

# JPEG

Verlustbehafteter Komprimierungsalgorithmus für digitale Bilder

## 1. Transformation RGB → Luminanz/Chrominanz (Keine Kompression)

Auge ist empfindlicher auf Helligkeitsunterschiede als auf Farbunterschiede, daher werden Farbinformationen komprimiert.

RGB wird in YCbCr umgewandelt (verlustfreie Vorbereitung)

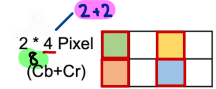
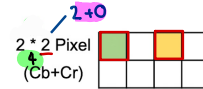
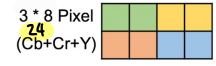
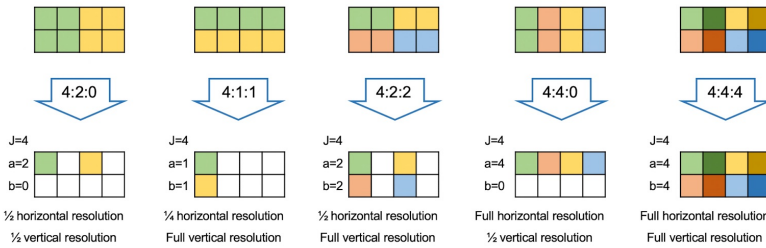
Y: Luminanz (Graustufenintensität), Cb: Chrominanz (Blauanteil), Cr: Chrominanz (Rotanteil)

## 2. Downsampling der beiden Chrominanz-Komponenten (Verlustbehaftete Kompression)

Mehrere Pixel in der Chrominanz-Ebene werden zusammengefasst. (verlustbehaftete Komprimierung)

Schema-Notation: (J:a:b), J: Breite in Pixel, a: Anzahl Pixel auf 1. Zeile, b: Anzahl Pixel auf 2. Zeile

Kompressionsrate:  $\frac{\text{Resultierende Pixel}}{\text{Ursprüngliche Pixel}}$  (über alle Ebenen)



$$R = \frac{12}{24} = \frac{1}{2}$$

$$R = \frac{16}{24} = \frac{2}{3}$$

## 3. Pixel-Gruppierung in 8x8 Blöcke (Keine Kompression)

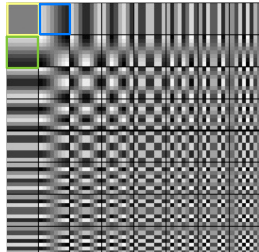
## 4. Diskrete Kosinus Transformation (DCT) (Verlustlose Kompression)

Das gesamte Bild wird in 8x8 Blöcke (64 Pixel) aufgeteilt, wobei jeder Pixel einen Wert von 0 (Schwarz) - 255 (Weiss) hat, welcher die Helligkeit repräsentiert.

8x8 Block

214	199	184	169	155	140	125	110
214	199	184	169	155	140	125	110
214	199	184	169	155	140	125	110
214	199	184	169	155	140	125	110
214	199	184	169	155	140	125	110
214	199	184	169	155	140	125	110
214	199	184	169	155	140	125	110
214	199	184	169	155	140	125	110

64x64 DCT Muster



8x8 DCT Block

1297	268	15	25	1	7	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

Hohe Frequenzen werden ausgefiltert, da Auge diese nicht wahrnehmen kann. Daher hat nur DC-Wert und ein paar AC-Werte grossen Wert.

Grosse regelmässige Flächen → niedriger Frequenzanteil

Feine Details und genaue Auflösung → hoher Frequenzanteil

{ {1024, 512, 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}}

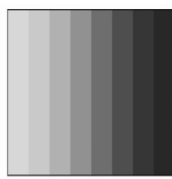


Bild #1	Bild #2	Bild #3	Bild #4	Bild #5	Bild #6
dunkler	heller	mehr Kontrast	weniger Kontrast	Gradient dunkel-hell	Gradient vertikal

Frequenzmatrix (oberste zwei Zeilen)	Bild #
{ {1024, 768, 0, 0, 0, 0, 0, 0}, { 0, 0, 0, 0, 0, 0, 0, 0}, ...	#3
{ {1536, 512, 0, 0, 0, 0, 0, 0}, { 0, 0, 0, 0, 0, 0, 0, 0}, ...	#2
{ {1024, 0, 0, 0, 0, 0, 0, 0}, { 512, 0, 0, 0, 0, 0, 0, 0}, ...	#6
{ {1024, -512, 0, 0, 0, 0, 0, 0}, { 0, 0, 0, 0, 0, 0, 0, 0}, ...	#5

- DC-Wert, bestimmt die Helligkeit des gesamten Blocks 0 = Schwarz, 2040 = Weiss  
Alles im Minus-Bereich ist Schwarz. Je höher desto heller, je tiefer desto dunkler.
- AC Wert, je höher desto mehr Kontrast, je tiefer desto weniger Kontrast.  
Im Minus-Bereich ist DCT Muster invertiert. (Bei 0 ist Muster nicht sichtbar)
- DCT Muster an Position wie Position des betrachteten Werts vergleichen.

## 5. Quantisierung (Verlustbehaftete Kompression)

Originale Koeffizienten werden durch Wert in Quantisierungstabelle geteilt und auf ganze Zahlen gerundet.

→ unwichtige DCT-Koeffiziente gehen verloren.

z.B. 1260 (Q00): 16 = 78.75 ≈ 79

-3 (Q70): 72 = -0.04 ≈ 0

Originale Koeffizienten (F <sub>vu</sub> )							
Q	0	1	2	3	4	5	6
0	1260	-1	-12	-5	2	-2	-3
1	-23	-18	-6	-3	-3	0	0
2	-11	-9	-2	2	0	-1	-1
3	-7	-2	0	2	1	0	0
4	-1	-1	2	2	0	-1	1
5	2	0	2	0	-1	2	-1
6	-1	0	0	-2	-1	2	-1
7	-3	2	-4	-2	2	1	-1

Quantisierungstabelle (Q<sub>vu</sub>)

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

Quantisierte Koeffizienten  
F<sub>vu</sub> = round(F<sub>vu</sub>/Q<sub>vu</sub>)

79	0	-1	0	0	0	0	0
-2	-1	0	0	0	0	0	0
-1	-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	0	0	0	0	0
0	0	0	0	0	0	0	0

## 6. Entropy-Coding (Verlustlose Kompression)

Verlustlose Komprimierung mit Run-Length-Coding (RLE) und Huffmankodierung

79	0	-1	0	0	0	0	0
-2	-1	0	0	0	0	0	0
-1	-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	0	0	0	0	0
0	0	0	0	0	0	0	0

RLE: (DC Wert) (Anzahl Nullen, Koeffizient) ... (EOB)

z.B. (79), (1, -2), (0, -1), (0, -1), (0, -1), (2, -1), (0, -1), (EOB)

→ EOB, End of Block, alles 0 bis zum Ende

### 24 Bit / Pixel

Originalbild mit True Color Auflösung.

### 1.5 - 2.0 Bit / Pixel (Kompressionsfaktor 12 ... 16)

Normalerweise nicht vom Original unterscheidbar.

Genügt den höchsten professionellen Anforderungen.

### 0.75 - 1.5 Bit / Pixel (Kompressionsfaktor 16 ... 32)

Exzellente Qualität.

Genügt den meisten Anforderungen.

### 0.5 - 0.75 Bit / Pixel (Kompressionsfaktor 32 ... 48)

Gute bis sehr gute Qualität.

Genügend für viele Anwendungen.

### 0.25 - 0.5 Bit / Pixel (Kompressionsfaktor 48 ... 96)

Bescheidene bis gute Qualität.

Genügend für gewisse Anwendungen.