

DCT: Diskrete Kosinus-Transformation

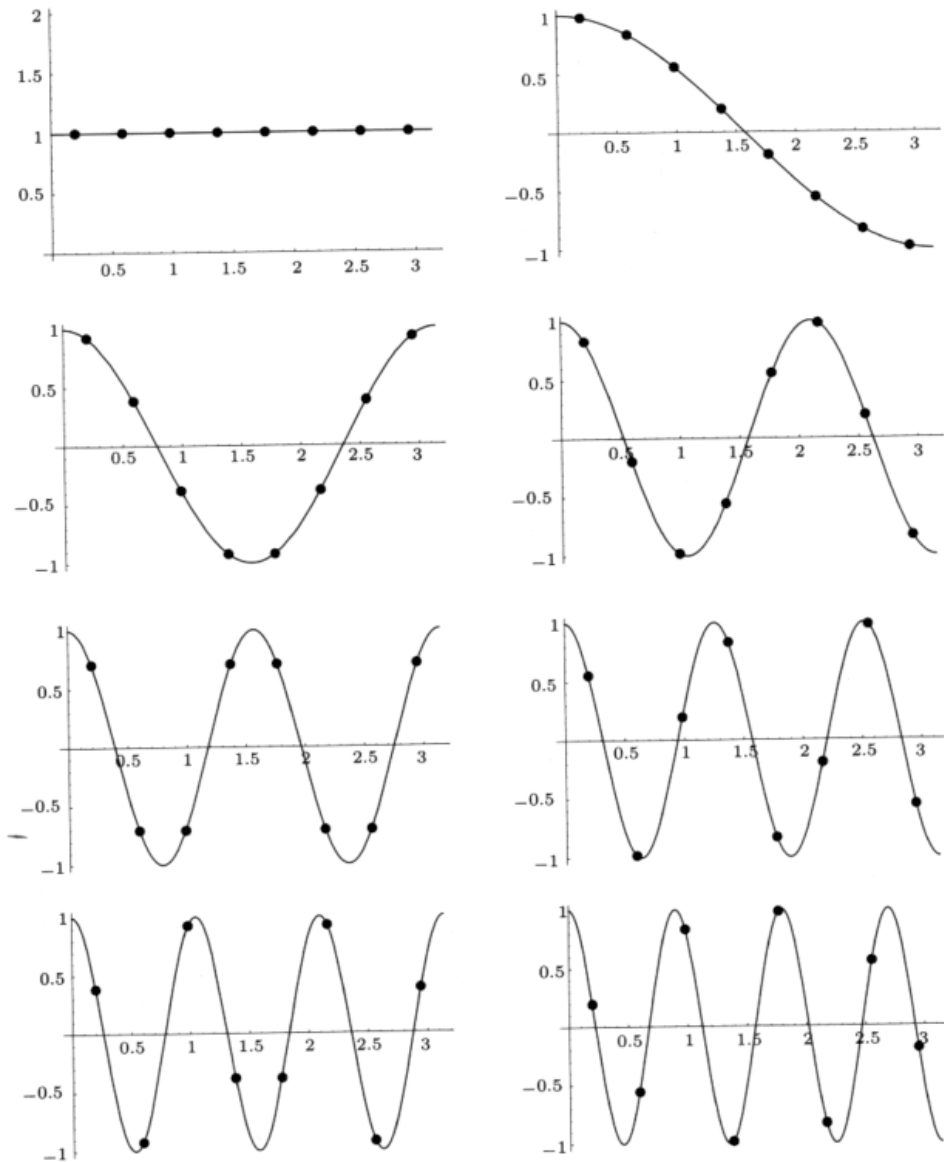


Figure 4.17: Calculating A One-Dimensional DCT.

Kosinusfunktionen für die 1D DCT:
zunehmende Frequenz
entsprechende Abtastpunkte

(Salomon)

DCT: 8x8 2D-Transformation

DCT:

$$G_{ij} = \frac{1}{4} C_i C_j \sum_{x=0}^7 \sum_{y=0}^7 p_{xy} \cos\left(\frac{(2x+1)i\pi}{16}\right) \cos\left(\frac{(2y+1)j\pi}{16}\right),$$

$$\text{where } C_f = \begin{cases} \frac{1}{\sqrt{2}}, & f = 0, \\ 1, & f > 0, \end{cases} \text{ and } 0 \leq i, j \leq 7.$$

IDCT:

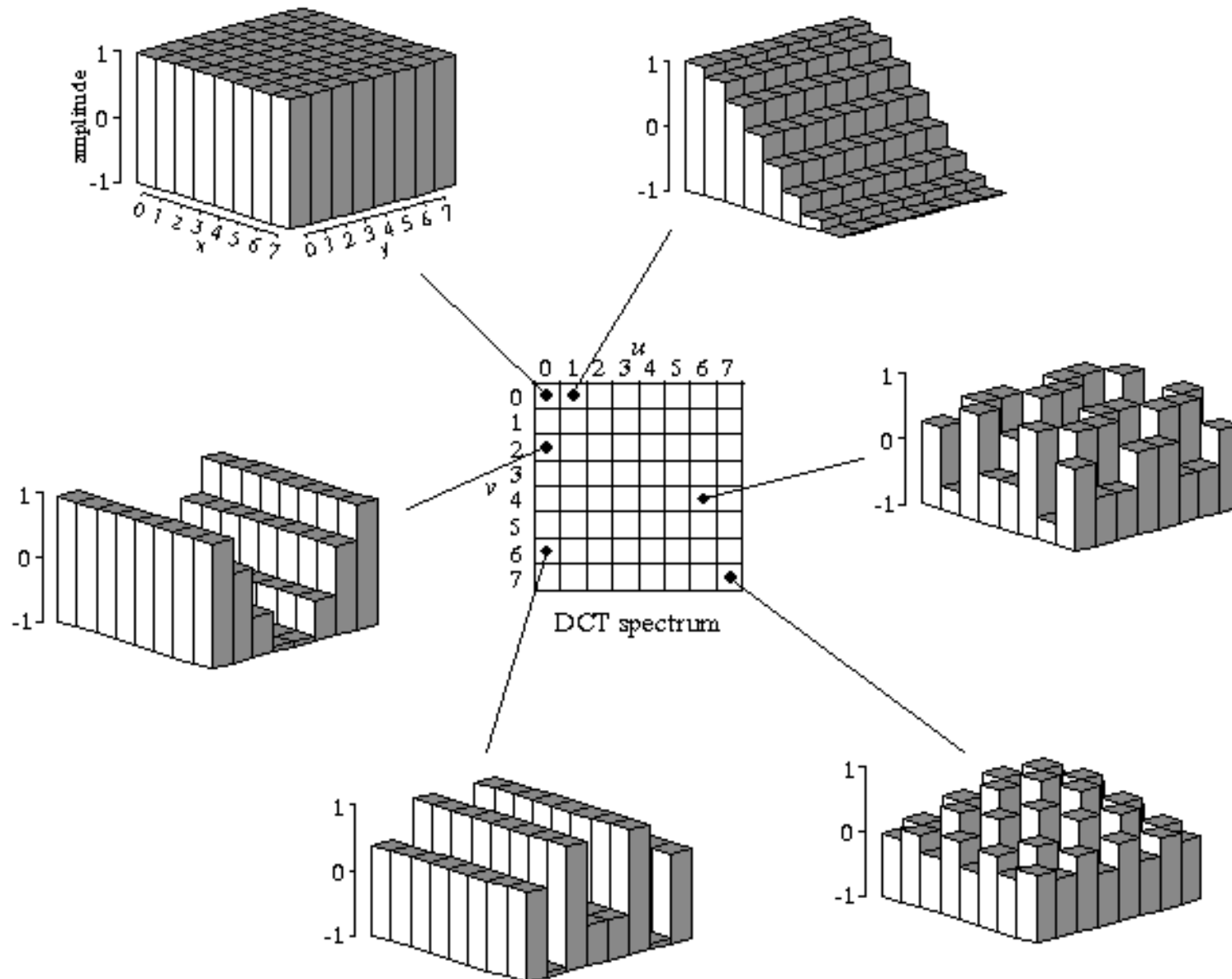
$$p_{xy} = \frac{1}{4} \sum_{i=0}^7 \sum_{j=0}^7 C_i C_j G_{ij} \cos\left(\frac{(2x+1)i\pi}{16}\right) \cos\left(\frac{(2y+1)j\pi}{16}\right),$$

$$\text{where } C_f = \begin{cases} \frac{1}{\sqrt{2}}, & f = 0; \\ 1, & f > 0. \end{cases}$$

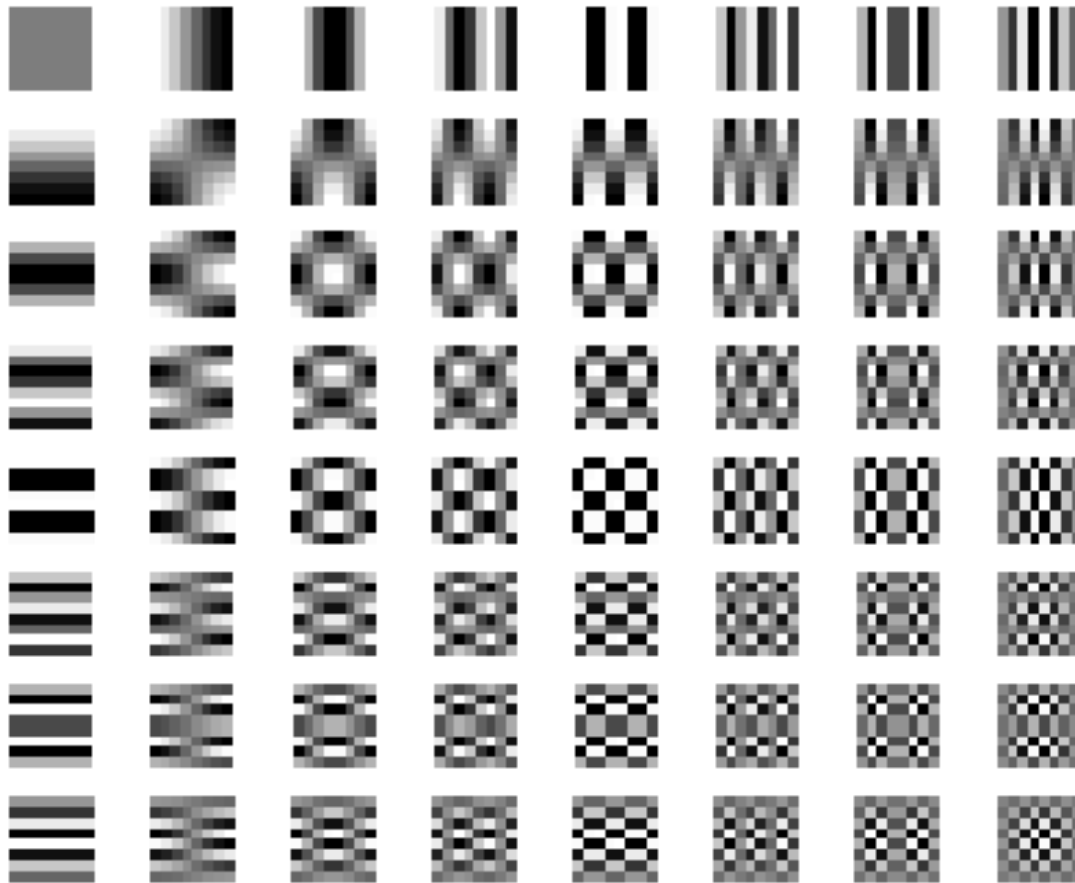
effiziente Implementierung?

- Vorberechnung der $\cos(x)$ Terme
- Kombination von 1D-Transformationen (min. Anzahl von Ops)
- zeilenweise Berechnung (Cache-freundlich)
- MMX, SSE, Spezialhardware

DCT: Basisfunktionen

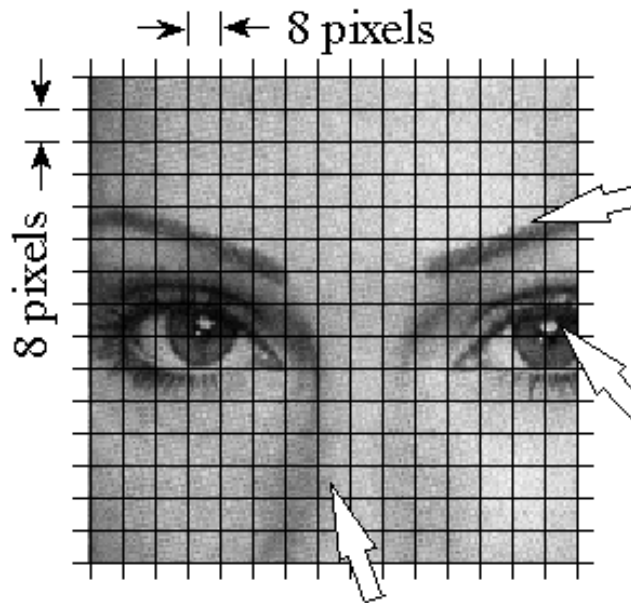


DCT: Basisfunktionen



- die Basisfunktionen der 2D 8x8 DCT
- entsprechende Bilder liefern max. Antwort der DCT
1 (weiß) .. 0 (grau) .. -1 (schwarz)

DCT: Beispiel



154	154	175	182	189	168	217	175
154	147	168	154	168	168	196	175
175	154	203	175	189	182	196	182
175	168	168	168	140	175	168	203
133	168	154	196	175	189	203	154
168	161	161	168	154	154	189	189
147	161	175	182	189	175	217	175
175	175	203	175	189	175	175	182

175	154	203	175	189	168	196	168
175	168	168	168	140	175	168	203
133	168	154	196	175	189	203	154
168	161	161	168	154	154	189	189
147	161	175	182	189	175	217	175
175	175	203	175	189	175	175	182

3	0	0	0	-1	9
-4	-6	-2	1	-1	4
-10	-3	-1	4	-10	-3
1	2	-2	0	0	-2
0	0	-2	0	0	-5
3	-1	3	-2	2	1
1	0	1	0	0	0
3	5	2	-2	3	0
4	3	0	4	3	3
4	-3	-13	3	-4	3
-5	3	-5	3	3	3

Spectrum

v spectrum

3	1	0	-3	1
4	-4	-4	5	-8
3	3	2	0	9
0	0	-5	-1	0
1	4	4	2	0
3	0	0	1	1
5	3	2	-1	-1
0	0	0	-4	0

ctrum

4	-2	-10	-1	0
3	7	13	1	3
8	-7	-6	5	-3
8	2	-1	0	-1
12	2	1	-1	4
1	-4	-1	5	6
4	3	2	-3	5
5	1	2	1	0

pectrum

3	-3	6	-3	4
2	0	3	1	2
0	0	0	-1	9
-4	-6	-2	1	-1
4	-10	-3	1	4
-1	2	-2	0	0
0	-2	0	0	-5
3	-1	3	-2	2
1	0	1	0	0
3	5	2	-2	3
0	4	3	3	3
4	-3	-13	3	-4
3	-5	3	3	3

Quantization Error

g. Using 10 bits

0	0	0	0	-1	0	0	0
-1	0	0	0	0	0	0	-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	1	0	0	0	-1	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

h. Using 8 bits

0	-3	-1	-1	1	0	0	-1
1	0	-1	-1	0	0	0	-1
-1	-2	1	0	-2	0	-2	-2
-1	-2	-1	2	0	2	0	1
0	-2	1	0	0	1	0	0
0	-4	-1	0	1	0	0	0
0	-2	0	1	-1	-1	1	-1
-1	-3	1	1	1	-3	-2	-1

i. Using 5 bits

-13	-7	1	4	0	0	10	-2
-22	6	-13	5	-5	2	-2	-13
-9	-15	0	-17	-8	8	12	25
-9	16	1	9	1	-5	-5	13
-20	-3	-13	-16	-19	-1	-4	-22
-11	6	-8	16	-9	-3	-7	6
-14	10	-9	4	-15	3	3	-4
-13	19	12	9	18	5	-5	10

(DSP Guide)

JPEG: Übersicht

JPEG := "Joint Picture Experts Group"

- Verfahren zur Kodierung von Grau- und Farbbildern
- 1991 standardisiert
- diverse Varianten, "baseline" oder "lossless"

Kodierung via DCT von Blöcken a 8x8 Pixeln

YCbCr Farbmodell (Farbsubsampling)

Kompressionsraten bis ca. 40:1 möglich

- JFIF: Bezeichnung für das Dateiformat (.jpg,.jpeg)
- JBIG: "Joint Bi-level image experts group" (s/w-Bilder)
- JPEG-2000: aktuelle, erweiterte Version

(www.jpeg.org)

JPEG: Konzepte, Ziele

sehr flexibles Format:

- fast beliebige Bildauflösung
- beliebiges Bild/Pixel-Aspektverhältnis
- unabhängig vom Farbmodell
- für alle Arten von Bildern, aber insb. für Photos

- progressive Kodierung (schnelle Vorschau)

- Implementierung in Software, bei Bedarf in Hardware:
- "Motion-JPEG", Anwendung für Video

JPEG: Beispiel



229 KByte, 3:1, 7 bpp



24 KByte, 32:1, 0.74 bpp



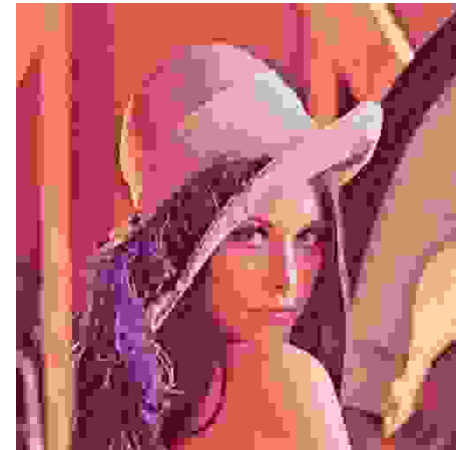
18 KByte, 43:1



12 KByte, 66:1, 0.36 bpp



10 KByte, 78:1

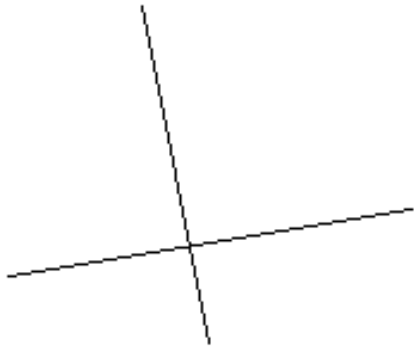


5 KByte, 140:1, 0.17 bpp

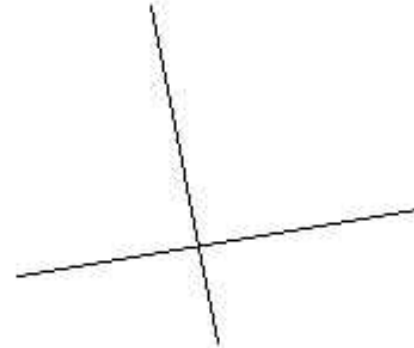
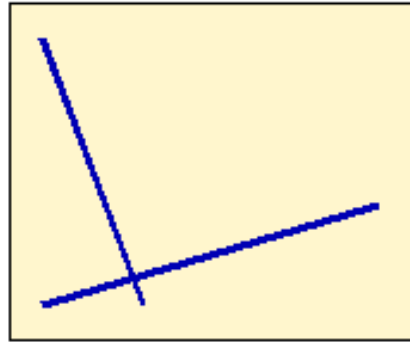
=> Vergleich: PPM-raw, 512x512x24: 786 KByte, 24 bpp

=> praktisch optimale Qualität bei 10:1, brauchbar bis ca. 40:1

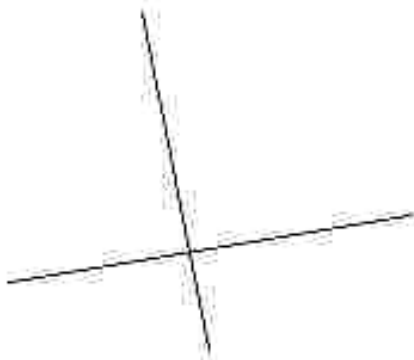
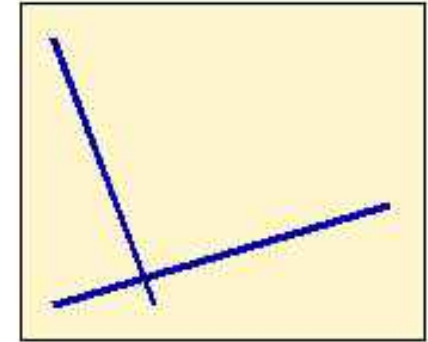
JPEG: "ringing"



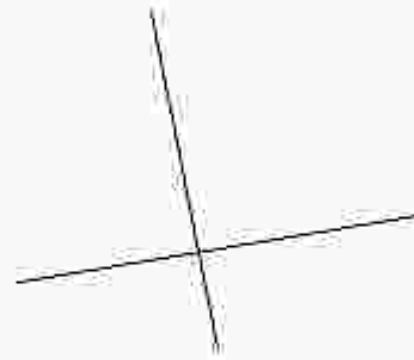
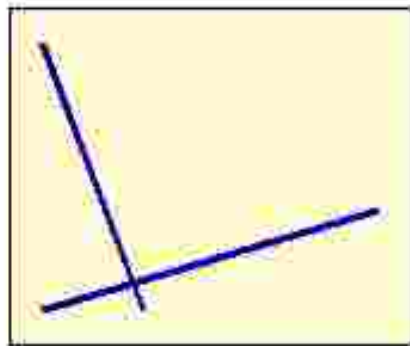
Original (GIF)



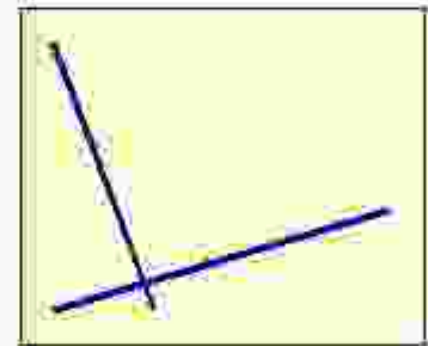
JPEG xv quality 40%



JPEG xv quality 10%



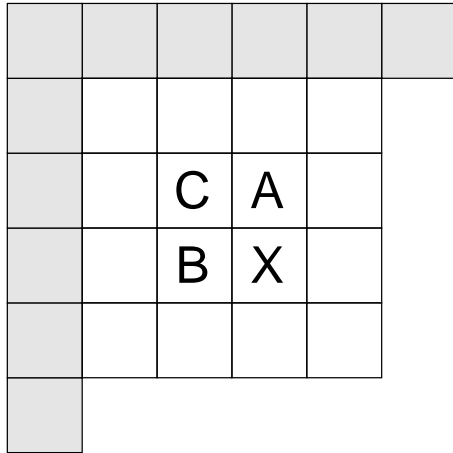
JPEG xv quality 5%



=> Überschwingen der Kosinusfunktionen bei harten Bildkontrasten

=> verstärkter Effekt bei starker Kompression wegen fehlender Koeffizienten

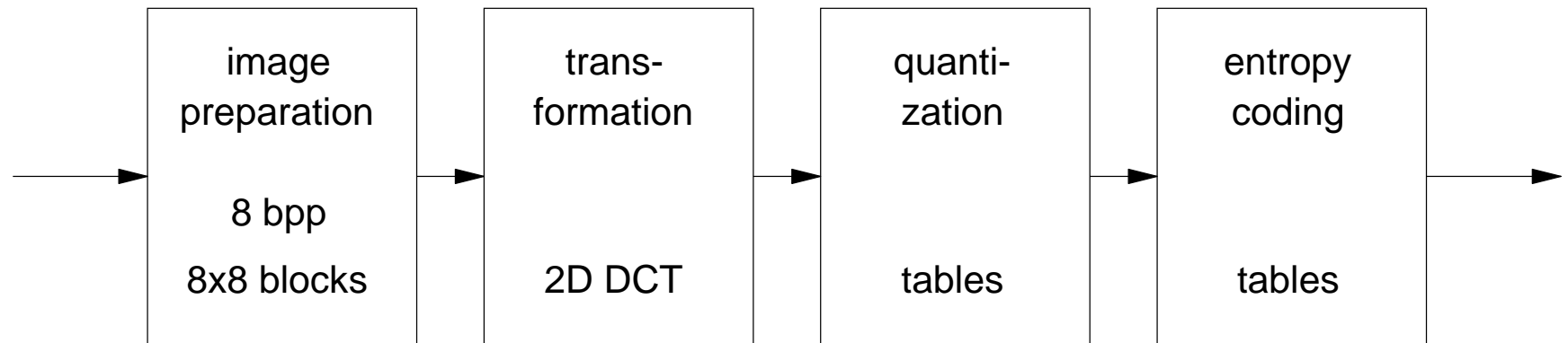
JPEG: verlustfreie Kodierung



Nr.	Vorhersage V:
0	-
1	A
2	B
3	C
4	$A+B-C$
5	$A+(B-C)/2$
6	$B+(A-C)/2$
7	$(A+B)/2$

- Vorhersage V von Pixelwerten aus Nachbarpixeln
- sieben verschiedene Algorithmen für V
- kodiert Index des Algorithmus und Differenz $(X-V)$
- anschliessend Huffman-Kodierung
- Kompressionsfaktor ca. 2

JPEG: Verarbeitungsschritte



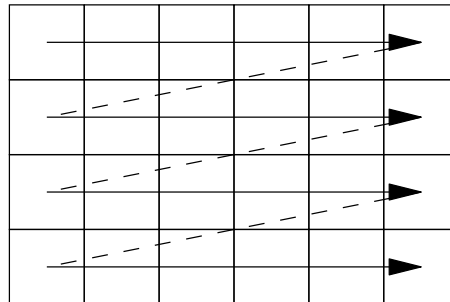
- Eingangsdaten in YUV-Darstellung und auf 8 bits/pixel wandeln
- Aufteilung in Blöcke, feste Größe 8x8
- 2D Kosinustransformation
- Quantisierung der Koeffizienten, feste (oder variable) Tabellen
- kombinierte Huffman-/Runlength-Kodierung

JPEG: Vorbereitung

verlustfreie Kodierung mit 2 .. 16 bpp

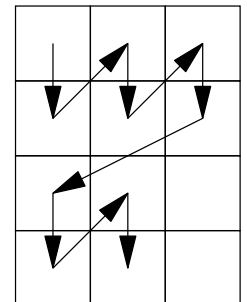
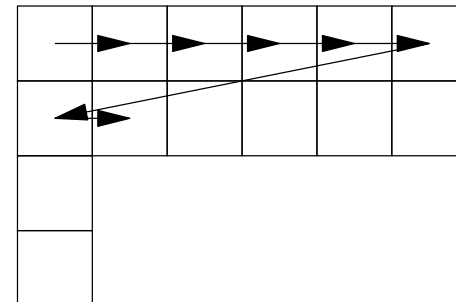
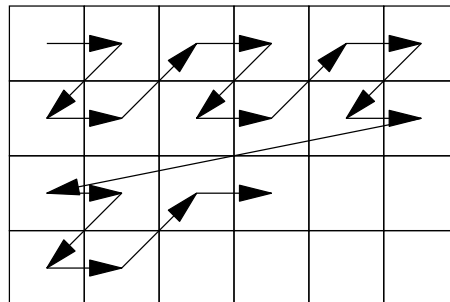
verlustbehaftete Kodierung intern mit 8 bpp / 12 bpp

"normale" Kodierung:

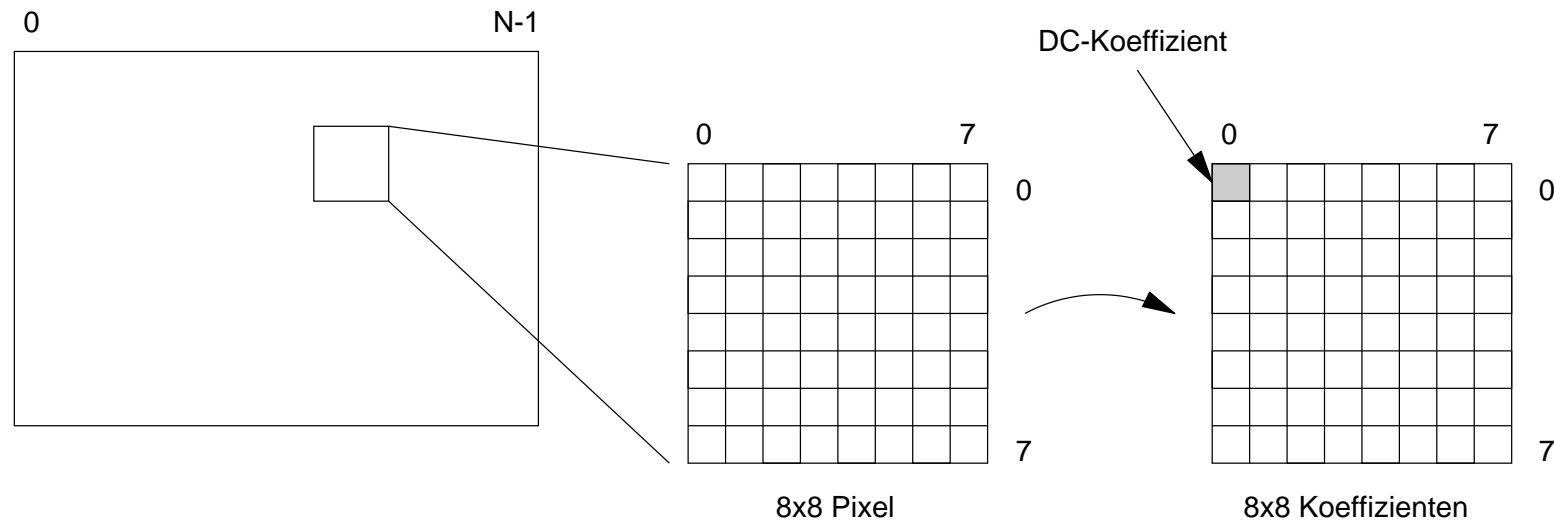


 = Block mit 8x8 Pixeln

"interleaved"
(progressive)



JPEG: Blöcke



- JPEG transformiert Blöcke, nicht einzelne Pixel
- Annahme: benachbarte Pixel haben ähnliche Werte

=> nach der 2D-DCT liegen die größten Koeffizienten "links oben"

- kein Ausnutzen von Korrelationen über Blockgrenzen hinaus
- Neigung zu "Blockartefakten" bei starker Kompression

JPEG: Quantisierung

Original Group

a. Eyebrow

231	224	224	217	217	203	189	196
210	217	203	189	203	224	217	224
196	217	210	224	203	203	196	189
210	203	196	203	182	203	182	189
203	224	203	217	196	175	154	140
182	189	168	161	154	126	119	112
175	154	126	105	140	105	119	84
154	98	105	98	105	63	112	84

b. Eye

42	28	35	28	42	49	35	42
49	49	35	28	35	35	35	42
42	21	21	28	42	35	42	28
21	35	35	42	42	28	28	14
56	70	77	84	91	28	28	21
70	126	133	147	161	91	35	14
126	203	189	182	175	175	35	21
49	189	245	210	182	84	21	35

c. Nose

154	154	175	182	189	168	217	175
154	147	168	154	168	168	196	175
175	154	203	175	189	182	196	182
175	168	168	168	140	175	168	203
133	168	154	196	175	189	203	154
168	161	161	168	154	154	189	189
147	161	175	182	189	175	217	175
175	175	203	175	189	175	175	182

DCT Spectrum

d. Eyebrow spectrum

174	19	0	3	1	0	-3	1
52	-13	-3	-4	-4	-4	5	-8
-18	-4	8	3	3	2	0	9
5	12	-4	0	0	-5	-1	0
1	2	-2	-1	4	4	2	0
-1	2	1	3	0	0	1	1
-2	5	-5	-5	3	2	-1	-1
3	5	-7	0	0	0	-4	0

e. Eye spectrum

70	24	-28	-4	-2	-10	-1	0
-53	-35	43	13	7	13	1	3
23	9	-10	-8	-7	-6	5	-3
6	2	-2	8	2	-1	0	-1
-10	-2	-1	-12	2	1	-1	4
3	0	0	11	-4	-1	5	6
-3	-5	-5	-4	3	2	-3	5
3	0	4	5	1	2	1	0

f. Nose spectrum

174	-11	-2	-3	-3	6	-3	4
-2	-3	1	2	0	3	1	2
3	0	-4	0	0	0	-1	9
-4	-6	-2	1	-1	4	-10	-3
1	2	-2	0	0	-2	0	-5
3	-1	3	-2	2	1	1	0
3	5	2	-2	3	0	4	3
4	-3	-13	3	-4	3	-5	3

Quantization Error

g. Using 10 bits

0	0	0	0	-1	0	0	0
-1	0	0	0	0	0	0	-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	1	0	0	0	-1	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

h. Using 8 bits

0	-3	-1	-1	1	0	0	-1
1	0	-1	-1	0	0	0	-1
-1	-2	1	0	-2	0	-2	-2
-1	-2	-1	2	0	2	0	1
0	-2	1	0	0	1	0	0
0	-4	-1	0	1	0	0	0
0	-2	0	1	-1	-1	1	-1
-1	-3	1	1	1	-3	-2	-1

i. Using 5 bits

-13	-7	1	4	0	0	10	-2
-22	6	-13	5	-5	2	-2	-13
-9	-15	0	-17	-8	8	12	25
-9	16	1	9	1	-5	-5	13
-20	-3	-13	-16	-19	-1	-4	-22
-11	6	-8	16	-9	-3	-7	6
-14	10	-9	4	-15	3	3	-4
-13	19	12	9	18	5	-5	10

- typisches Beispiel für (gerundete) Wert der Koeffizienten
- weitere Quantisierung: Koeffizienten durch Tabellenwerte teilen

JPEG: Quantisierung: Tabellen

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	193	121	120	101
72	92	95	98	112	100	103	99

Quantisierung für Luminanz (Y)

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

Quantisierung für Chrominanz (UV)

- einzelne Koeffizienten nach DCT bereits gerundet
- weitere Quantisierung nach Tabellen,
- einzelner Koeffizient wird durch Tabellenwert geteilt
- vordefinierte Tabellen nutzen physiologische Daten
Beispiel: $Y00' = Y00/16$, $V13' = V13/66$, usw.

JPEG: Quantisierung: Beispiel

32	4	-3	0	0	0	1	0
6	-2	0	0	-2	0	0	1
-7	3	2	0	0	2	0	0
0	0	0	0	0	0	0	0
6	0	1	0	0	0	0	0
0	-1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

(a)

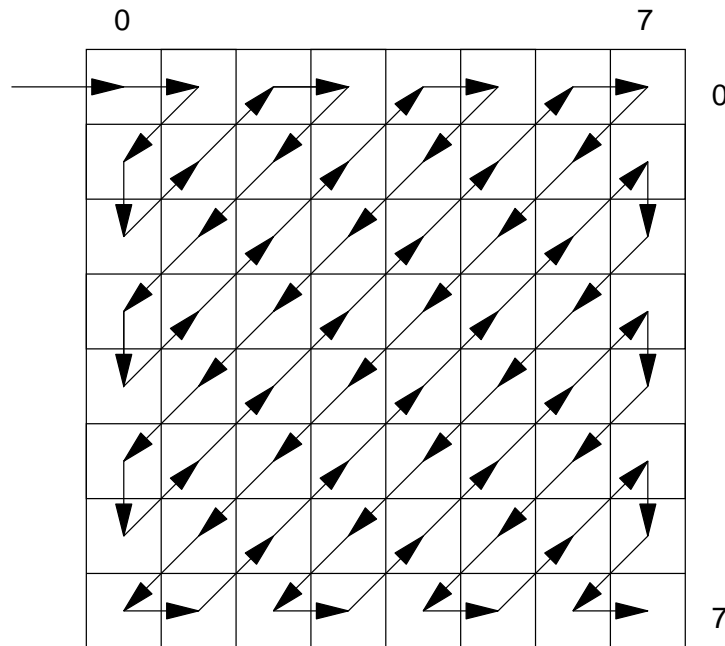
70	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

(b)

Table 4.53: (a) The Quantized Coefficients of Table 4.49.
(b) Those of Table 4.50.

- Beispiel für (typ.) Werte der quantisierten Koeffizienten
- DC-Koeffizient hat großen Wert
- Koeffizienten "links oben" sind klein oder null
- Koeffizienten "rechts unten" sind (fast) alle null
- weitere Kodierung mit Huffman/Runlength-Verfahren

JPEG: ZigZag



DC AC01 AC10 AC20 AC11 AC02 ...
AC03 AC12 AC21 ... AC76 AC77

- Koeffizienten kleiner Frequenz liegen "oben links"
- "ZigZag"-Anordnung der Koeffizienten für die Huffman-Phase
- erzeugt Paare von "00..00"-Folgen und kleinen Integern
- lange "000..0" Folge am Ende

JPEG: Huffman

- Beispiel-Sequenz von Koeffizienten (nach der Quantisierung):

1118, 2, 0, -2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1, 0, ...

- (Z, V)-Paare: Anzahl der Nullen vor dem Wert V

DC=1118, (0, 2), (1, -2), (13, -1), (...)

- Kodetabellen für (Z,V)-Paare
- Zahldarstellung mit r bits, r abhängig von Tabellenzeile

(0, 2) -> 01 | 10

(1, -2) -> 1100 | 1110

(13, -1) -> 11111111010 | 1111111111

- am Ende (EOB) Marker, z.B. 64 Koeffizienten -> 50 bits

JPEG: Huffman-Tabellen

Z	R				
	1 6	2 7	3 8	4 9	5 A
0	00 1111000	01 11111000	100 1111110110	1011 111111110000010	11010 1111111110000011
1	1100 111111110000100	11011 111111110000101	11110001 111111110000110	111110110 111111110000111	11111110110 111111110001000
2	11100 111111110001010	11111001 111111110001011	1111110111 111111110001100	111111110100 111111110001101	111111110001001 111111110001110
3	111010 1111111110010001	111110111 1111111110010010	111111110101 1111111110010011	111111110001111 1111111110010100	1111111110010000 1111111110010101
4	111011 1111111110011001	1111111000 1111111110011010	111111110010110 1111111110011011	111111110010111 1111111110011100	1111111110011000 1111111110011101
5	1111010 1111111110100001	11111110111 1111111110100010	111111110011110 1111111110100011	111111110011111 1111111110100100	1111111110100000 1111111110100101
6	1111011 1111111110101001	111111110110 1111111110101010	1111111110100110 1111111110101011	1111111110100111 1111111110101100	1111111110101000 1111111110101101
7	11111010 1111111110110001	111111110111 1111111110110010	1111111110101110 1111111110110011	1111111110101111 1111111110110100	1111111110110000 1111111110110101
8	111111000 1111111110111001	111111111000000 1111111110111010	1111111110110110 1111111110111011	1111111110110111 1111111110111100	1111111110111000 1111111110111101
9	111111001 1111111111000010	1111111110111110 1111111111000011	1111111110111111 1111111111000100	1111111111000000 1111111111000101	1111111111000001 1111111111000110
A	111111010 1111111111001011	1111111111000111 1111111111001100	1111111111001000 1111111111001101	1111111111001001 1111111111001110	1111111111001010 1111111111001111
B	1111111001 1111111111010100	1111111111010000 1111111111010101	1111111111010001 1111111111010110	1111111111010010 1111111111010111	1111111111010011 1111111111011000
C	1111111010 1111111111011101	1111111111011001 1111111111011110	1111111111011010 1111111111011111	1111111111011011 1111111111100000	1111111111011100 1111111111100001
D	11111111000 1111111111100110	1111111111100010 1111111111100111	1111111111100011 1111111111101000	1111111111100100 1111111111101001	1111111111100101 1111111111101010
E	111111111101011 1111111111110000	1111111111101100 1111111111110001	1111111111101101 1111111111110010	1111111111101110 1111111111110011	1111111111101111 1111111111110100
F	11111111001 1111111111111001	1111111111110101 1111111111111010	1111111111110110 1111111111111011	1111111111110111 1111111111111101	1111111111111000 1111111111111110

Table 4.56: Recommended Huffman Codes For Luminance AC Coefficients.

(erster Teil der Tabelle)

JPEG2000

- JPEG hat sich bewährt

aber diverse Erweiterungen wünschenswert:

- bessere Qualität bei geringen Bitraten (< 0.25 bpp)
- einheitliches Format für S/W-, Grau-, und Farbbilder
- Kombination von verlustfreier und verlustbehafteter Kodierung:
- progressive Kodierung: Vorschau ... verlustfreie Kodierung
- Qualitätsstufen, z.B. 72 dpi Monitor vs. 2400dpi Drucker
- "region of interest" Kodierung
- Robustheit, bessere Korrektur von Bitfehlern
- offene, erweiterbare Architektur
- Unterstützung für Indizierung / Inhaltssuche / MPEG-7
- Unterstützung für Verschlüsselung, Wasserzeichen, ...

(Christopoulos, IEEE Tr. CE 46-4, 1103, 2000)

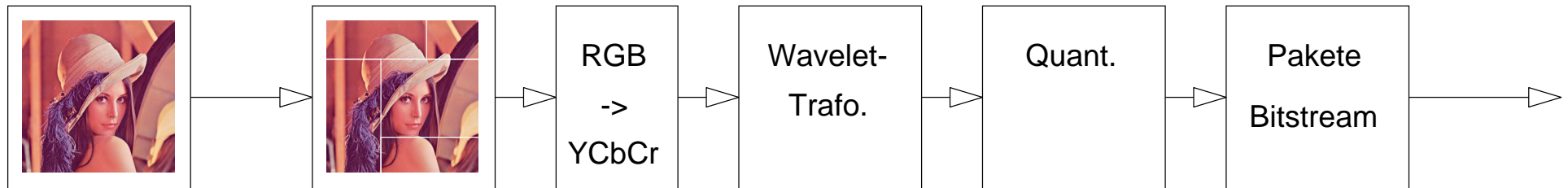
JPEG 2000

ähnliches Prinzip wie JPEG:

	JPEG	JPEG 2000
• Bildvorbereitung, Farbsubsampling		
• lineare Transformation	DCT	Wavelets
• Entropiekodierung	Huffman	arithmetisch
• keine feste Blockgröße für die Wavelet-Transformation		
• gesamtes Bild, oder "Tiles" beliebiger Größe		
• verlustfreie / -behaftete Kodierung (Integer/Float, zwei Wavelets)		
• Wavelets kodieren Bild in verschiedener Auflösung		
• daher weniger Neigung zu Blockartefakten		
• Bildqualität nicht unbedingt besser als JPEG oder PNG		
• nur bei sehr hoher Kompression		

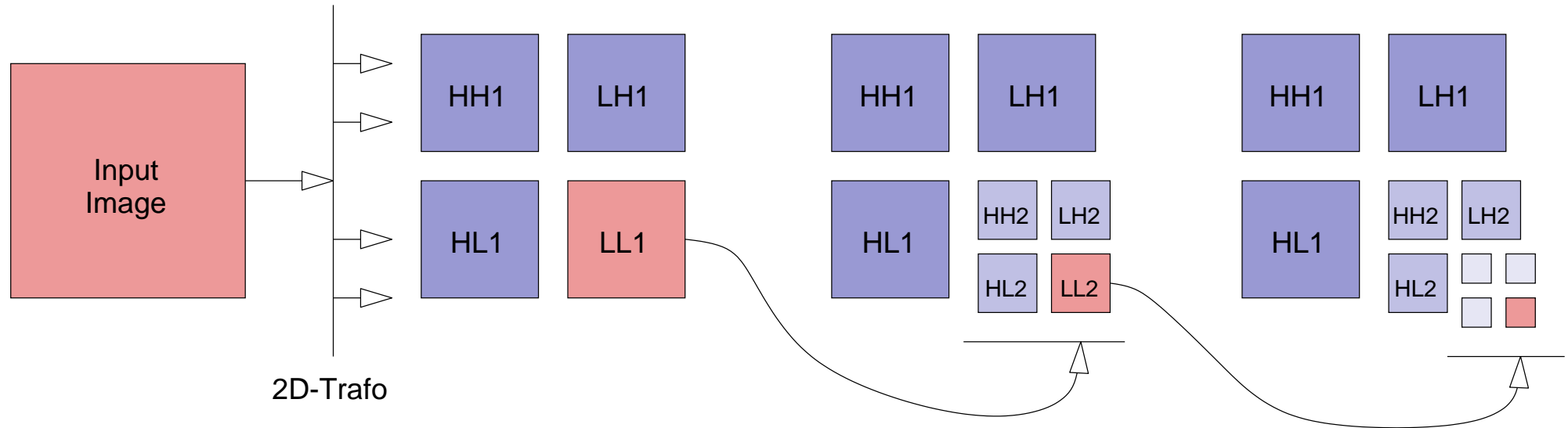
(Santa-Cruz et al. SPIE ADIP 23, 4446, 2000)

JPEG 2000: Prinzip



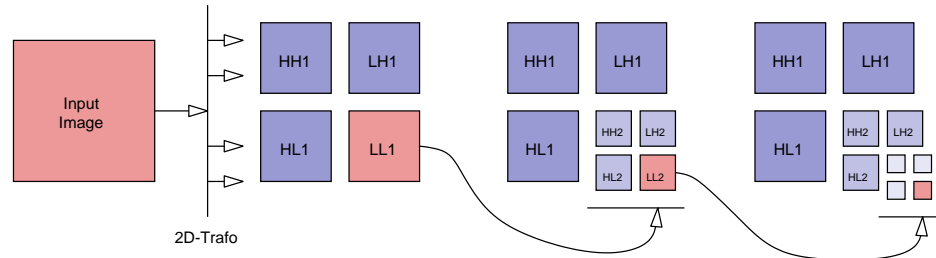
- Aufteilung in Bereiche ("tiles") (Speicherbedarf kleiner)
- reversible Farbtransformation (RGB -> YCbCr)
- 2D-Wavelet-Transformation (zwei Varianten)
- Quantisierung ("skalierbar")
- Bitstream (auch teilweise dekodierbar)

JPEG 2000: 2D-Wavelet-Transformation



- Hochpass- und Tiefpassfilterung des Eingangsbildes
- Eingangsbild wird in vier Teilbilder zerlegt (HH, LH, HL, LL)
- Operation wird auf dem Tiefpass-Teilbild wiederholt
- der erste Schritt liefert grobe Strukturen des Bildes
- spätere Schritte immer feinere Strukturen
- so viele Schritte kodieren / dekodieren wie nötig

JPEG 2000: (5,3) Wavelet



$$H[n] = x[2n+1] + \left[\frac{1}{2} - \frac{1}{2} x[2n] - \frac{1}{2} x[2n+2] \right]$$

Kodierung
(Analyse)

$$L[n] = x[2n] + \left[\frac{1}{2} + \frac{1}{4} H[n-1] + \frac{1}{4} H[n] \right]$$

$$x[2n] = L[n] - \left[\frac{1}{2} + \frac{1}{4} H[n-1] + \frac{1}{4} H[n] \right]$$

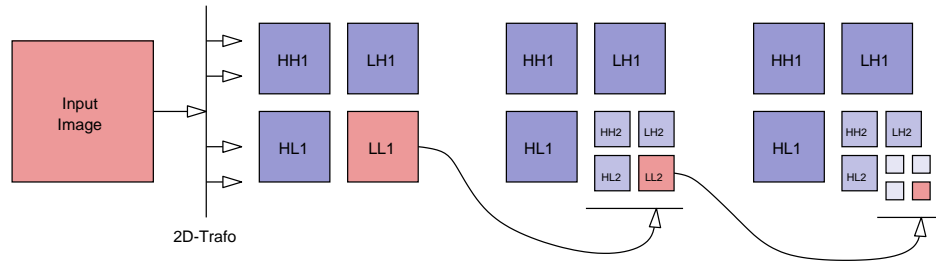
Dekodierung
(Synthese)

$$x[2n+1] = H[n] - \left[\frac{1}{2} - \frac{1}{2} x[2n] - \frac{1}{2} x[2n+2] \right]$$

- verlustfrei durch sorgfältiges Auf/Abrunden

(A. Bilgin, M.W. Marcellin in 'lossless compression')

JPEG 2000: (9,7) Wavelet

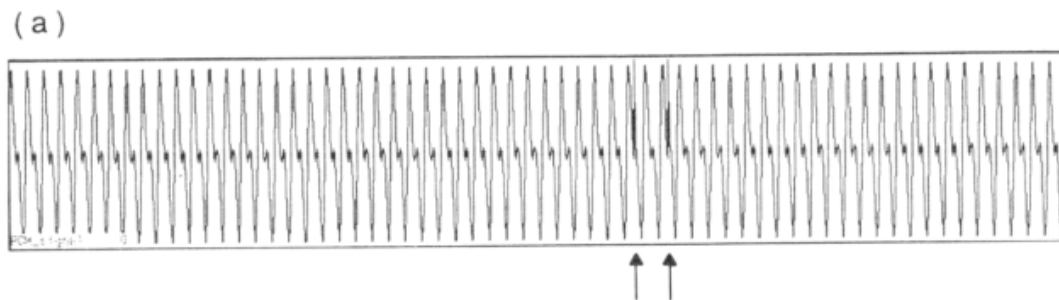


$$L_0(z) = 0.602949018236 + 0.266864118443(z^1 + z^{-1}) \\ -0.078223266529 (z^2 + z^{-2}) \\ -0.016864118443 (z^3 + z^{-3}) \\ +0.026748757411 (z^4 + z^{-4})$$

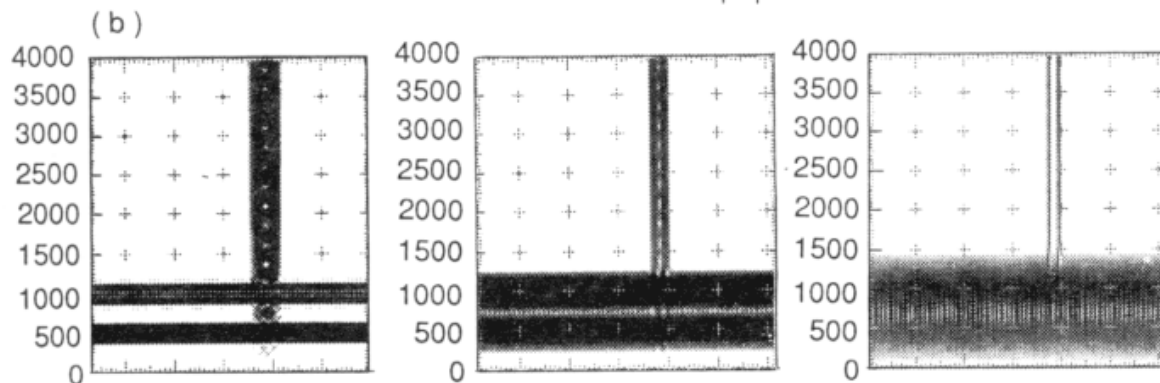
$$H_0(z) = 0.557543526229 - 0.295635881557 (z^1 + z^{-1}) \\ -0.028771763114 (z^2 + z^{-2}) \\ + 0.045635881557 (z^3 + z^{-3})$$

- verlustbehaftet, Gleitkomma, gut für hohe Kompressionsraten

Wavelet- vs. Fourier-Analyse



Eingangssignal:
zwei Sinussignale 500 Hz/1 KHz,
plus zwei Störimpulse (Pfeile)



Fourieranalyse:
Fenstergröße 12.8 / 6.4 / 3.2 msec.
abnehmende Frequenzauflösung
zunehmende Zeitauflösung

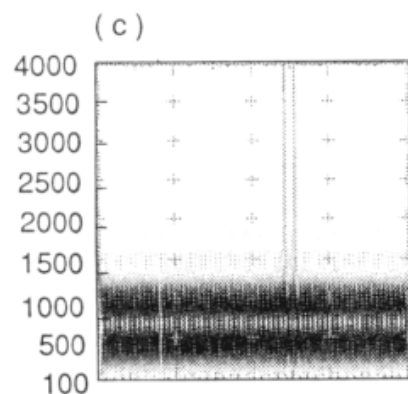


FIG. 1.3. (a) The signal $f(t)$. (b) Windowed Fourier transforms of f with three different window widths. These are so-called spectrograms: only $|T^{\text{win}}(f)|$ is plotted (the phase is not rendered on the graph), using grey levels (high values = black, zero = white, intermediate grey levels are assigned proportional to $\log|T^{\text{win}}(f)|$) in the t (abscissa), ω (ordinate) plane. (c) Wavelet transform of f . To make the comparison with (b) we have also plotted $|T^{\text{wav}}(f)|$, with the same grey level method, and a linear frequency axis (i.e., the ordinate corresponds to a^{-1}). (d) Comparison of the frequency resolution between the three spectrograms and the wavelet transform. I would like to thank Oded Ghitza for generating this figure.

Wavelet-Analyse:
gute Zeitauflösung, mittlere Frequenzauflösung

(Daubechies: Ten lectures on wavelets, 1.2)

JPEG 2000: Beispiel



JPEG 2000, 9819 bytes, 0.3bpp



JPEG, 9643 bytes



JPEG, 11904 bytes

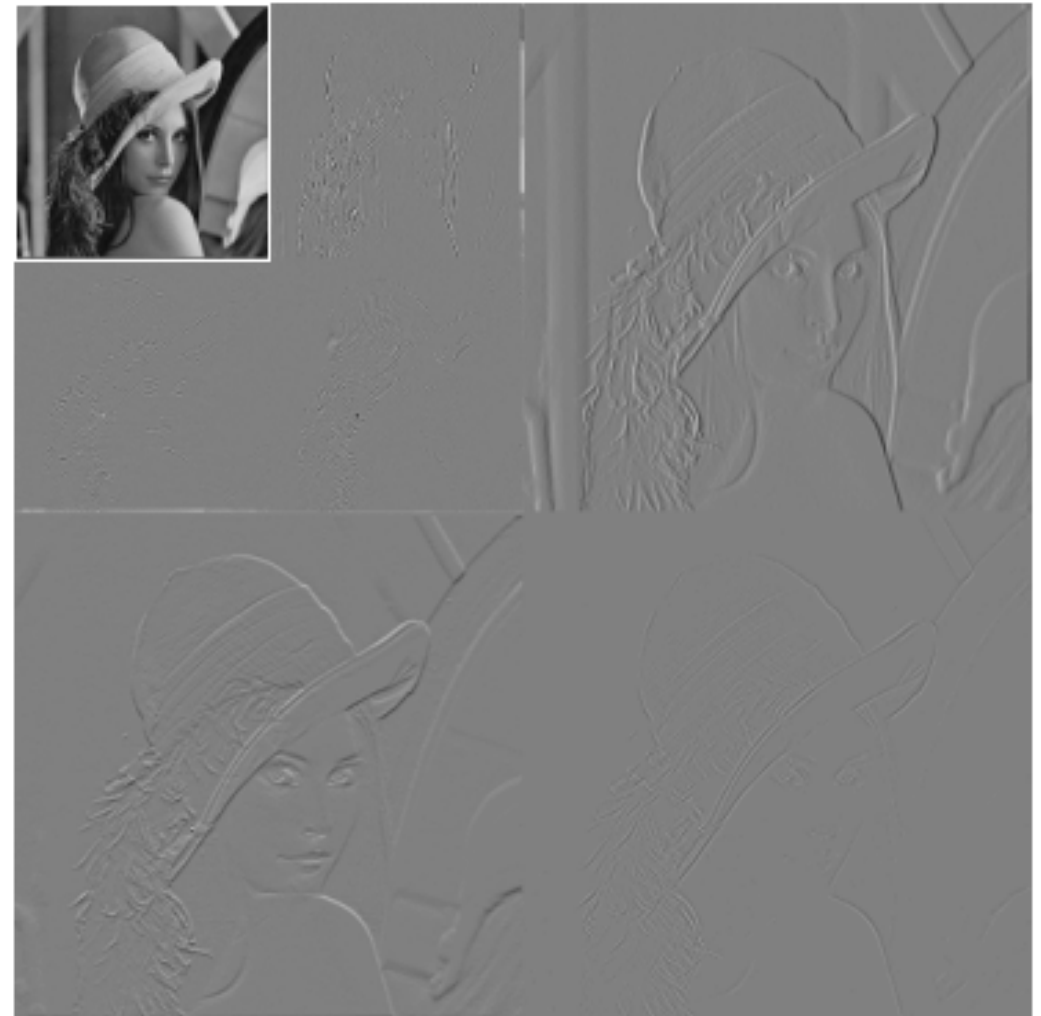
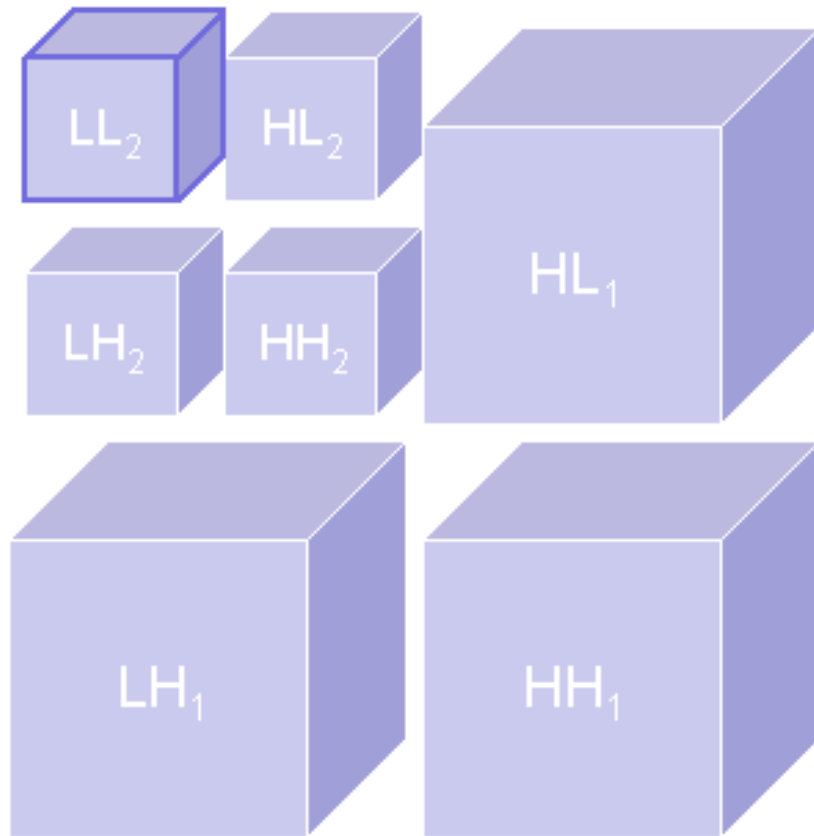
- Vorführung des JJ2000 Encoders/Decoders

```
java JJ2KEncoder -i lena.ppm -o lena.j2k -rate 0.3
```

```
java JJ2KDecoder -i lena.j2k
```

(jj2000.epfl.ch)

JPEG 2000: Skalierung



- rekursive Anwendung der Wavelet-Transformation
- Tiefpaßanteile "oben links", Details "unten und rechts"

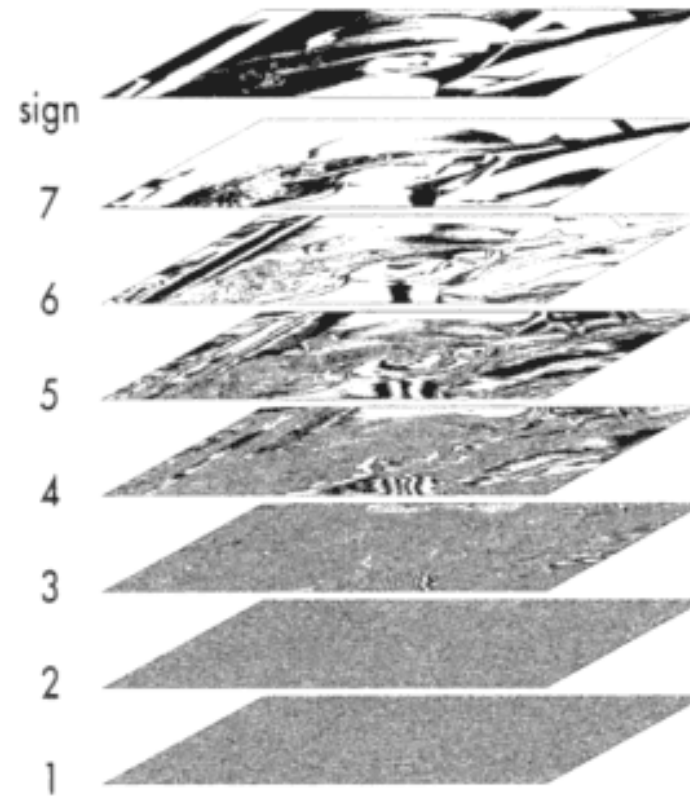
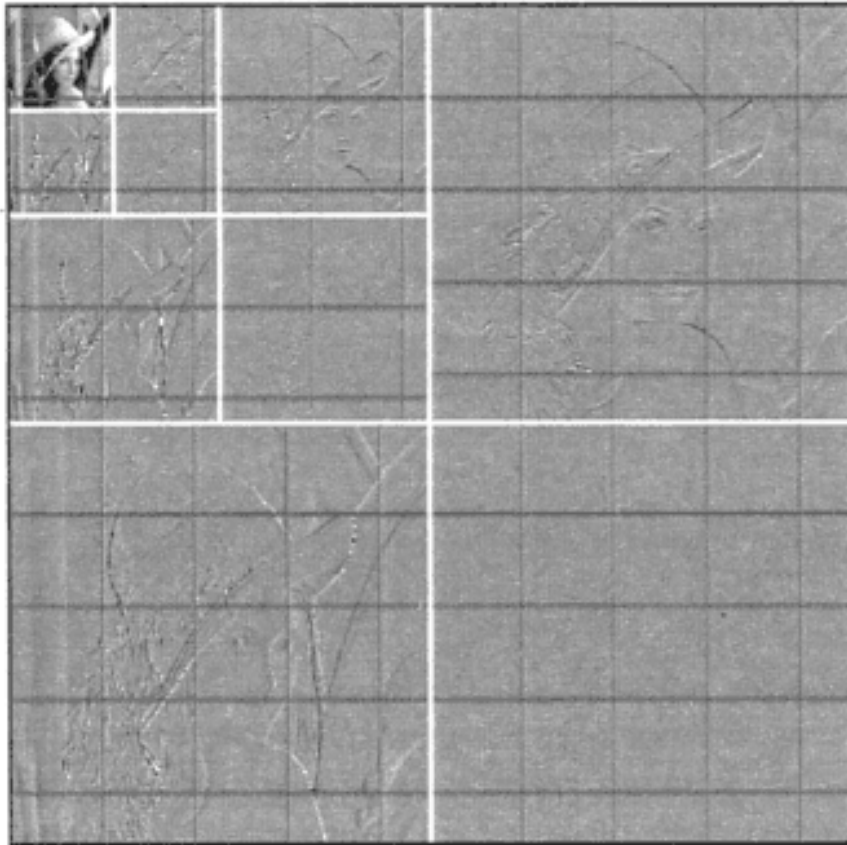
JPEG 2000: *progressive Kodierung*



Flexible Progression: Verwendet man nur einen Teil der komprimierten Daten (Prozentangabe), zeigt sich, in welcher Reihenfolge die Pakete in der Datei gespeichert sind. Hier zu sehen (von oben nach unten): qualitative Verbesserung, Steigerung der Auflösung und Komponentenprogression.

(c't 22/2001, S.187)

JPEG 2000: Bitplanes



Das Bild (links) liegt nach einer dreistufigen Wavelet-Transformation in vier Auflösungsstufen vor (gelb). Die Auflösungsstufen werden in gleich große Blöcke zerlegt (blau). Auf der rechten Seite sieht man die Bitebenen des linken oberen Blocks.

- weitere Skalierungsmöglichkeit: Übertragung der MSBs zuerst