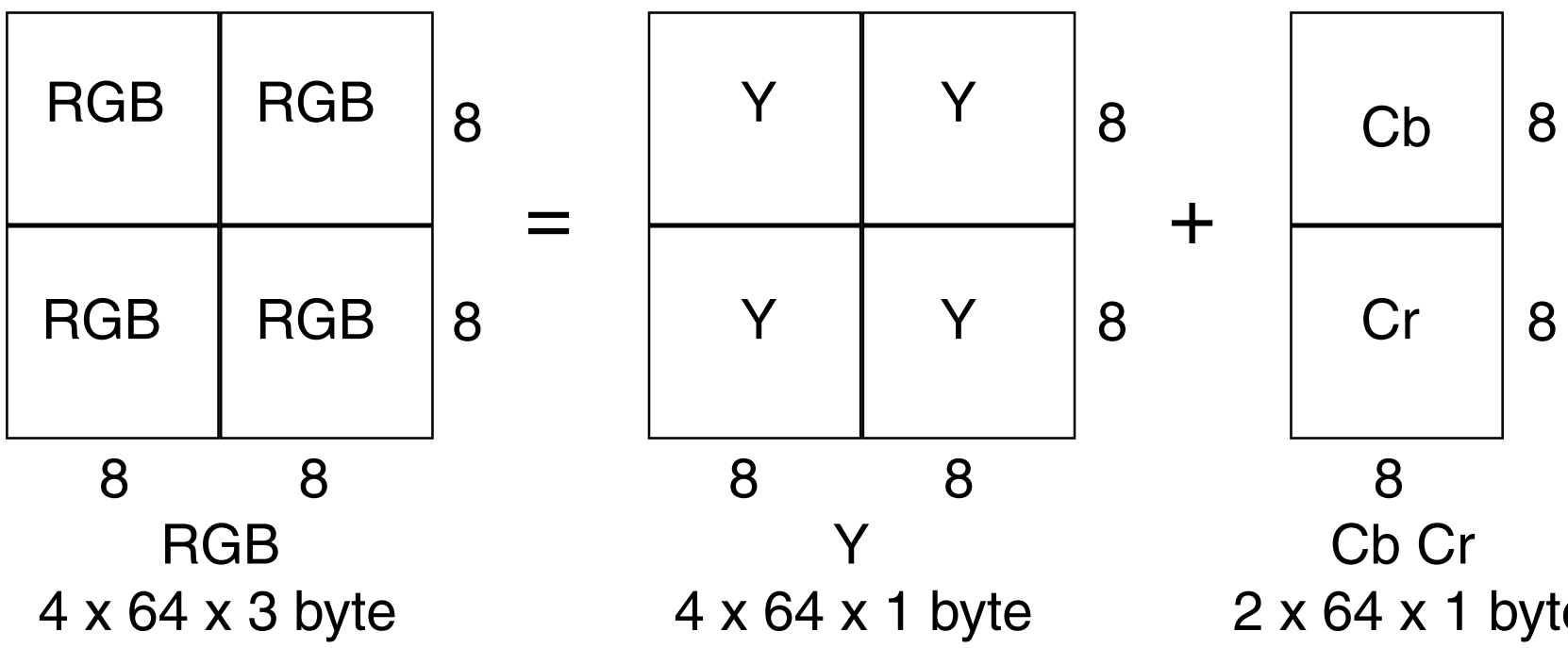
JPEG Compression

The image is divided in 16x16 pixel boxes, each is subdivided in 8x8 boxes.



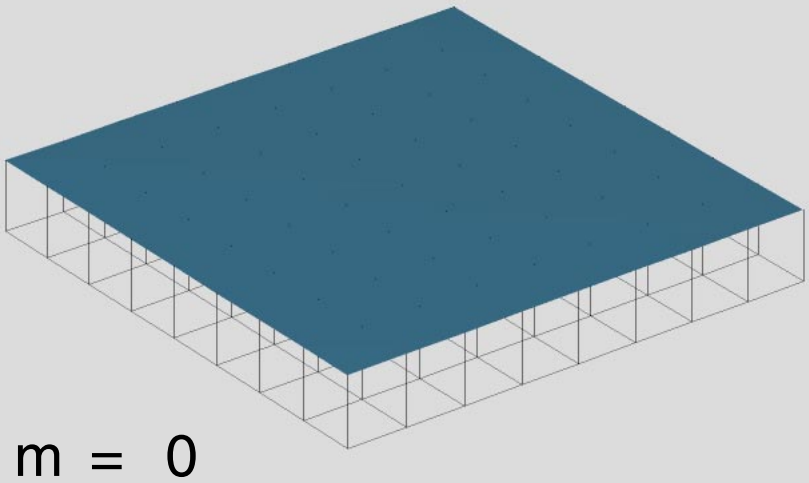
The Luminance Y is calculated for each 8x8 box, but the chrominance Cb,Cr is taken from each second row and each second column only, in fact by averaging four values for one result. This is already a compression ratio of two.

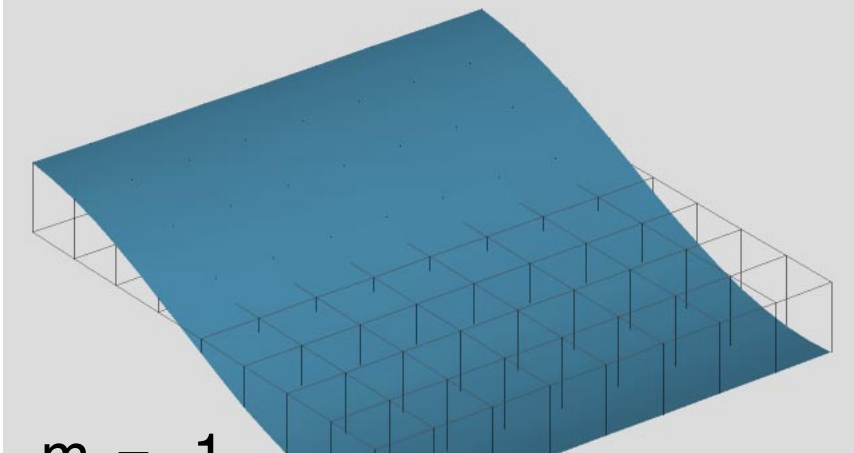
$$
\begin{aligned}
Y &= 0.299\,R + 0.587\,G + 0.114\,B \\
Cb &= -0.169\,R - 0.331\,G + 0.500\,B \\
Cr &= 0.500\,R - 0.419\,G - 0.081\,B
\end{aligned}
$$

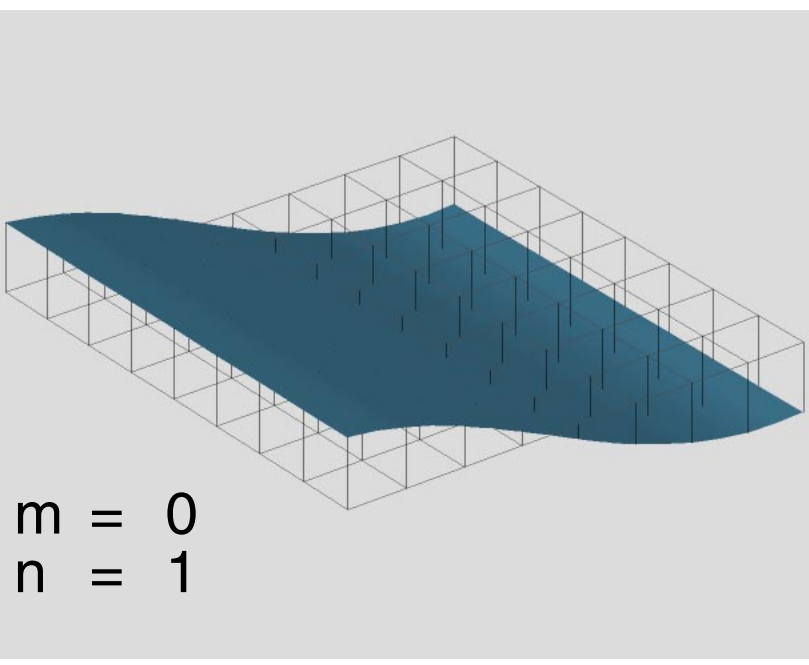
Y is a substitute for the Luminance (based on inverse gamma-encoded data), but the residual colors Cb and Cr do not have any physical meaning.
Then the content of each 8x8 box, which we call here C_{xy}, altogether 64 values C_{00} to C_{77}, is treated by the same mathematical algorithm, the Discrete Cosine Transformation DCT.
On the next page we see a subset of 64 functions F_{mn}. F_{00} is a constant function, F_{10} has a half-cosine slope in x-direction and F_{01} in y-direction. F_{77} is the most complex function, 4 cosine periods in each direction.
Then it has to be calculated how much each function contributes to the discrete landscape C_{xy}. Result is a set of 64 coefficients K_{mn}.
K_{00} is the DC- or mean value. K_{77} the highest frequency content.

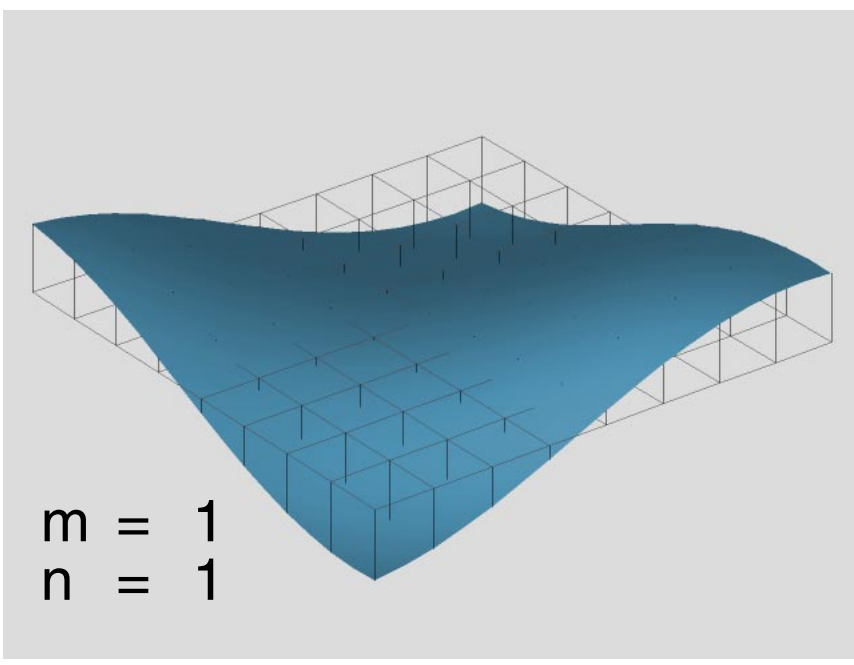
The DCT is lossless and reversible, besides some minor round-off errors because the coefficients K_{mn} are stored ShortInt -128 to +127.
In any continuous tone image the high frequencies are rare, with a few exceptions here and there.

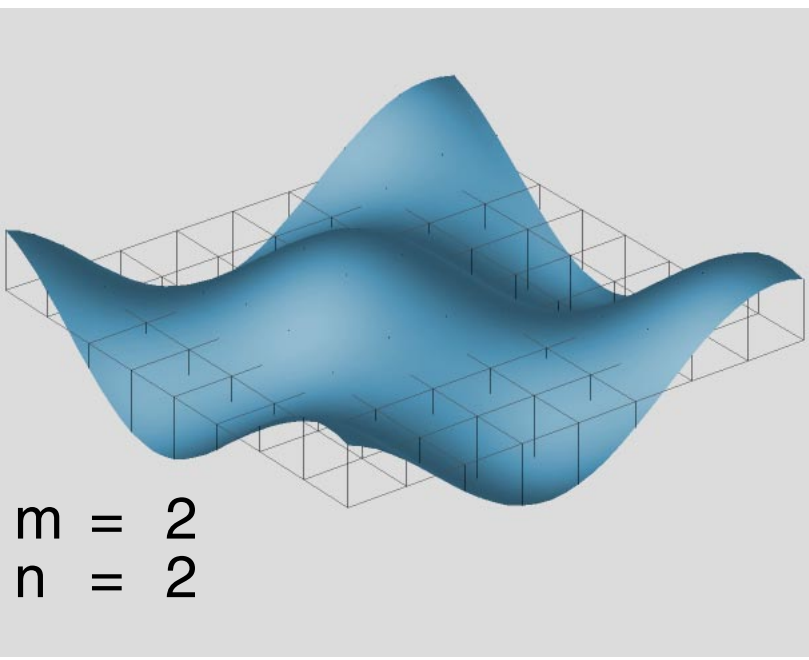
Discrete Cosine Transformation / Zoom 200%

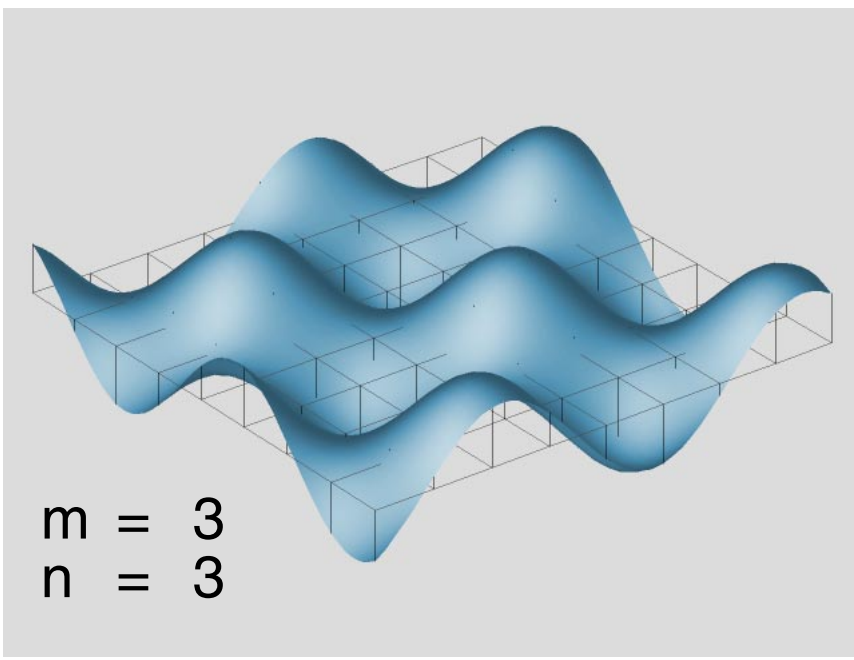


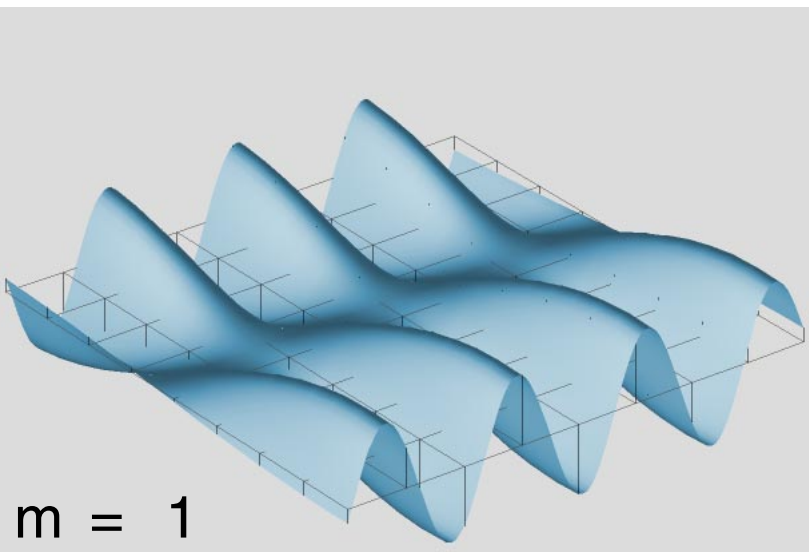


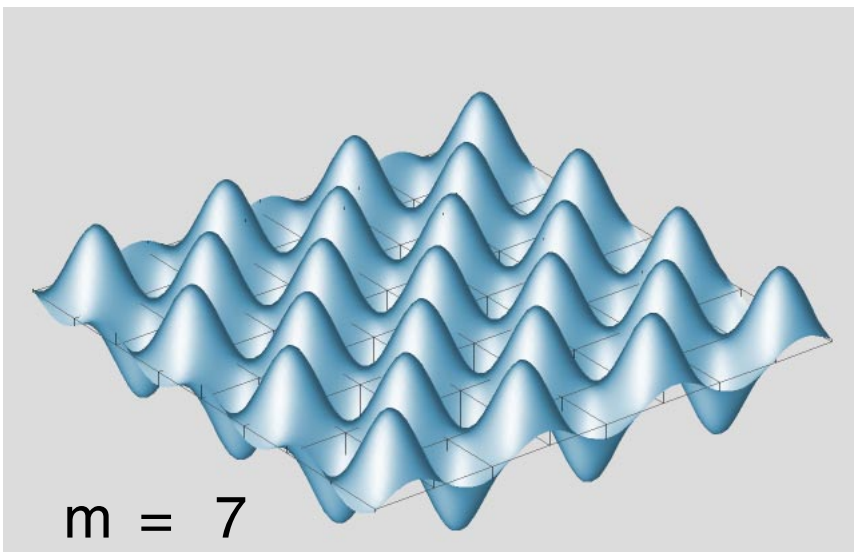












This is a table of 64 DCT coefficients, in ShortInt -128...+127 .

The low frequency part is top left, the high frequency part bottom right.
A linear fading or shaded color bar in vertical direction has mainly coefficients K_{00} and K_{01}.
Minor corrections are necessary because the linear slope needs to be replaced by cosines.

K_{00}	K_{10}	K_{20}	K_{30}	K_{40}	K_{50}	K_{60}	K_{70}
K_{01}	K_{11}	K_{21}	K_{31}	K_{41}	K_{51}	K_{61}	K_{71}
K_{02}	K_{12}	K_{22}	K_{32}	K_{42}	K_{52}	K_{62}	K_{72}
K_{03}	K_{13}	K_{23}	K_{33}	K_{43}	K_{53}	K_{63}	K_{73}
K_{04}	K_{14}	K_{24}	K_{34}	K_{44}	K_{54}	K_{64}	K_{74}
K_{05}	K_{15}	K_{25}	K_{35}	K_{45}	K_{55}	K_{65}	K_{75}
K_{06}	K_{16}	K_{26}	K_{36}	K_{46}	K_{56}	K_{66}	K_{76}
K_{07}	K_{17}	K_{27}	K_{37}	K_{47}	K_{57}	K_{67}	K_{77}

DCT - Coefficients

Then each coefficient is divided by values from the so called Quantization Tables.
These are different for Y and Cb, Cr.
Because the results are always rounded off to ShortInt, we get many small numbers and plenty zeros in the results, which replace the original DCT coefficients.
A slightly disturbed symmetry can prevent from generating patterns.

16	11	10					61
12	12	14					55
14	13	16					56
72	92	95					99

Y - Quantization Table

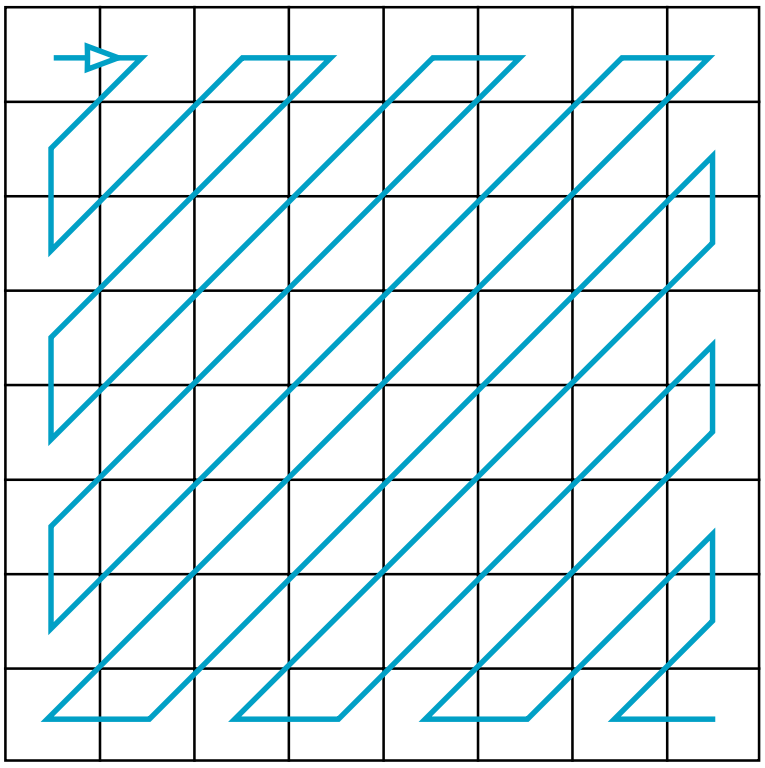
Especially the high frequency part bottom right consists mainly of zeros.
The Quantization Values are larger for Cb,Cr than for Y, thus the color part is treated once again very badly.
The tables were probably found by experiments by the JPEG experts.

Finally, each 8x8 box contains only a few significant small numbers. This is the lossy compression.

17	18	24	47	99			99
18	21	26	66	99			99
24	26	56	99				99
47	66	99					
99	99						
99	99	99					99

CbCr - Quantization Table

The divided DCT coefficients are scanned in ZigZag order.
First the low frequency part, finally the high frequency part.
It can be expected, that here are many sequential zeros.



ZigZag Table

The sequential zeros are lossless compressed by Runlength Encoding RLE. For example 0,0,0,0,0 will deliver 0,5 , thus a much shorter Code.

Then we have to take into account, that additionally many small numbers appear, e.g. -2, -1, +1, +2 . The greatest number is approximately 128/16= 8.

Now the lossless Huffman Encoding is applied. Small numbers are coded by few bits, large number by more bits, instead of all numbers by one Byte. This is controlled by a Huffman Table. Huffman and Runlength together pack the divided DCT coefficients highly compressed but lossless.
Of course it is a rather complex task to apply Huffman because a sequential bitstream needs to packed by number codes of different bitlengths.

The receiver finds in the File Header two Quantization Tables and two Huffman Tables. The Color Matrix is a standard, the ZigZag Table as well.

The Decoding works exactly in opposite direction:
Decode Huffman Bitstream. Unpack Anti-ZigZag. Multiply by Quantization. Apply Inverse DCT.
Repeat this until six 8x8 blocks for one 16x16 RGB block are found.
Apply color doubling for rows and columns. Apply Inverse Color Matrix. Assemble 16x16 RGB block. Draw block.
The compression ratio is controlled by the Quantization Tables. Multiplying the tables by 0.5 will improve the quality but increase the filesize. Multiplying by 2 will do just the opposite. Appropriate factors are found by experiments.

The Compression Ratio is the quotient
C = UncompressedFileSize / CompressedFileSize

C = 10
Excellent quality; compression cannot be detected by sharpeners
C = 20...30
Good quality; compression can be detected by sharpeners
C = 50
Bad quality; boxes visible

Each JPEG File needs at least 0.5 kByte for the header and the tables in the author´s program. In PhotoShop the minimum size is eventually 4kByte, if additional information is embedded (preview, color management).
For Web applications it is recommended to avoid additional information by 'Save for Web' in Photoshop.

JPEG compression tends to show softening effects. Nevertheless single sharp lines are not ignored. Sharpening in advance to the compression can improve the quality, but the file size will be increased because high frequency parts are now essential.

Though JPEG is based on TrueColor, standard Encoders and Decoders can be used for grayscale images. In a color image,the gray part may anyway consume 75% of the file size, because of lower compression.

This report by Gernot Hoffmann is mainly based on the Diploma Thesis of Hermann Hildebrandt (1999) and the cooperation with him.
Later, the author had replaced all kernels by Assembler procedures and functions, even for the floating point operations, mainly the DCT.
All tests for encoding and decoding were performed using PhotoShop and some other Windows programs.
The JPEG package is a part of the non-commercial image processing program ZEBRA. The illustration was generated by the experimental Computer Graphics System ZEFIR.

Gernot Hoffmann
September 18 / 2003

Website