as follows:

$$\hat{f}(t) = \int f(x) e^{-2\pi i t x} dx$$

where *e* can be written as

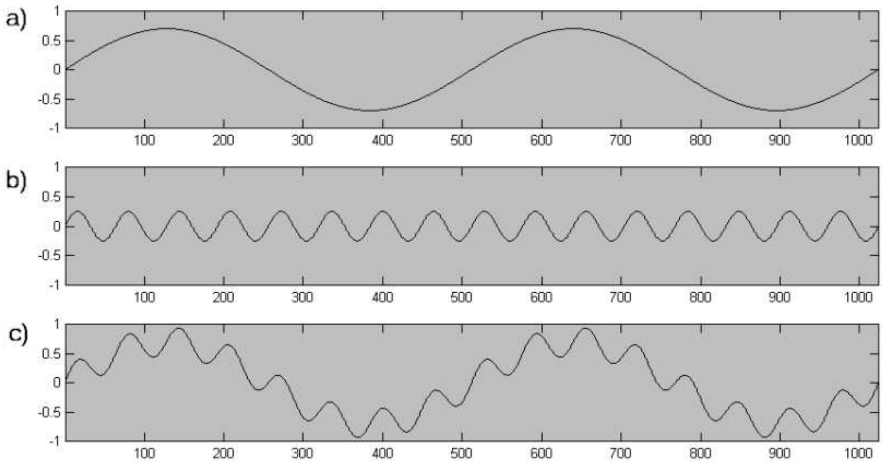
$$e^{ix} = \cos(x) + i \sin(x)$$

Note:

The *sin* part makes the function complex. If we only use the *cos* part the transform remains real-valued.

Overlaying the Frequencies

A transform asks how the amplitude for each base frequency must be chosen such that the overlay (sum) approximates the original function.



The output signal (c) is represented as a sum of the two sine oscillation (a) und (b).

One-Dimensional Cosine Transform

The Discrete Cosine Transform (DCT) is defined as follows:

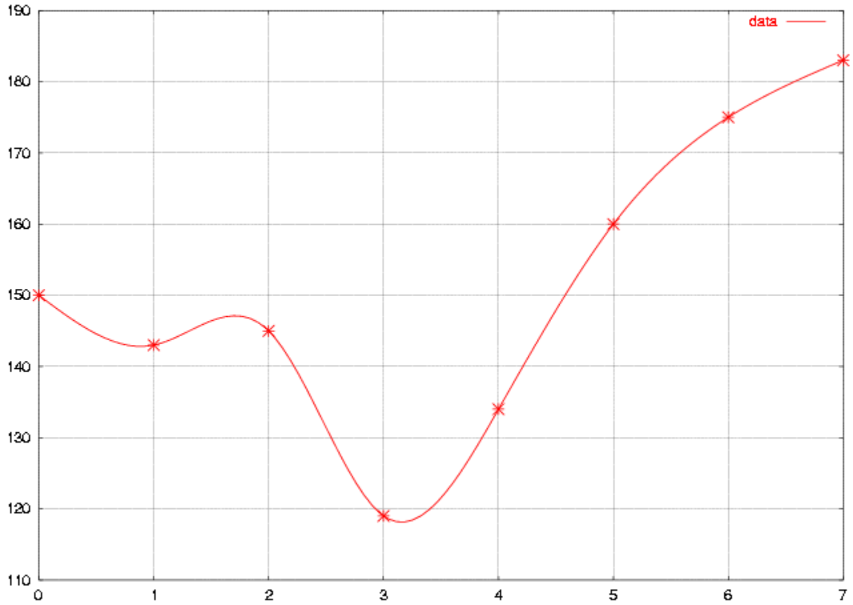
$$S_u = \frac{1}{2} C_u \sum_{x=0}^{7} s_x \cos \frac{(2x+1)u\pi}{16}$$

with

$$C_u = \begin{cases} \frac{1}{\sqrt{2}} \; for \; u = 0 \\ 1 \; otherwise \end{cases}$$

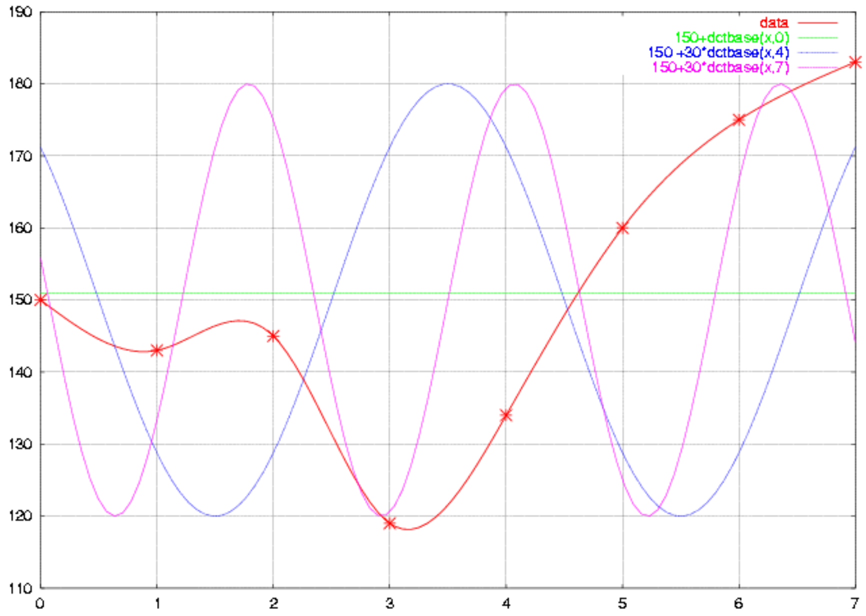
Example for a 1D Approximation (1)

The following one-dimensional signal is to be approximated by the coefficients of a 1D–DCT with eight base frequencies.



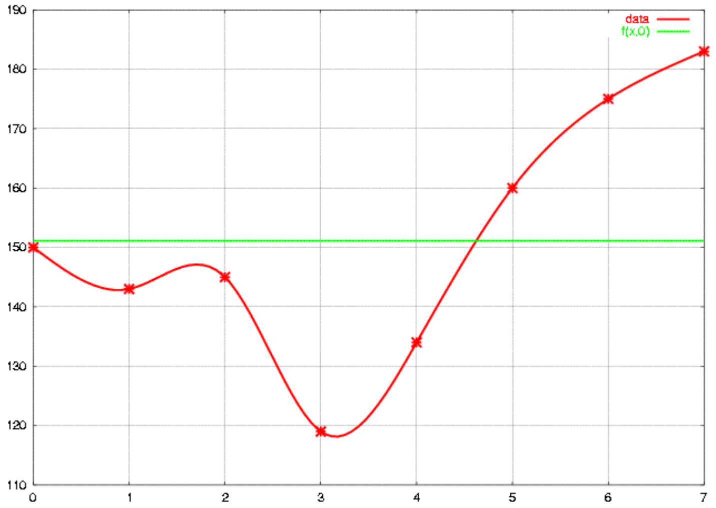
Example for a 1D Approximation (2)

Some of the DCT kernels to be used in the approximation.



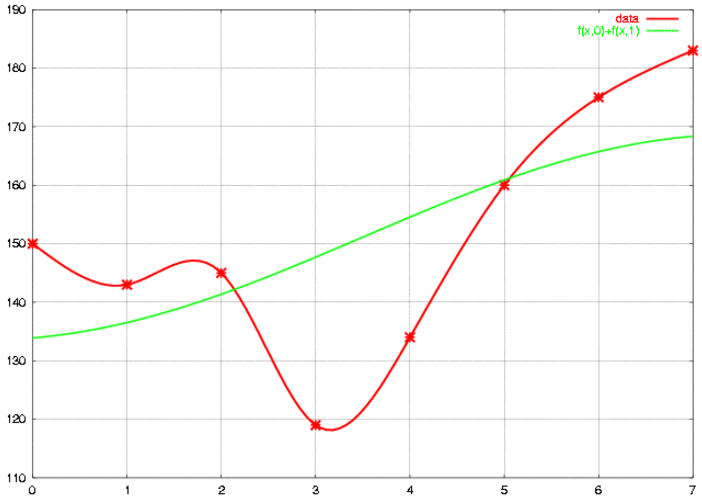
Example for a 1D Approximation (3)

DC coefficient



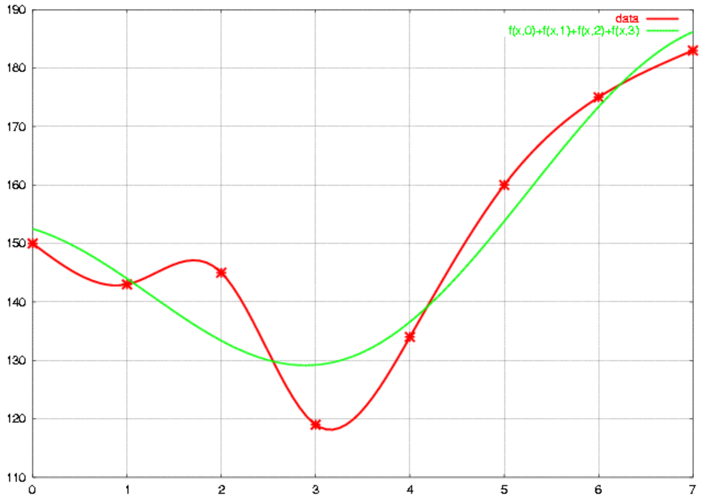
Example for a 1D Approximation (4)

DC coefficient + 1st AC coefficient



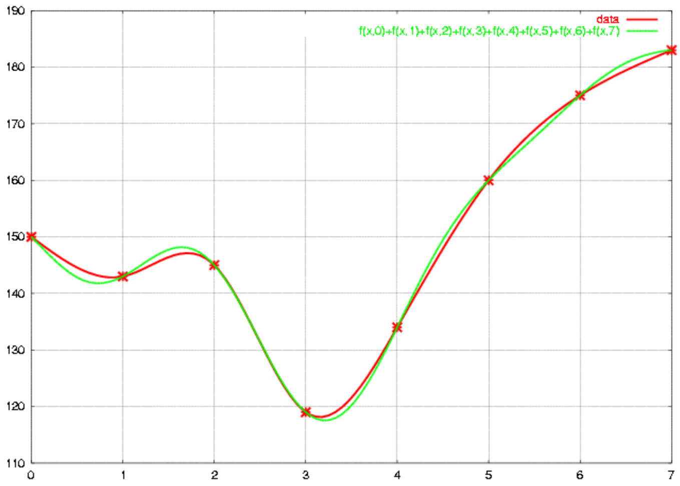
Example for a 1D Approximation (5)

DC coefficient + AC coefficients 1-3



Example for a 1D Approximation (6)

DC coefficient + AC coefficients 1-7



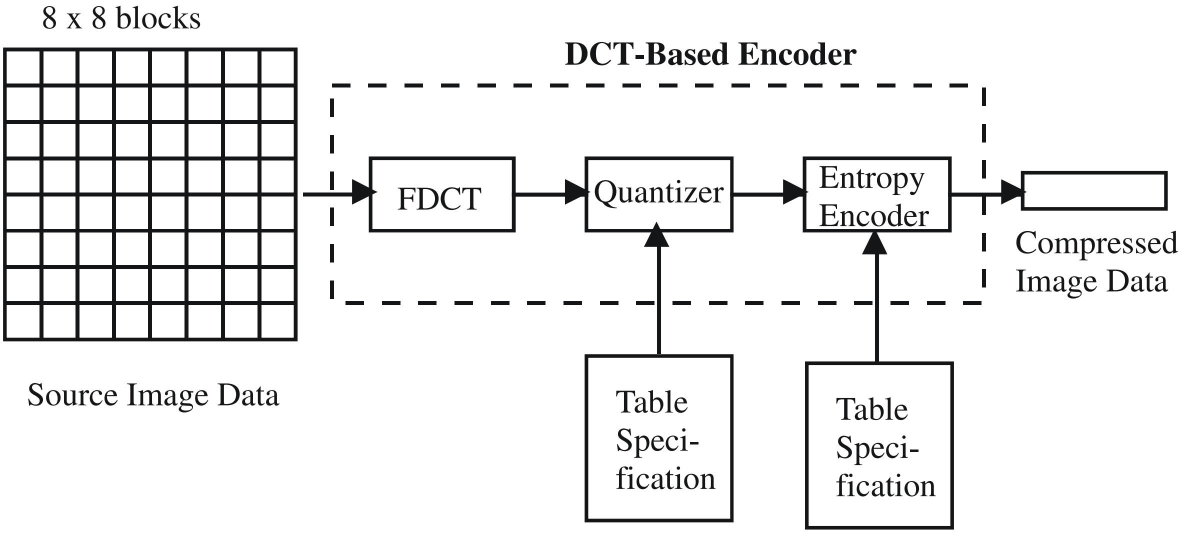
2.2.5 JPEG

The Joint Photographic Experts Group (JPEG, a working group of ISO) has developed a very efficient compression algorithm for still images which is commonly referred to under the name of the group.

Compression is done in in four steps:

1. Image preparation
2. Discrete Cosine Transform (DCT)
3. Quantization
4. Entropy Encoding

The DCT-based JPEG Encoder



8 x 8 blocks

Source Image Data

DCT-Based Encoder

FDCT

Quantizer

Entropy Encoder

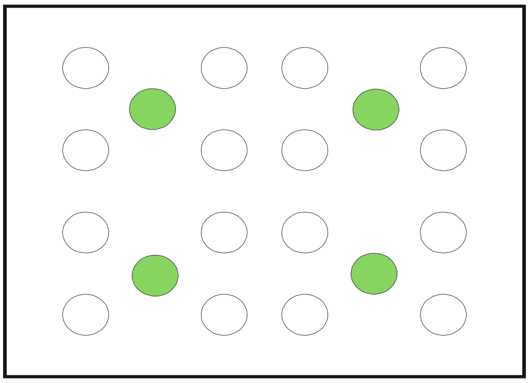
Table Specification

Table Specification

Compressed Image Data

Coding of the Color Components with a Lower Resolution (“Color Subsampling")

One advantage of the YUV color model is that the color components U and V of a pixel can be represented with a lower resolution than the luminance value Y. The human eye is more sensitive to brightness than to variations in chrominance. Therefore JPEG uses **color subsampling**: for each group of four luminance values one chrominance value for each U and V is sampled.



○ **= Luminance**

● **= Chrominance**

In JPEG four Y blocks of size of 8x8 together with one U block and one V block of size 8x8 each are called a macroblock.

JPEG "Baseline" Mode

A compression algorithm based on a transform from the time domain into the frequency domain.

Image transformation

FDCT (Forward Discrete Cosine Transform). Very similar to the Fourier transformation. It is used separately for every 8x8 pixel block of the image.

$$S_{vu} = \frac{1}{4} C_u C_v \sum_{x=0}^{7} \sum_{y=0}^{7} s_{yx} \cos\frac{(2x+1)u\pi}{16} \cos\frac{(2y+1)v\pi}{16}$$

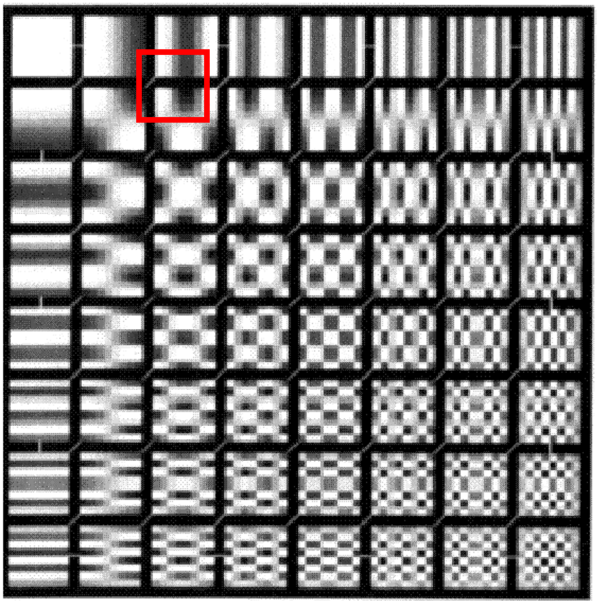
with

$$C_u, C_v = \begin{cases} \frac{1}{\sqrt{2}}\, for\; u,v = 0 \\ 1\; otherwise \end{cases}$$

This transformation is computed 64 times per block. The result are 64 coefficients in the frequency domain.

Base “Frequencies” for the 2D-DCT

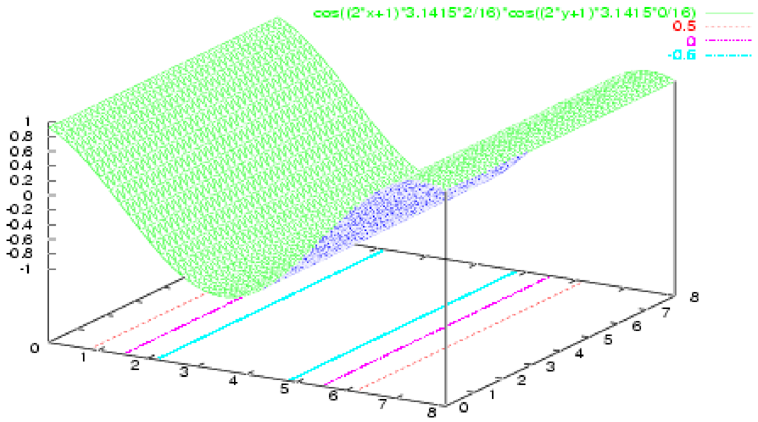
To cover an entire block of size of 8x8 we use 64 base “frequencies”, as shown below.



Example of a Base Frequency

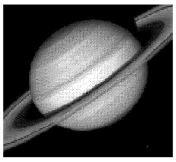
The figure below shows the DCT kernel corresponding to the base frequency (0,2) shown in the highlighted frame (first row, third column) on the previous page.

$$\frac{\cos(2x+1)\cdot 2\pi}{16}\cdot\frac{\cos(2y+1)\cdot 0\pi}{16}$$

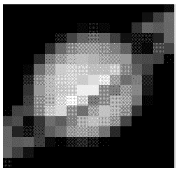


Example: Encoding of an Image with the 2D-DCT and block size 8x8

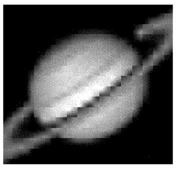
Original



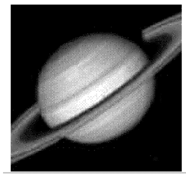
1 Coefficient



4 Coefficients

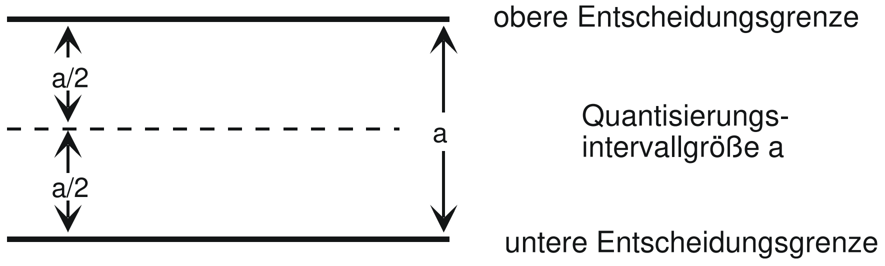


16 Coefficients



Quantization

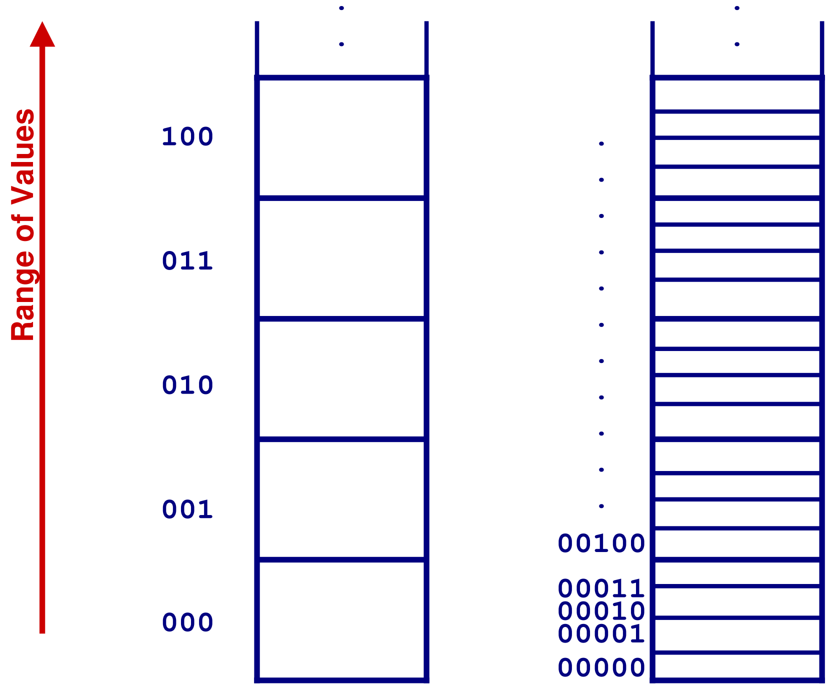
The next step in JPEG is the quantization of the DCT coefficients. Quantization means that the range of allowable values is subdivided into intervals of fixed size. The larger the intervals are chosen, the larger the quantization error will be when we decompress.


obere Entscheidungsgrenze
a/2
a/2
a
Quantisierungs-intervallgröße a
untere Entscheidungsgrenze

Maximum quantization error: a/2

Quantization: Quality vs. Compression Ratio

Coarse Quantization


Range of Values
100
011
010
001
000
00100
00011
00010
00001
00000

Quantization

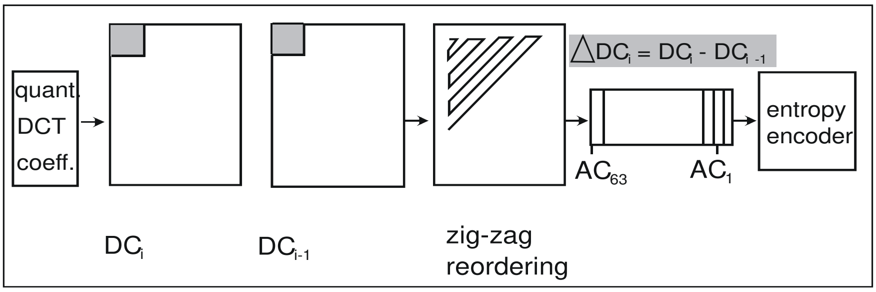
In JPEG the number of quantization intervals can be chosen separately for each DCT coefficient (Q-factor). The Q-factors are specified in a **quantization table**.

Entropy-Encoding

The quantization step is followed by an entropy encoding (lossless encoding) of the quantized values:

- The DC coefficient is the most important one (basic color of the block). The DC coefficient is encoded as the difference between the current DC coefficient value and the one from the previous block (differential coding).
- The AC coefficients are processed in zig-zag order. This places coefficients with similar values in sequence.

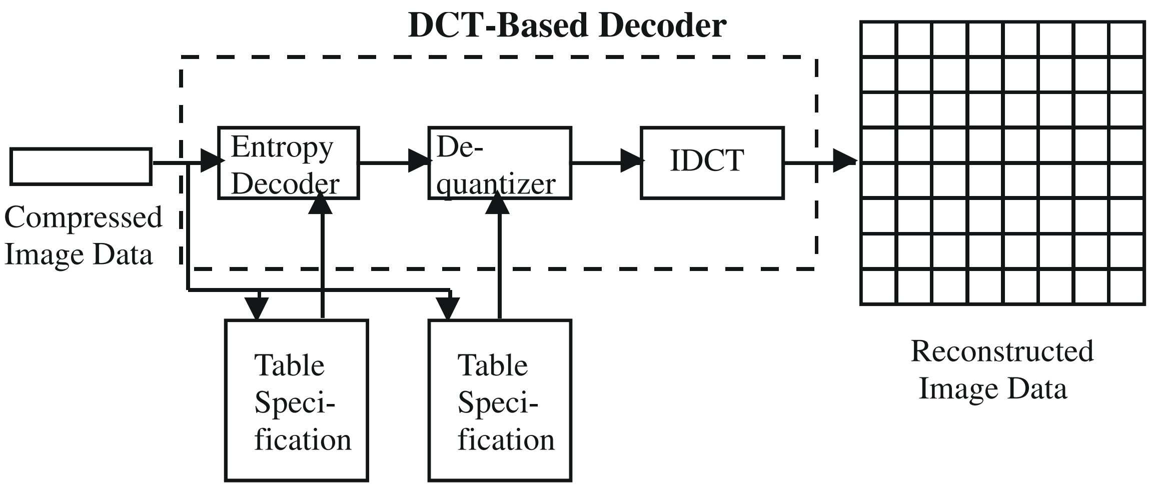
Quantization and the Entropy Encoding



Zig-zag reordering of the coefficients is better than a read out line-by-line because the input to the entropy encoder has a few non-zero and many zero coefficients (representing higher frequencies, i.e., sharp edges). The non-zero coefficients tend to occur in the upper left-hand corner of the block, the zero coefficients in the lower right-hand corner.

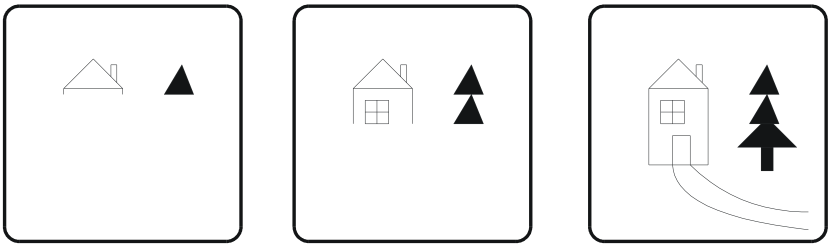
Run-length encoding is used to encode the values of the AC coefficients. The zig-zag read out maximizes the run-lengths. The run-length values are then Huffman-encoded (this is similar to the Fax compression algorithm).

JPEG Decoder

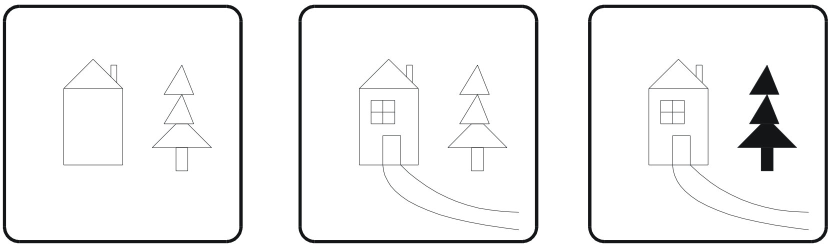

DCT-Based Decoder
Compressed Image Data
Entropy Decoder
De-quantizer
IDCT
Table Speci-fication
Table Speci-fication
Reconstructed Image Data

Different Modes in JPEG

JPEG Sequential Mode



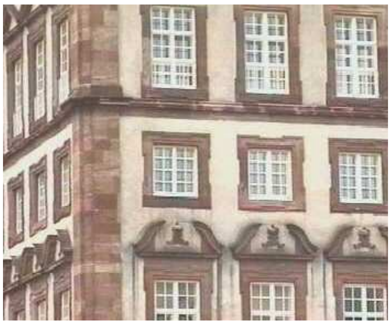
JPEG Progressive Mode



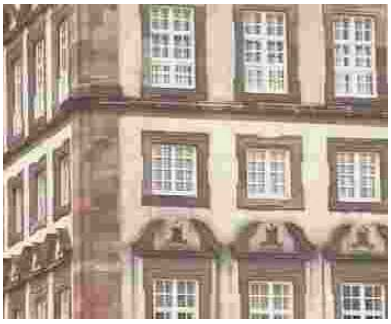
Quantization Factor and Image Quality

Example: Palace in Mannheim

Palace, original image

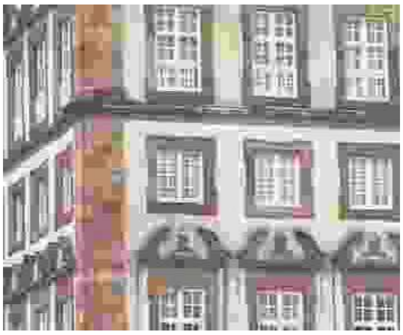


Palace image with Q=6

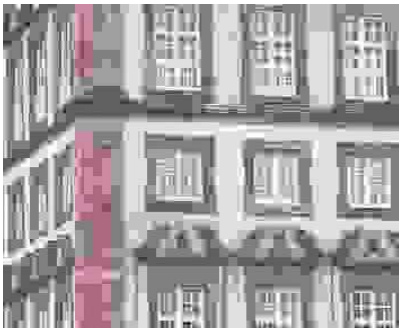


Palace Example (continued)

Palace image with Q=12

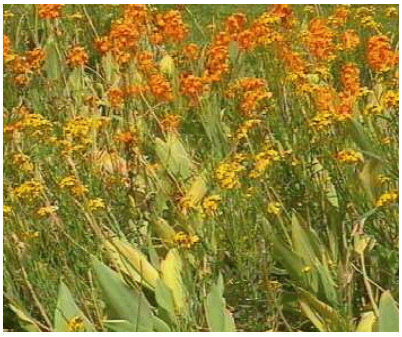


Palace image with Q=20

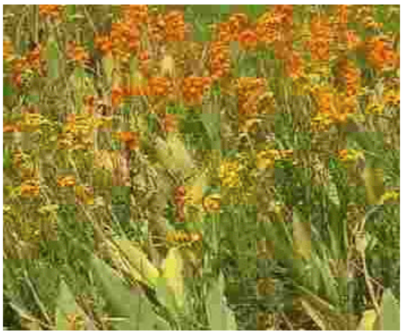


Flower Example

Flower, original image

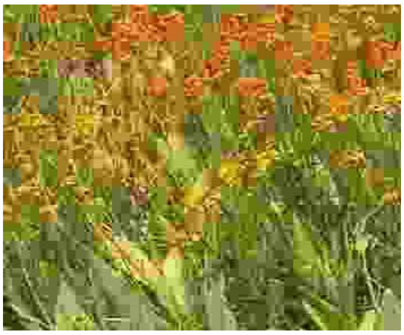


Flower with Q=6

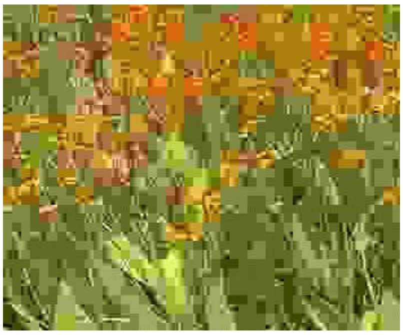


Example flower (Continuation)

Flower with Q=12



Flower with Q=20



2.2.6 Compression with Wavelets

Motivation

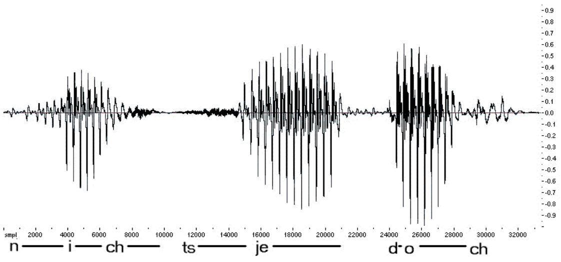
Signal analysis and signal compression.

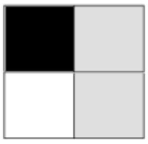
Known: image compression algorithm
- based on the pixel values (BTC; CCC; XCCC)
- based on transformation in the frequency domain
  (Fourier-Transformation, DCT)

What is a transformation ?
- In mathematics: Changing of the bases
- Interpretation: Representation of another way.

Example

"Standard" representation of a signal:
- Audio signal with frequencies over the time
- Image as pixel values on locations





=

0	224
255	144

That is not the "real" signal, but we are familiar with it.

The frequency domain

In the representation, we are familiar with, the values represents a time/location relation. That is called time domain.

In the Frequency domain the *changes* of a signal are in the focus.

- How strong is the variation of the amplitude of the audio signal?
- How strong varies the transition from one pixel to the next ?
- Which frequencies are in the given signal ?

Review: Fourier-Transformation

We remember our self:

Fourier-Transformation from f:

$$\hat{f}(t) = \int f(x) e^{-2\pi i t x} dx$$

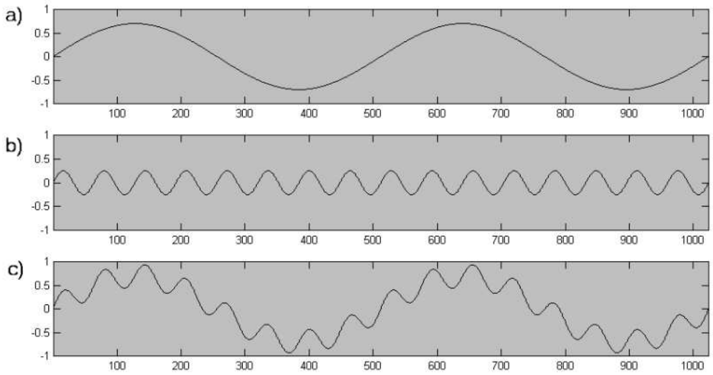
And the e- function could be written as:

$$e^{ix} = \cos(x) + i \sin(x)$$

Sinus and Cosines are known: They reach from $-\infty$ to ∞

Suited transformation

A transformation weights every single frequency to prepare it for a accumulation of all frequencies for to reconstruction of the original signal.



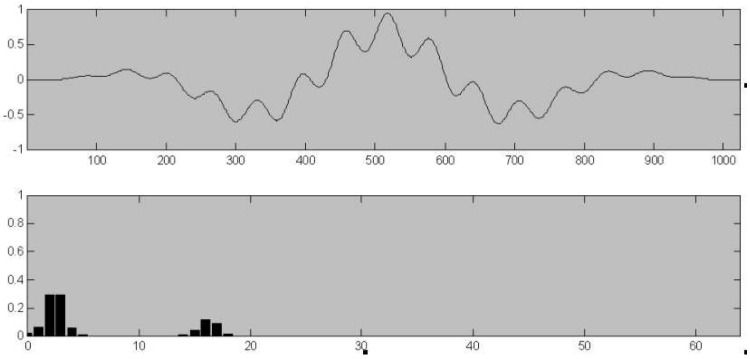
The output signal (c) is represented as a sum of the two sine oscillation (a) und (b).

By the way:

JPEG uses the DCT and not the FT – because the sinus function changes the FT to a complex transformation. If the cosines is used, the FT stands in the real room.

$$e^{ix} = \cos(x) + i\sin(x)$$

Problems with the Fourier-Transformation



If a signal with a high “locality” should be represented, great many sine and cosine oscillations must be added. The example shows a signal (upper figure), which disappears on the edges. It is put together with sine oscillations from 0-5 Hz and 15-19 Hz (lower figure).

Wanted: A frequency representation by functions, which features a high locality. With this functions it is possible to construct the signal only with a view addends from different frequencies.

The solution: Wavelets !

What is a Wavelet ?

A wavelet is a function ψ, which satisfy the following permissibility condition:

$$0 < c_{\psi} := 2\pi \int_{R} \frac{|\hat{\psi}(\omega)|^{2}}{|\omega|} d\omega < \infty$$

As follows:

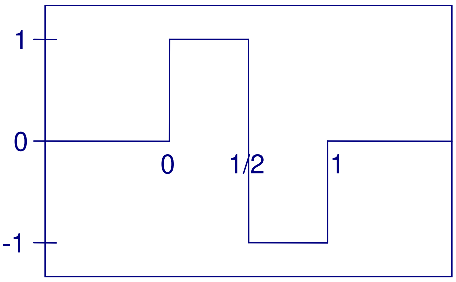
$$0 = \hat{\psi}(0) = \int \psi(x) e^{-2\pi i 0 x} dx = \int \psi(x) dx$$

A wavelet is a function, which exists only during a limited interval <> 0 and which has the same “over the curve” like “under the curve”.

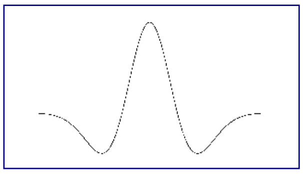
wavelet = small wave (engl.)
ondelette = petite onde (frz.)
Wellchen = kleine Welle (dt.)

Example-Wavelets

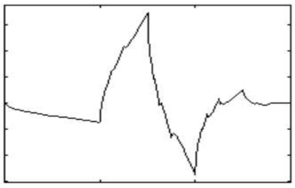
Haar-Wavelet



Mexican Hat



Daubechies-2



Practical application

Limitations:

Instead of the general theory, we consider only:

- discrete Wavelet Transformations (DWT)
- dyadic DWT, that means “Factor 2”
- orthogonal Wavelets

... From now on all things are very easy and “Hands On” ...

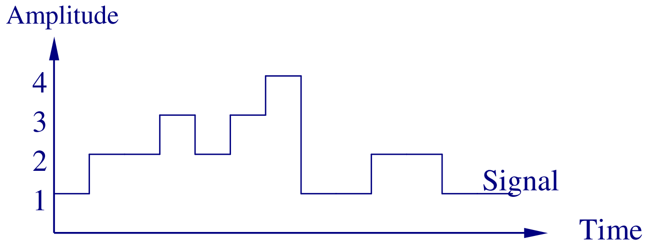
Stéphane Mallat discovered a correlation between orthogonal Wavelets and Filters, which are known in the signal processing and the engineering science a long time before.

That’s the reason for the terms “high-pass filter” (~Wavelet) and “low-pass filter” (~Scaling Function).

Example: Haar-Transformation (I)

We execute a Wavelet-Transformation with the Haar-Wavelet without any care about the theory. After all, we will give the relation to the learned theory…

Objective: Decomposition of a one dimensional signal (e.g. Audio) in Wavelet-coefficients.



(a) Graphical Representation

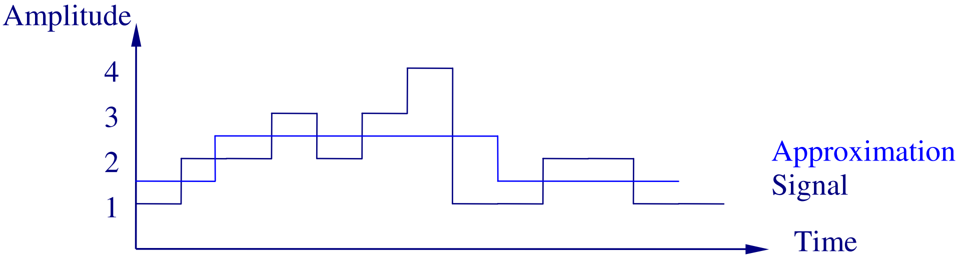
1	2	2	3	2	3	4	1	1	2	2	1	1

(b) Representation by coefficients over the time

Example: Haar-Transformation (II)

How can we represent the signal in another way without any loss of information ?

A rougher representation uses (e.g.) the mean value between to values.



1	2	2	3	2	3	4	1	1	2	2	1	1	Signal

1.5	2.5	2.5	2.5	1.5	1.5		Approximation

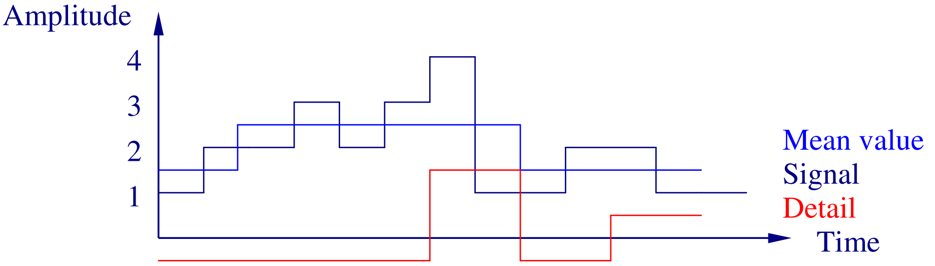
1/2	1/2

Filter for the calculation of the mean value (Approximation):

A filter “will be situated” over the signal. The values, which are laying “one upon the other” will be multiplicated, and all together added (convolution).

Example: Haar-Transformation (III)

With the representation of the signal by the approximation we loose Information ! To reconstruct the signal we must know how far the two values are away from the mean value.



1	2	2	3	2	3	4	1	1	2	2	1	1	Signal

1.5	2.5	2.5	2.5	1.5	1.5		Mean value
-0.5	-0.5	-0.5	1.5	-0.5	0.5		Detail

Filter for the calculation of the differences (detail):

1/2	-1/2

Example: Haar-Transformation (IV)

We have decomposed the original signal in another representation. Notice: The number of coefficients we need for a complete representation is unchanged. (That is the meaning of the mathematical term “base transformation”).

1	2	2	3	2	3	4	1	1	2	2	1	1	Signal
1.5		2.5		2.5		2.5		1.5		1.5			Mean value
-0.5		-0.5		-0.5		1.5		-0.5		0.5			Detail

To reconstruct the original signal with the approximation and the details synthesis filters used.

1	1	Synthesis filter for the 1. Value
1	-1	Synthesis filter for the 2. Value

With that:

1.5*1+(-0.5)*1 = 1 (Synthesis of the 1. value)
1.5*1+(-0.5)*(-1) = 2 (Synthesis of the 2. value)
2.5*1+(-0.5)*1 = 2 (Synthesis of the 1. value)
2.5*1+(-0.5)*(-1) = 3 (Synthesis of the 2. value)
etc.

Example: Haar-Transformation (V)

All together we need four filters for the decomposition and the synthesis of the original signal:

- Approximation filter for the mean value
- Detail filter for the differences
- Synthesis filter for the 1. Value
- Synthesis filter for the 2. Value

1/2	1/2
1/2	-1/2
1	1
1	-1

The decomposition of the signal in approximations and details can now be continued with the input signal.

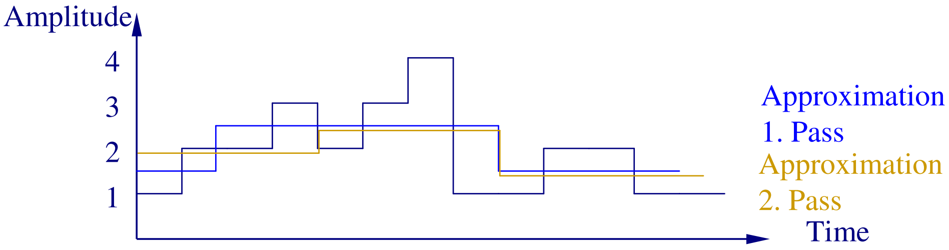
Declarations:
Approximation filter =: Low-pass filter
Detail filter =: High-pass filter

Treatment of the signal in rougher resolutions =: Multi scale analysis.

Example: Haar-Transformation (VI)

Recursion with the calculated approximation (low-pass filtered, with a “rougher” version of the input signal):

Store the details (they will be needed for the synthesis), and work further with the approximations.



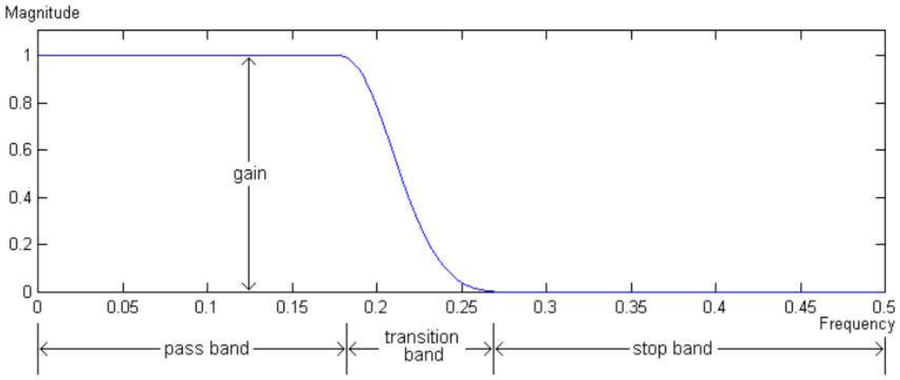
1 2 2 3 2 3 4 1 1 2 2 1 1 Signal

1.5 2.5 2.5 2.5 1.5 1.5 Approx. 1

2 2.5 1.5 Approx. 2

2.25 Approx. 3

High- and Low-pass filter



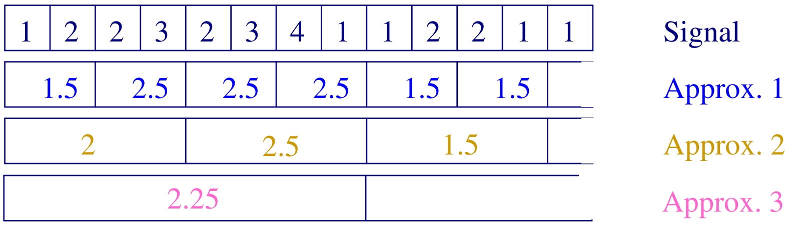
The figure shows a low-pass filter. A low-pass filter let pass lower frequencies (multiplication with 1) and block higher frequencies (multiplication with 0). The transfer from the “Pass Band” to the “Stop Band“ is in real not very sharp, that means, that mostly exist a frequency band which is neither filtered completely nor be unchanged. – the so called “Transition Band” – but it is not in our focus.

The high pass filter works vice versa.

Multiresolution Analysis

If a signal (a function, a “domain”) will be successively viewed in rougher scales (e.g. Haar-Transformation), we call it Multiresolution Analysis.

We look back



1 2 2 3 2 3 4 1 1 2 2 1 1 Signal

1.5 2.5 2.5 2.5 1.5 1.5 Approx. 1

2 2.5 1.5 Approx. 2

2.25 Approx. 3

A coefficient of the signal represents a value. After the first pass through the Wavelet-Transformation the coefficient of the low-pass includes information about two signal values. In step 2 the coefficient includes information about four signal values etc.

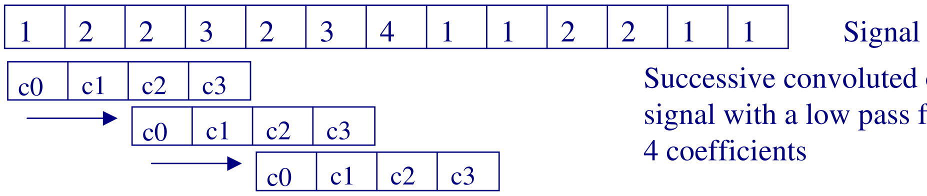
The “scope of engagement” of a coefficient will be stretched with every step. We are doing always the same but in different resolutions.

Common Wavelet-Transformation

We have learned something about the four filters of the Haar-Transformation and the synthesis.

1/2	1/2
1/2	-1/2
1	1
1	-1

Common Wavelet-Filters are more complex. For a complete transformation we need *in any case* a low-pass filter, a high-pass filter an two synthesis filter. The filter will be layed over the signal and convoluted with them (that means a multiplication and a addition). After that, the filter will be moved about 2 elements of the signal.



Successive convoluted of the signal with a low pass filter with 4 coefficients

Important notice: With all filters with a length > 2 exists a boundary value problem!

The usage of Wavelets: Audio analysis

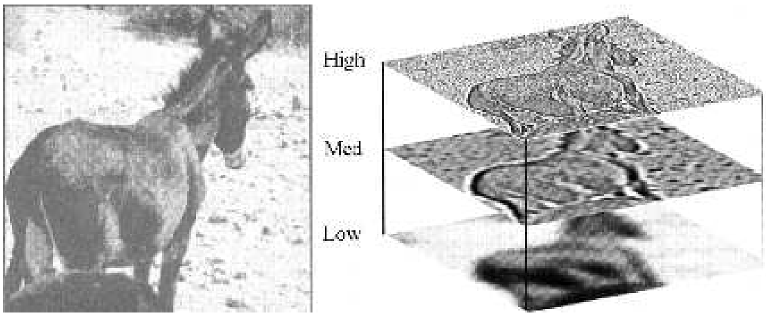
In each iteration step, the Discrete Wavelet Transformation DWT decomposes a signal into half of the resolution.

Human perception of audio signals ranges from 20 Hz to 20 kHz. The acoustic range is perceived in a logarithmic way: The frequency range from 100 Hz to 200 Hz is perceived with the same number of receptors as the range from 200 Hz to 400 Hz.

This is just what the Wavelet Transform models.

The usage of Wavelets: Image compression

It is useful, to present a signal in such a way in which a human being percepts it.



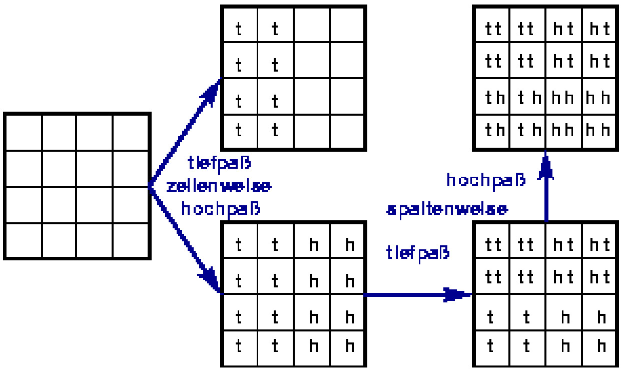
The segmentation “from rough to fine” makes it possible to start to view a image with the roughest presentation. If the memory/bandwidth is sufficient it is possible to present more details.

If the memory/bandwidth is insufficient to present the image lossless it is also possible to present the most important information.

Filters in multi-dimensions

With the conception of applications which use the Wavelet-Transformation with images (2-dim), videos (3-dim) it is necessary to think about the algorithm of multi-dimension filters, instead of the one-dimensional Wavelet-Transformation which is used with for audio.

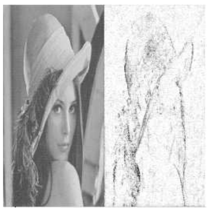
If we look at a small scope of wavelets, the so called “separable Wavelets” we can start with one-dimensional filters and use them also for the other dimensions.



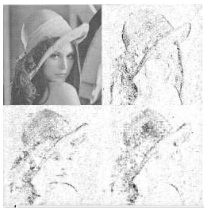
The figure shows a still image, which is first filtered line wise with the high-pass and low-pass and then the filters are used to transform the columns.

Image compression with Wavelets (I)

Original image „Lenna“

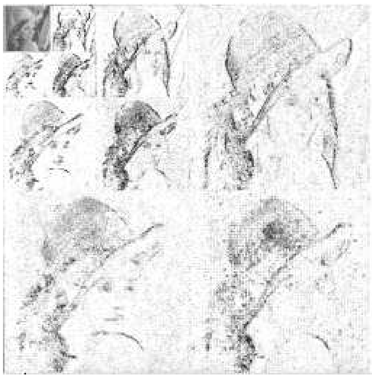


The 2-d problem is solved, if we start first with the lines. The approximation is written to left, the details to right.



The line wise filtered image is the initial image for the column wise filtering process. This results 4 versions inside a complete recursion step.

Image compression with Wavelets (II)



During the storage of the details (they are not treated any more), the approximation will now be filtered by a low-pass and a high-pass filter. The resulted details will be stored, the approximations will be treated further more…

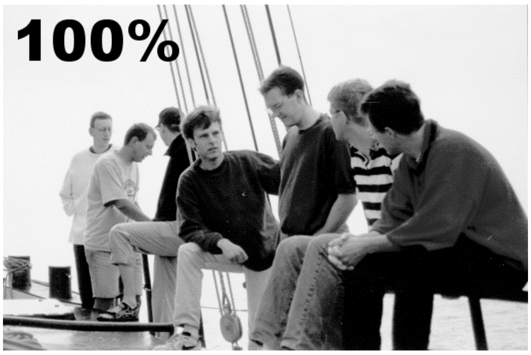
JPEG-2000 (I)

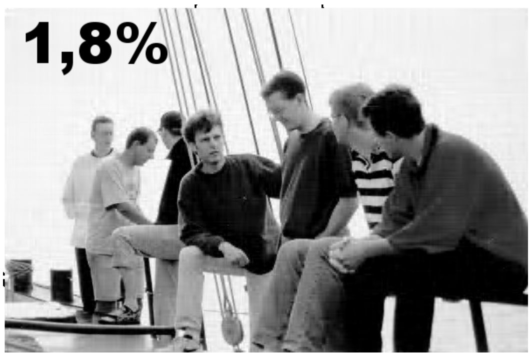
The new standard JPEG-2000 bases not any more on the DCT (like the JPEG), it bases on the Wavelet Transformation. The viewable artifacts, which results by a higher compression rate (and the implicit information losses) are not so disturbing for the human perception like the block artifacts by JPEG.

Conclusion:

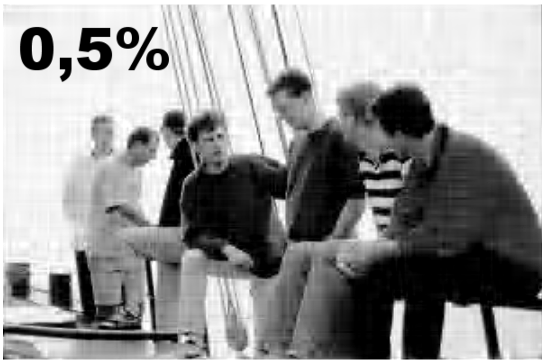
A serial of still images with different compression rates.

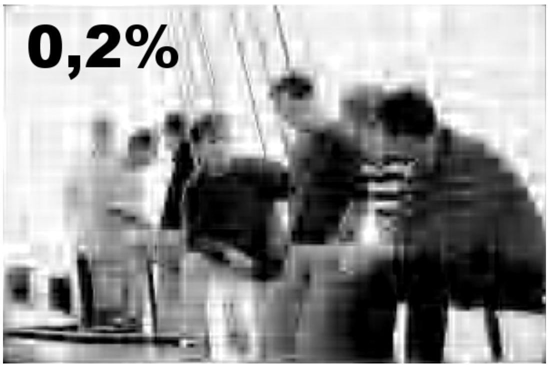
JPEG-2000 (II)





JPEG-2000 (III)





2.2.7 Image Compression with Fractals

Theory of the Fractals = Theory of the **self-similarity**. Self similarity is describable in a mathematical way.

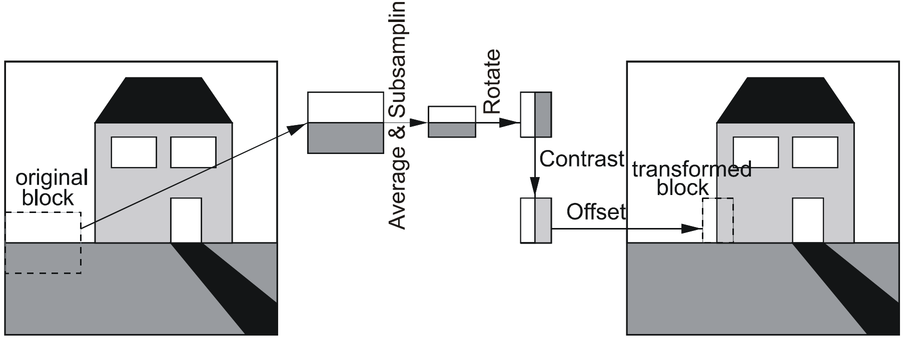
Examples from nature: Coastline of an island

Idea for the image compression

- Very often a part of a image is similar to another part of the image. More exact: It is possible to calculate with simple mathematical functions (translation, rotation and scaling) from one part of the image another part.
- Encoding: Full-Encoding of the first part of the image for the similar parts. Output of the **transformation operands.**

Image compression with Fractals

Example



Literature:

M. F. Barnsley, L. P. Hurd: Fractal Image Compression. A. K. Peters Ltd, 1993