Multimedia Coding and Transmission

Image Coding

Ifi, UiO

Norsk Regnesentral

Vårsemester 2005

Wolfgang Leister

This part of the course ...

- ... is held at Ifi, UiO ...
  (Wolfgang Leister)
- … and at University College Karlsruhe
  (Peter Oel, Clemens Knoerzer)

The story so far ...

- Data compression
  - information theory
  - run length encoding
  - Huffman coding
  - Zif-Lempel(-Welch) algorithm
  - Arithmetic coding

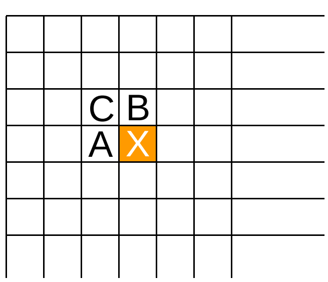
JPEG

- Joint Photographic Expert Group
- 1991 - 1993
- CCITT, ISO 10918
- Lossless coding (Comp. 2:1)
- Coding with loss (10:1-40:1)
- Parameter controls image quality
- Not limited for certain image types

JPEG

- Discrete Cosine Transformation (DCT)
- Huffman- or Arithmetic Coding
- Modes:
  - Lossless Coding
  - Sequential Coding
  - Progressive Coding
  - Hierarchical Coding
- Not a file format !!! → JFIFF

Lossless Coding

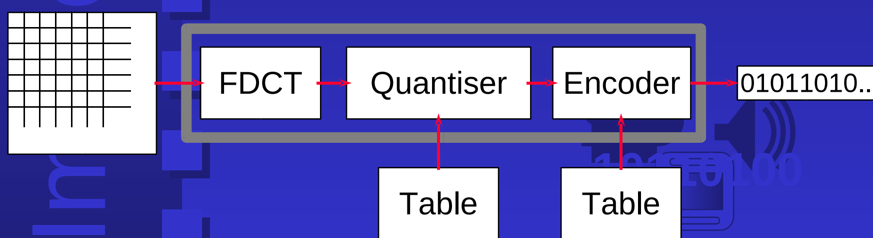


Nr.	Prediction
0	-
1	A
2	B
3	C
4	A+B-C
5	A+(B-C)/2
6	B+(A-C)/2
7	(A+B)/2

(Vhs. Diff.)(Vhs. Diff.)(Vhs. Diff)...

Lossy Coding

- Subdivision in 8x8 Blocks
- Transformation in frequency space
- Quantising
- Coding (Huffman or arithmetic Coding)

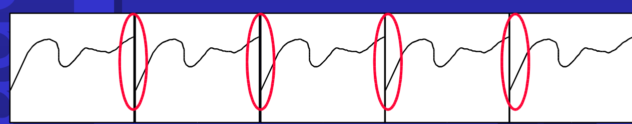


Why DCT?

- Why use frequency domain?
  - better statistic distribution
  - many low frequency parts
  - few high frequent parts
  - quantising better possible
  - Humans see high frequencies only for high contrast values

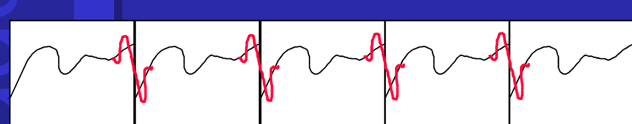
Why DCT?

- Why not Fourier Transform?
  - 8x8 Blocks
  - FT: ringing at block edges



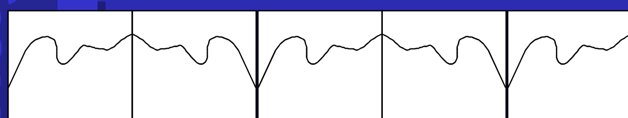
Why DCT?

- Why not Fourier Transform?
  - 8x8 Blocks
  - FT: ringing at block edges

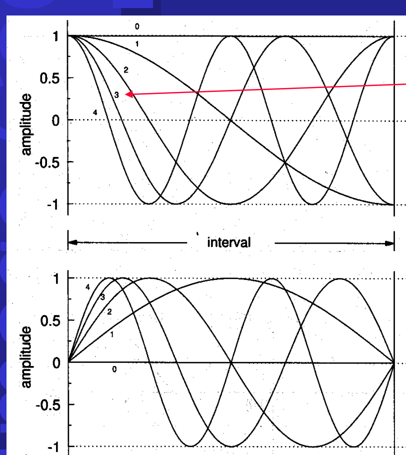


Why DCT?

- Why not Fourier Transform?
  - 8x8 Blocks
  - FT: ringing at block edges
  - Mirroring produces even function
  - Sinus coefficients disappear

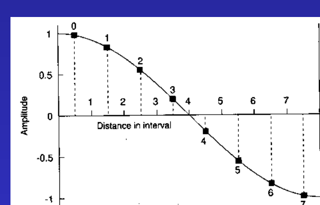


DCT



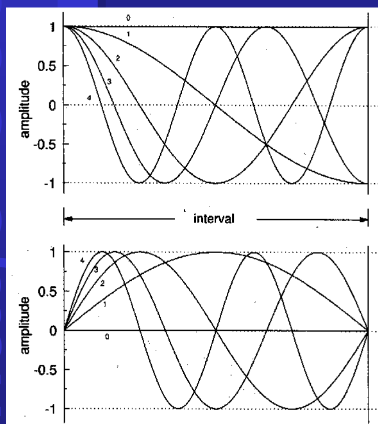
cosine, sine functions

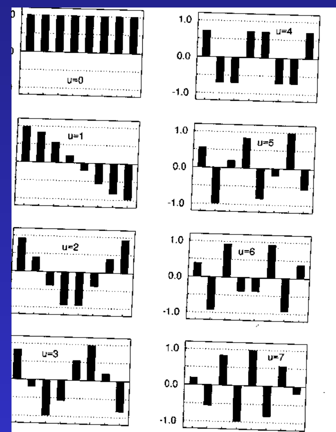
$f(t) = \cos(u \cdot t)$



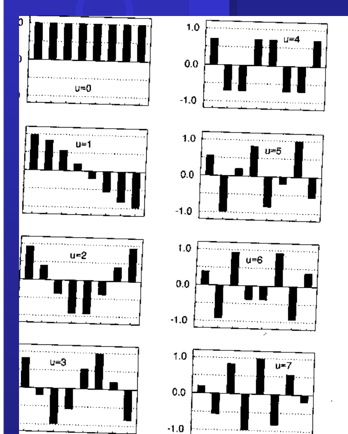
sampling of function

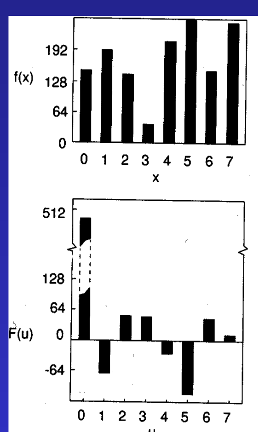
DCT - basis functions

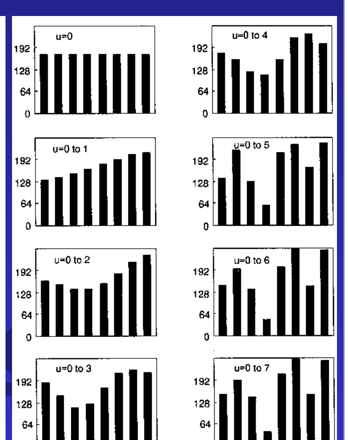




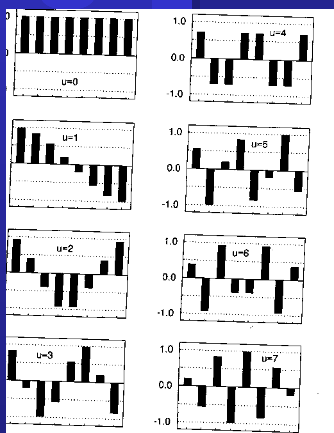
DCT - example

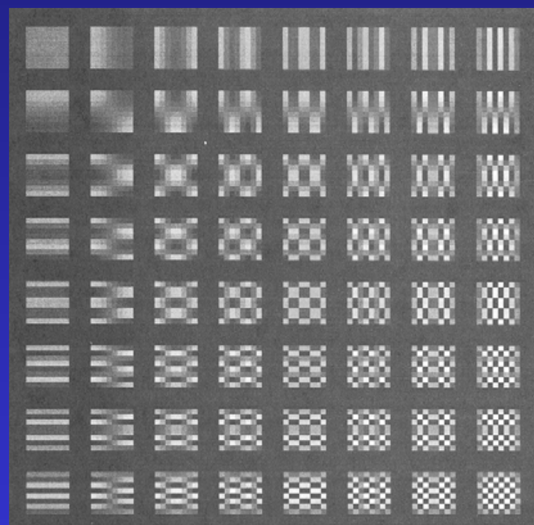






DCT 1D - 2D





Inverse DCT (decoding)

pixels

coefficients

$$f(x,y) = \frac{1}{4}\left[\sum_{u=0}^{7}\sum_{v=0}^{7} C(u)C(v)F(u,v) * \cos\frac{(2x+1)u}{16}\cos\frac{(2y+1)v}{16}\right]$$

for $C(u), C(v) = \frac{1}{\sqrt{2}}$ when $u, v = 0$

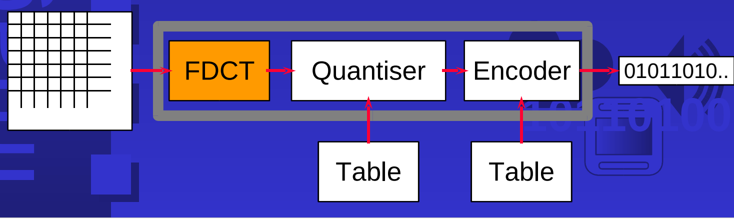
$C(u), C(v) = 1$ else

(Forward)DCT

- unsigned → signed
- FDCT: $F(u,v) = \frac{1}{4} C(u)C(v) \left[\sum_{x=0}^{7} \sum_{y=0}^{7} f(x,y) * \cos\frac{(2x+1)u}{16} \cos\frac{(2y+1)v}{16} \right]$

$$\text{with } C(u), C(v) = \frac{1}{\sqrt{2}} \text{ for } u,v = 0$$

$$C(u), C(v) = 1 \text{ else}$$



(Forward)DCT

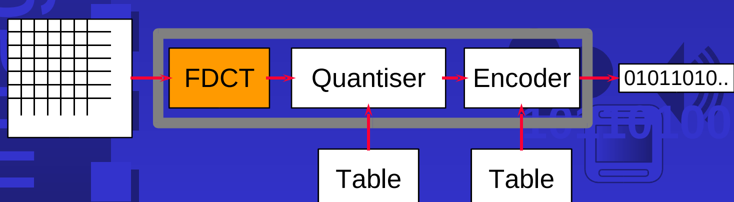
139	144	149	153	155	155	155	155
144	151	153	156	159	156	156	156
150	155	160	163	158	156	156	156
159	161	162	160	160	159	159	159
159	160	161	162	162	155	155	155
161	161	161	161	160	157	157	157
162	162	161	163	162	157	157	157
162	162	161	161	163	158	158	158

unsigned → signed

11	16	21	25	27	27	27	27
16	23	25	28	31	28	28	28
22	27	32	35	30	28	28	28
31	33	34	32	32	31	31	31
31	32	33	34	34	27	27	27
33	33	33	33	32	29	29	29
34	34	33	35	34	29	29	29
34	34	33	33	35	30	30	30

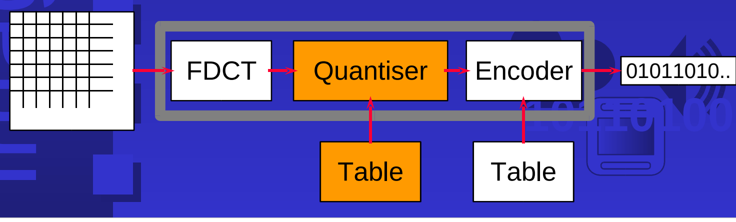
→ FDCT

235.6	-1.0	-12.1	-5.2	2.1	-1.7	-2.7	1.3
-22.6	-17.5	-6.2	-3.2	-2.9	-0.1	0.4	-1.2
-10.9	-9.3	-1.6	1.5	0.2	-0.9	-0.6	-0.1
-7.1	-1.9	0.2	1.5	0.9	-0.1	0.0	0.3
-0.6	-0.8	1.5	1.6	-0.1	-0.7	0.6	1.3
1.8	-0.2	1.6	-0.3	-0.8	1.5	1.0	-1.0
-1.3	-0.4	-0.3	-1.5	-0.5	1.7	1.1	-0.8
-2.6	1.6	-3.8	-1.8	1.9	1.2	-0.6	-0.4



Quantising

$$F^{Q}(u,v) = \text{Integer}\left(\frac{F(u,v)}{Q(u,v)}\right)$$



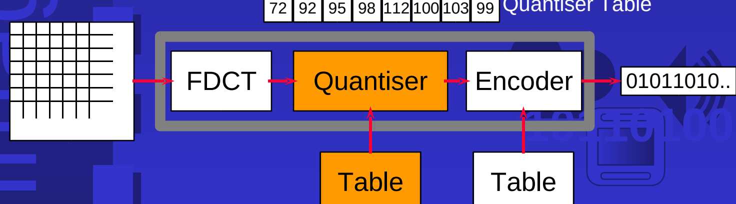
Quantising

235.6	-1.0	-12.1	-5.2	2.1	-1.7	-2.7	1.3
-22.6	-17.5	-6.2	-3.2	-2.9	-0.1	0.4	-1.2
-10.9	-9.3	-1.6	1.5	0.2	-0.9	-0.6	-0.1
-7.1	-1.9	0.2	1.5	0.9	-0.1	0.0	0.3
-0.6	-0.8	1.5	1.6	-0.1	-0.7	0.6	1.3
1.8	-0.2	1.6	-0.3	-0.8	1.5	1.0	-1.0
-1.3	-0.4	-0.3	-1.5	-0.5	1.7	1.1	-0.8
-2.6	1.6	-3.8	-1.8	1.9	1.2	-0.6	-0.4

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Quantiser Table

15	0	-1	0	0	0	0	0
-2	-1	0	0	0	0	0	0
-1	-1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0



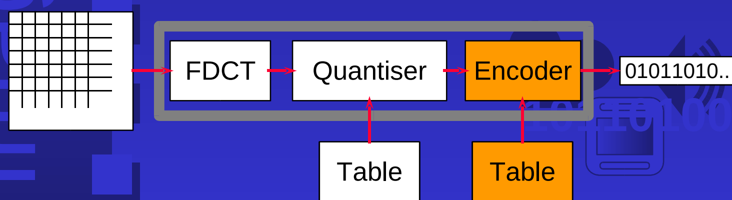
Coding

- 1 DC-Coefficient
- 63 AC-Coefficientes

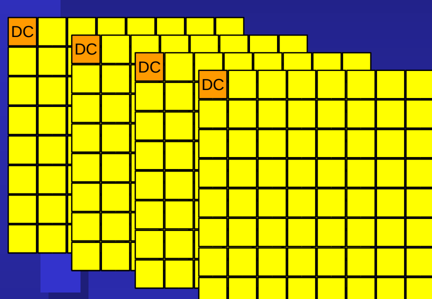
DC

15	0	-1	0	0	0	0	0
-2	-1	0	0	0	0	0	0
-1	-1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

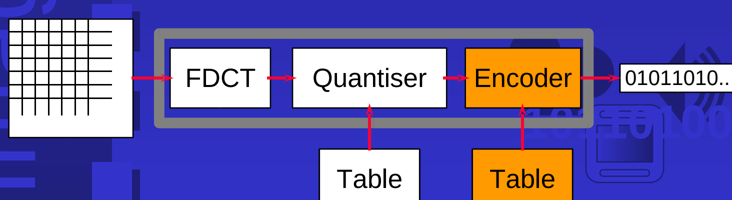
AC



DC-Coding



$$\Delta DC_i = DC_i - DC_{i-1}$$



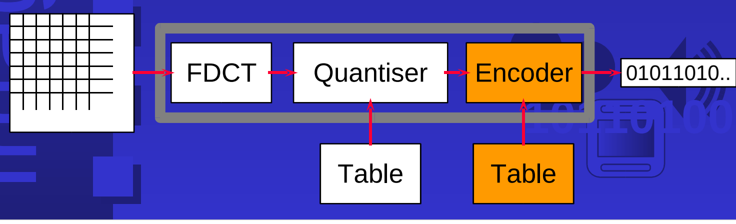
DC-Coding

$\Delta DC_i = DC_i - DC_{i-1}$

ΔDC_i=(Length, Value)

Length=Huffman-Coded

Value=(Sign, abs Value)



DC-Coding

Length	Value
0	0
1	-1, 1
2	-3,-2, 2,3
3	-7..-4, 4..7
4	-15..-8, 8..15
5	-31..-16, 16..31
6	-63..-32, 32..63
7	-127..-64, 64..127
8	-255..-128, 128..255
9	-511..-256, 256..511
10	-1023..-512, 512..1023
11	-2047..-1024, 1024..2047

Examples:

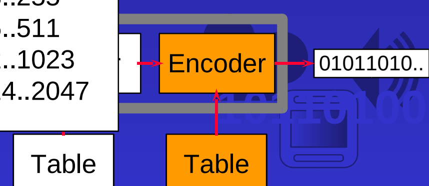
-3 : (2).1.1

-1 : (1).1.

0 : (0)..

5 : (3).0.01

67 : (7).0.000011

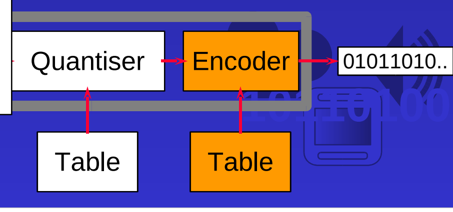


DC-Coding

Length	Code
0	00
1	010
2	011
3	100
4	101
5	110
6	1110
7	11110
8	111110
9	1111110
10	11111110
11	111111110

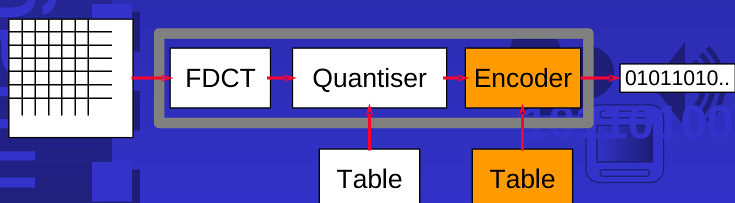
Examples:

-3	: (2).1.1	: 011.1.1
-1	: (1).1.	: 010.1.
0	: (0)..	: 00..
5	: (3).0.01	: 100.0.01
67	: (7).0.000011	: 11110.0.000011

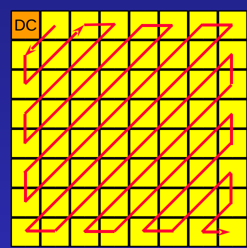


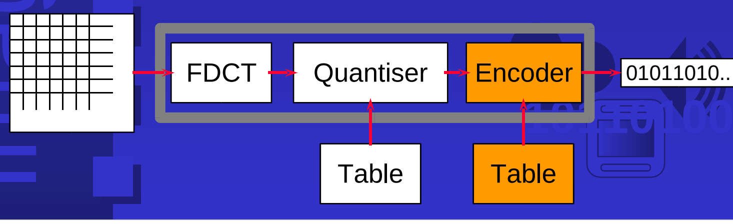
AC-Coding

- Zig-Zag Serialising
- Zero run length
- Huffman-Coding



Zig-Zag Serialising

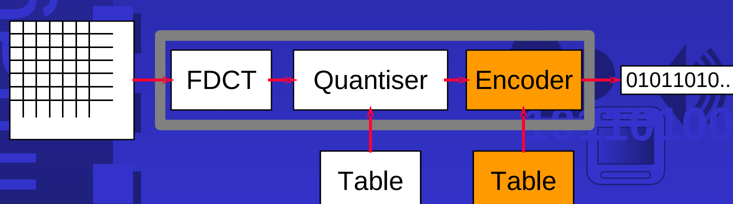


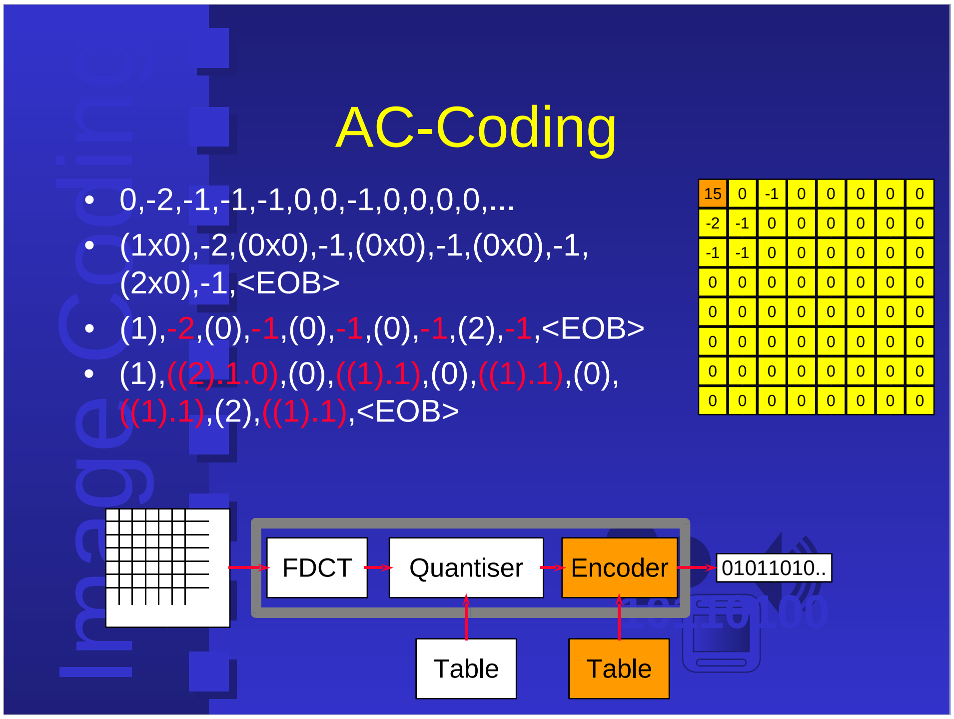


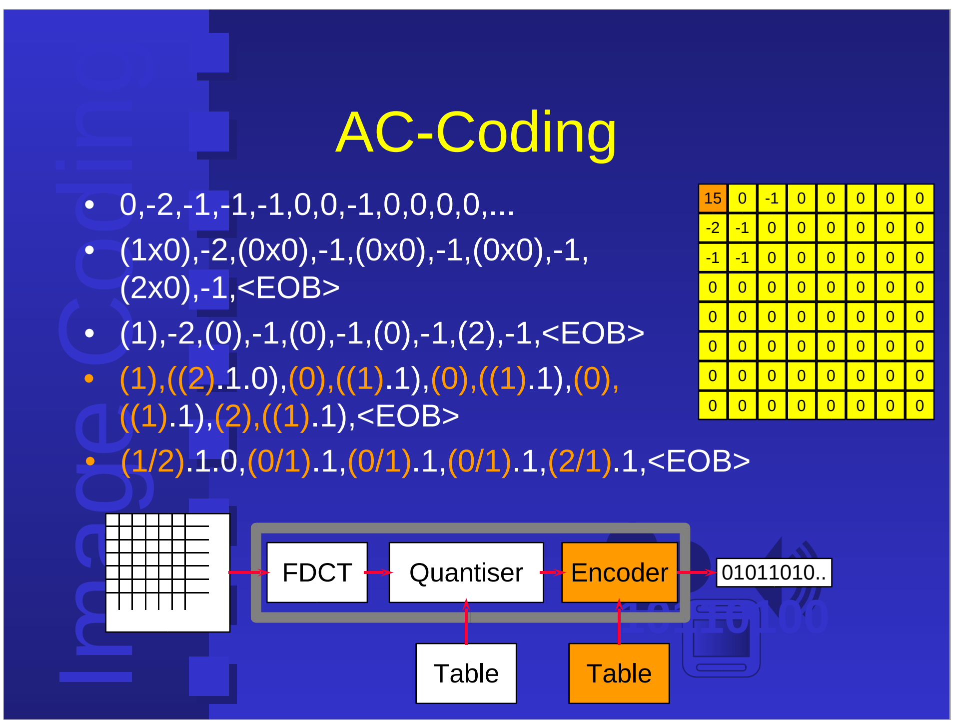
AC-Coding

- 0,-2,-1,-1,-1,0,0,-1,0,0,0,0,...
- (1x0),-2,(0x0),-1,(0x0),-1,(0x0),-1,(2x0),-1,<EOB>
- (1),-2,(0),-1,(0),-1,(0),-1,(2),-1,<EOB>

15	0	-1	0	0	0	0	0
-2	-1	0	0	0	0	0	0
-1	-1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0




AC-Coding
0,-2,-1,-1,-1,0,0,-1,0,0,0,0,...
(1x0),-2,(0x0),-1,(0x0),-1,(0x0),-1,
(2x0),-1,<EOB>
(1),-2,(0),-1,(0),-1,(0),-1,(2),-1,<EOB>
(1),((2).1.0),(0),((1).1),(0),((1).1),(0),
((1).1),(2),((1).1),<EOB>
15 0 -1 0 0 0 0 0
-2 -1 0 0 0 0 0 0
-1 -1 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
FDCT
Quantiser
Encoder
01011010..
Table
Table

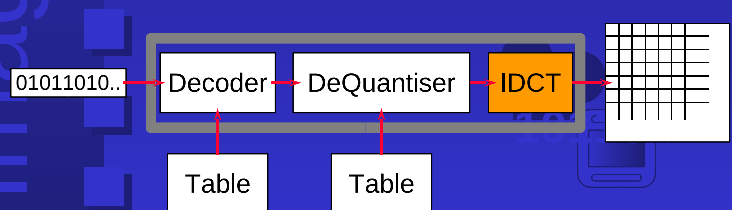

AC-Coding
0,-2,-1,-1,-1,0,0,-1,0,0,0,0,...
(1x0),-2,(0x0),-1,(0x0),-1,(0x0),-1,
(2x0),-1,<EOB>
(1),-2,(0),-1,(0),-1,(0),-1,(2),-1,<EOB>
(1),((2).1.0),(0),((1).1),(0),((1).1),(0),
((1).1),(2),((1).1),<EOB>
(1/2).1.0,(0/1).1,(0/1).1,(0/1).1,(2/1).1,<EOB>
15 0 -1 0 0 0 0 0
-2 -1 0 0 0 0 0 0
-1 -1 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
FDCT
Quantiser
Encoder
01011010..
Table
Table

<EOB>	1010	1/1	1100
0/1	00	1/2	11011
0/2	01	....	
0/3	100	2/1	11100
0/4	1011	2/2	11111001
0/5	11010	....	
0/6	1111000	15/1	...
0/7	11111000	...	
...		15/10	...
0/10	.....	<ZR16>	11111111001

- 0,-2,-1,-1,-1,(
- (1x0),-2,(0x0)
  (2x0),-1,<EO
- (1),-2,(0),-1,(
- (1),((2).1.0),(
  ((1).1),(2),((1)
- (1/2).1.0,(0/1).1,(0/1).1,(0/1).1,(2/1).1,<EOB>
- 11011 1 0 00 1 00 1 00 1 11100 1 1010
- 63 AC-Coefficients ⟶ 26 Bit

Decompression

- DeCoding
- Rescale by DeQuantising
- Inverse DCT



Inverse DCT

- IDCT:

$$f(x,y) = \frac{1}{4}\left[\sum_{u=0}^{7}\sum_{v=0}^{7} C(u)C(v)F(u,v) * \cos\frac{(2x+1)u}{16}\cos\frac{(2y+1)v}{16}\right]$$

for $C(u),C(v) = \frac{1}{\sqrt{2}}$ when $u,v = 0$

$C(u),C(v) = 1$ else

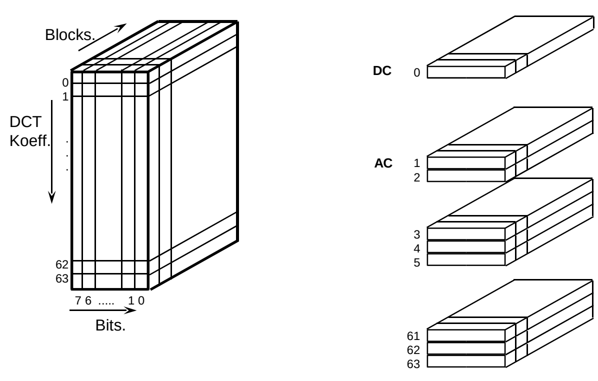
Compression Results

n:1	Quality
30 - 20	usable - good
20 - 10	good - very good
10 - 5	excellent
5 - 4	not distinguishable from original

Progressive Mode

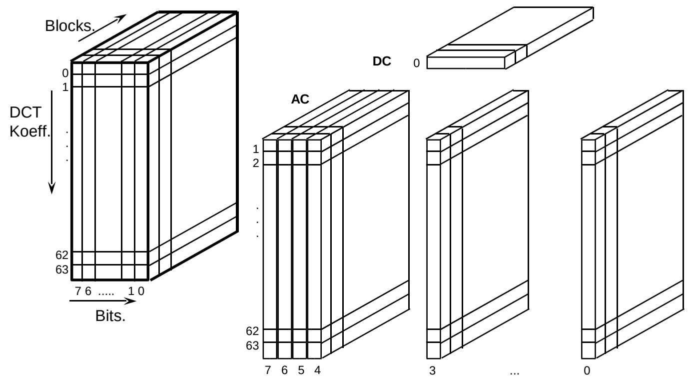
- Transfer coefficients partially in several runs.
- Two Possibilities:
  - Spectral Transfer
  - Approximated Transfer

Progressive Mode



Spectral Transfer

Progressiver Modus



Approximated Transfer

Hierarchical Mode

- Code image with low resolution first
- Code higher resolution as difference to previous lower resolution
- Image is represented in several resolutions
- Unnecessry data are not transferred

Images with several channels

- JPEG can use several (Colour)-Channels (e.g., YC_bC_r)
- Channels can have different resolution
- Resolution factor as whole number
- JPEG method does not bother about channels

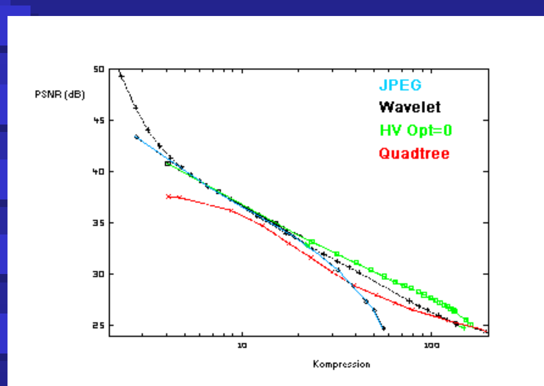
JFIFF

- JPEG defines algorithm only.
- JPEG is not a file format
- JPEG is colour-blind
- Parameters and tables are pre-defined
- Based on JPEG mode of TIFF 6.0
- Consists of segments which are defined by markers (like TIFF)

Is JPEG good enough?

- Visible blocks for high compression rates
- For compression rates > 40:1 JPEG does not work well.
- low frequencies are not taken into account.
- Wavelet Coding
  - Good quality up to 60:1
  - Linear degradation for higher compression rates
  - No visible blocks
  - JPEG 2000 Standard
- Fractal Coding

Comparison



Literature

- JPEG:
  - – Pennebaker,Mitchell: **JPEG, Still Image Data Compression Standard**, Van Nostrand Reinhold (1993)
- Wavelet-Coding:
  - – Daubechies: **Ten Lectures on Wavelets**, Society for Industrial and Applied Mathematics

Literature

- – Fournier: **Wavelets an their Applications in Computer Graphics**, SIGGRAPH `94 Course Notes
- Fractal Coding:
  - – Barnsley, Hurd: **Fractal Image Compression**, AK Peters Ltd (1993)
  - – Fisher: **Fractal Image Compression**, SIGGRAPH `92 Course Notes

Image Coding

The End of this Lecture