Computergrafik SS 2016

Oliver Vornberger

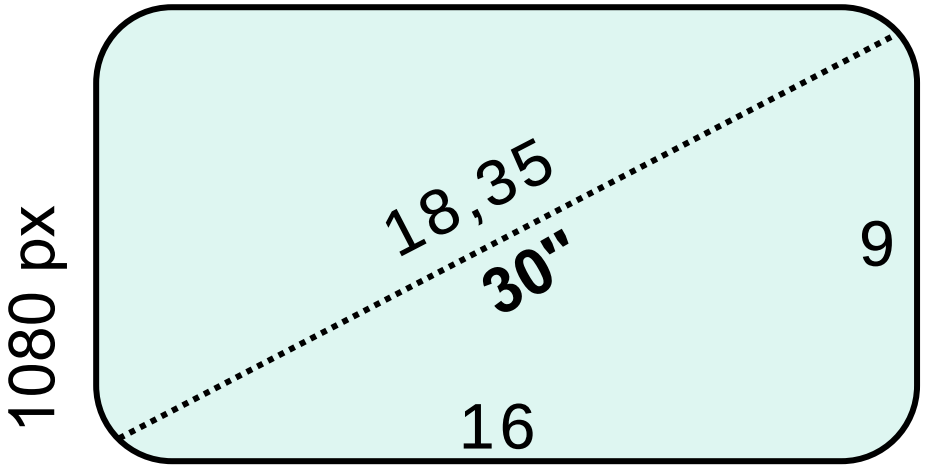
Kapitel 9:
Pixeldateien

Auflösung

gemessen in dots per inch (dpi)

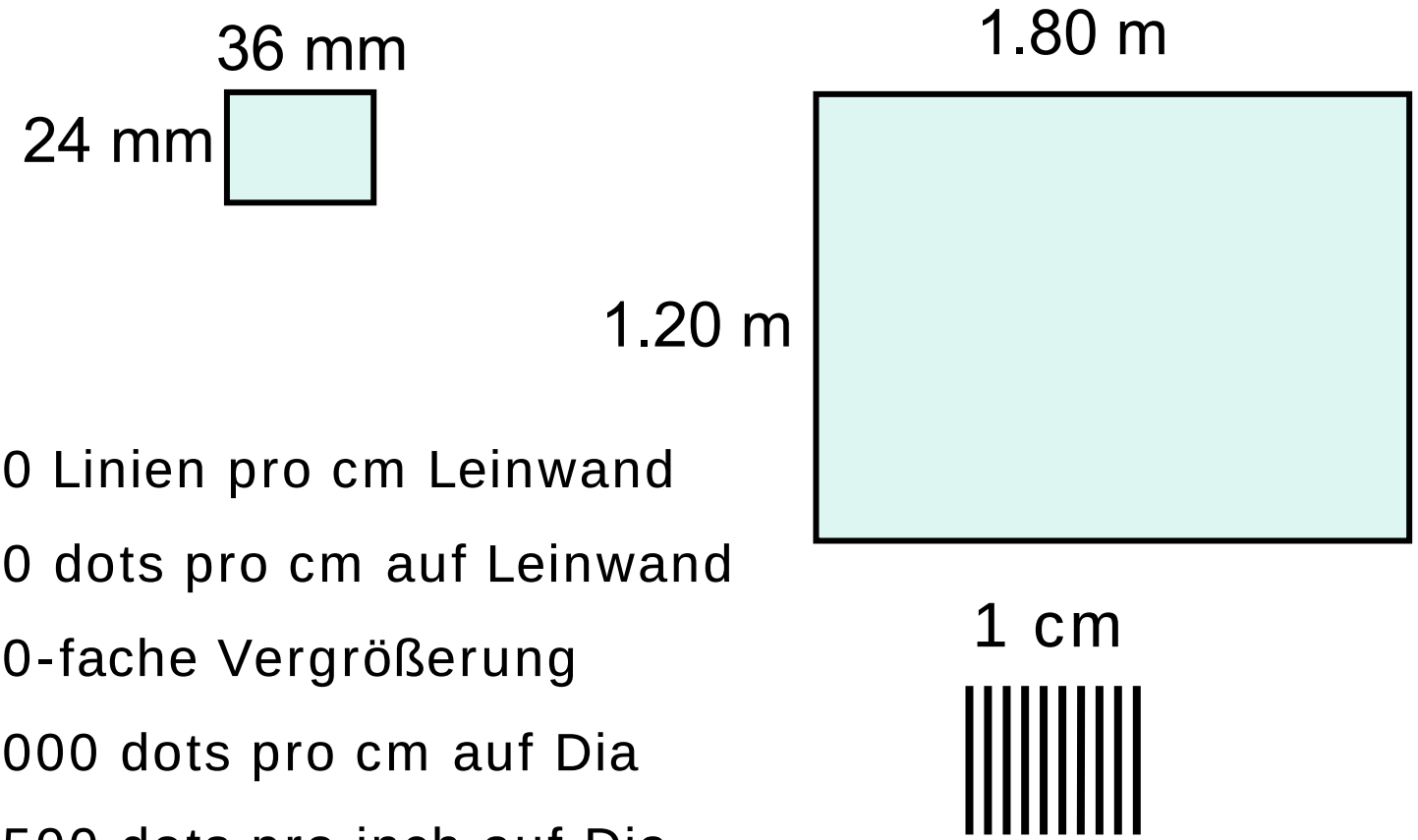
- Scanner-Auflösung
- Scan-Auflösung
- Bild-Auflösung
- Monitor-Auflösung
- Drucker-Auflösung
- Druck-Auflösung

30'' 16:9 Full-HD Monitor-Auflösung



1920 px / 26.1'' = 73.5 dpi

Dia-Auflösung



10 Linien pro cm Leinwand

20 dots pro cm auf Leinwand

50-fache Vergrößerung

1000 dots pro cm auf Dia

2500 dots pro inch auf Dia

Abzug vom Dia

- Dia, eingescannt mit 2500 dpi
- 3.60 / 2.54 * 2500 = 3543 Pixel
- 2.40 / 2.54 * 2500 = 2362 Pixel
- gedruckt mit 300 dpi ergibt

  = 3543 / 300 inch x 2362 / 300 inch

  = 30 cm x 20 cm

  = DIN-A4

Quiz

Um welcher Faktor ist die Druckfläche eines Bildes auf einem 600 dpi-Drucker kleiner als die Darstellung auf einem 75 dpi-Monitor ?

1 . . .
0,0 %

2 . . .
0,0 %

3 . . .
0,0 %

4 . . .
0,0 %

64

Quiz

Wieviel Kilobytes enthält der Scan eines 3 cm x 4 cm großen Passbildes mit einem 600 dpi-Scanner (1 Inch = 2.5 cm) ?

1 . . .
0,0 %

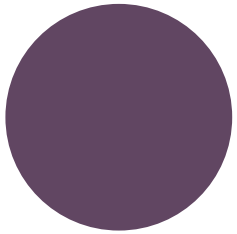
2 . . .
0,0 %

3 . . .
0,0 %

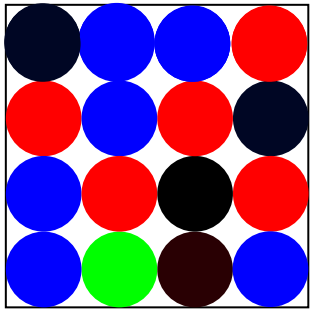
4 . . .
0,0 %

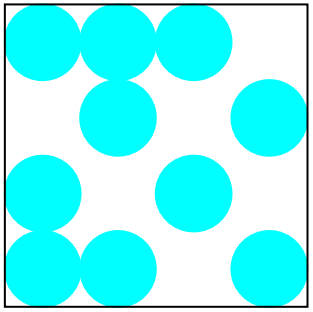
2025

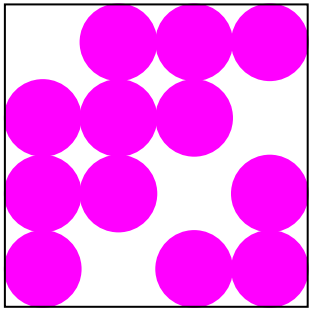
Rasterzelle

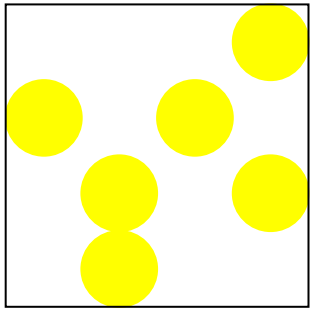


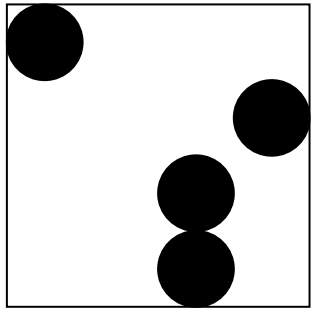
RGB	97	70	98	
CMY	158	185	157	
CMYK	158	192	97	64
4x4 Raster	10	12	6	4







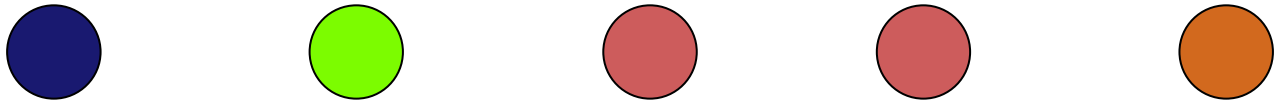




Demo in Adobe Photoshop

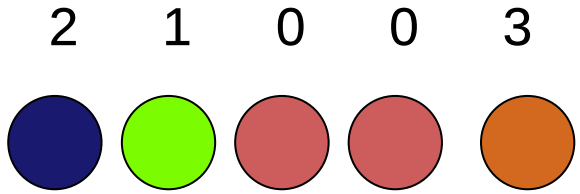
True Color

[25,25,112] [124,252,0] [205,92,92] [205,92,92] [25,25,112]



pro Pixel ein RGB-Wert = 3 Byte

Farbtabelle



bei 256 Einträgen

pro Pixel 1 Byte

bei 2^p Einträgen

pro Pixel p Bits

0	**205**	**92**	**92**
1	**124**	**252**	**0**
2	**25**	**25**	**112**
3	**210**	**105**	**30**
.			
.			
.			
255			

Standard-Farbpalette

- Bilde pro Farbwert 6 Abstufungen

x	0..25	26..76	77..127	128..178	179..229	230..255
q(x)	0	51	102	153	204	255

- trage alle Kombinationen in Tabelle ein
- Tabelle enthält 6·6·6 = 216 Einträge
- Quantisiere durch q(R),q(G),q(B)

$$q(x) := \left\lfloor \frac{x + 25}{51} \right\rfloor \cdot 51$$

- Vergib zuständigen Index

Bildbezogene Farbtabelle

Definiere Abstand zwischen zwei Farben

(a_r, a_g, a_b) (b_r, b_g, b_b)

$$d(a,b) = \sqrt{(a_r - b_r)^2 + (a_g - b_g)^2 + (a_b - b_b)^2}$$

Sei F die Menge der beobachteten Farben.

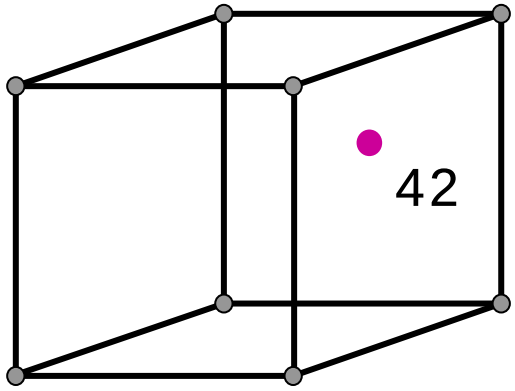
Suche Menge M von Repräsentanten.

Minimiere $$\Delta := \max_{x \in F} \min_{p \in M} d(p,x)$$

Popularity-Algorithmus

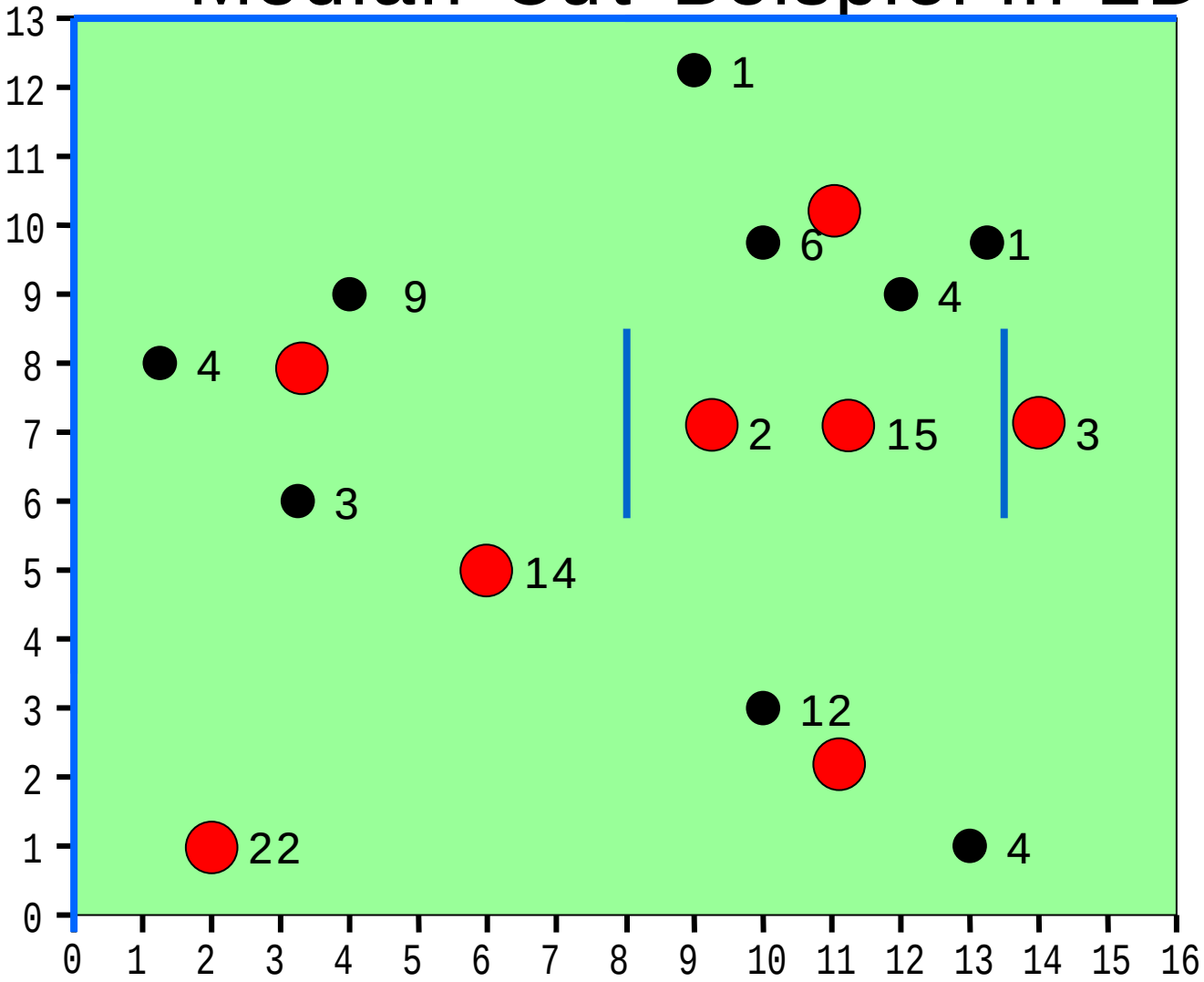
- Wähle die k häufigsten Farben
- Problem: selten vorkommende Farben werden sehr schlecht repräsentiert.

Median Cut



- Gesucht: p Repräsentanten
- Bilde RGB-Würfel
- an Position (x,y,z) vermerke die Häufigkeit der beobachteten Farbe (x,y,z)
- Bilde p Teilwürfel
- Bestimme Repräsentant pro Teilwürfel

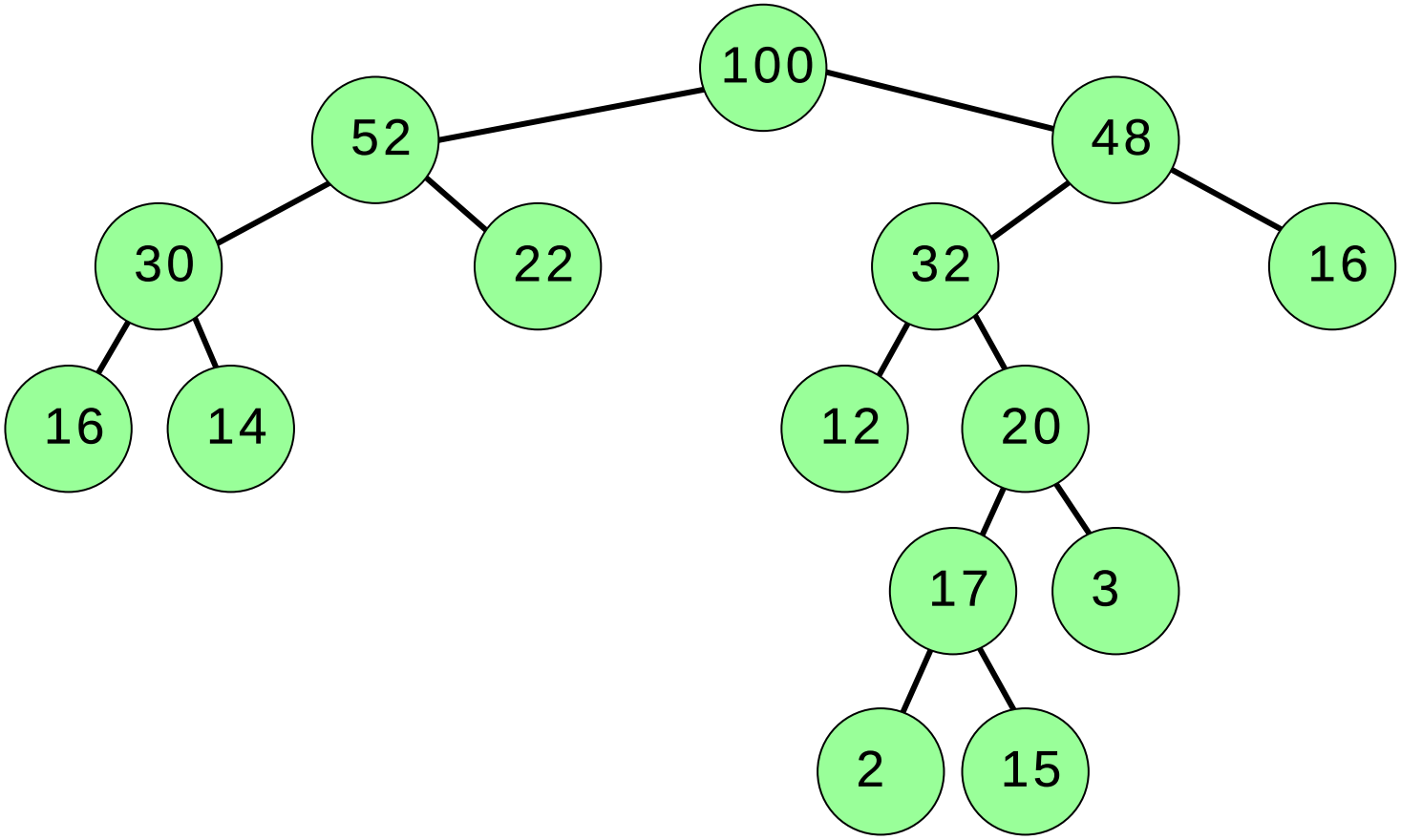
Median-Cut-Beispiel in 2D



10 x 10 Pixel

Farben aus [0..15], [0..15]

Median-Cut-Baum



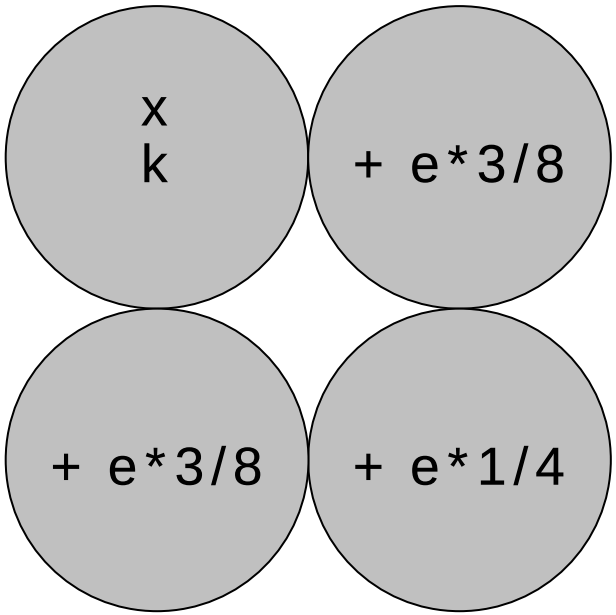
Floyd-Steinberg-Dithering

Farbe x

Repräsentant k = p(x)

Fehler e = d(x,k)

färbe mit k
und verteile Fehler

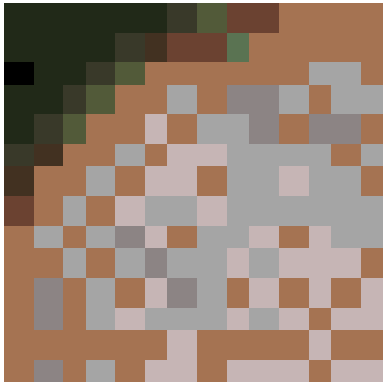


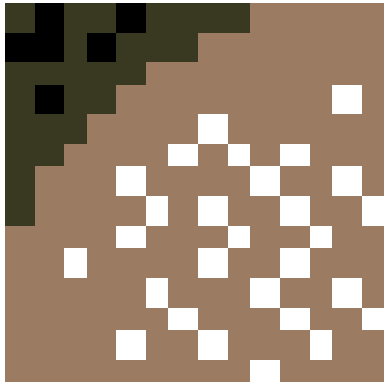
Floyd-Steinberg-Dithering-Beispiel

24 Bit

8 Bit

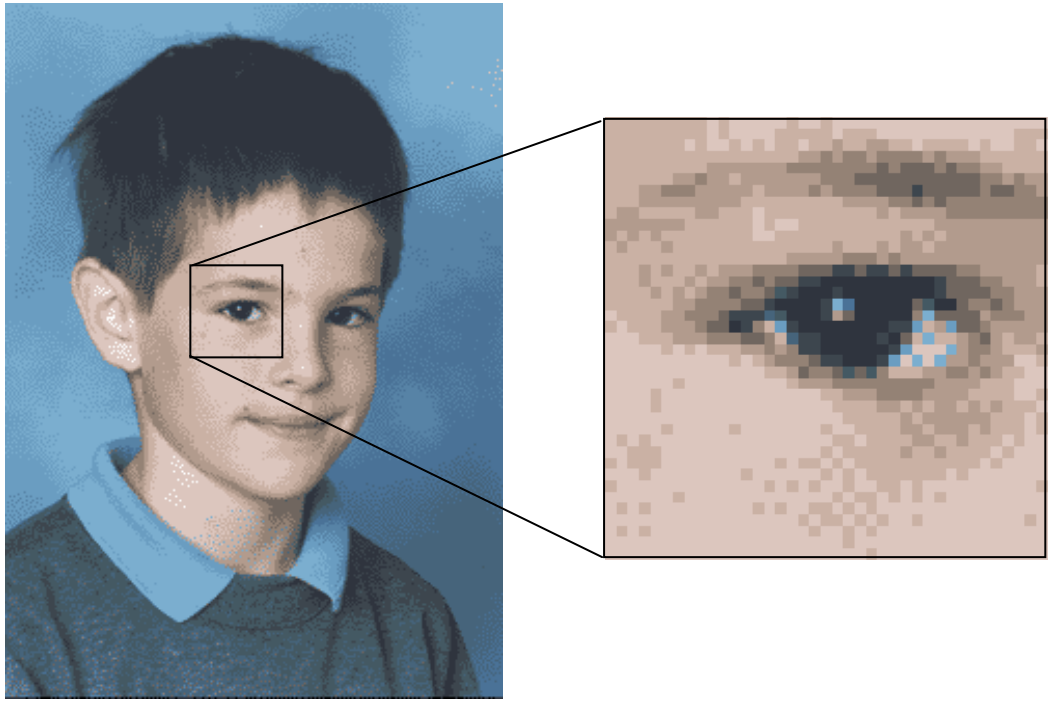
4 Bit





2 Bit

Adobe Photoshop



True-Color	24 Bit	Indiziert	4 Bit
jan.tif	239 KB	jan.gif	15 KB

Kompressionsfaktor 16

LZW

komprimiere Text
durch Verweise
auf Strings in Tabelle

ababcbababa

1 2 4 3 5 8

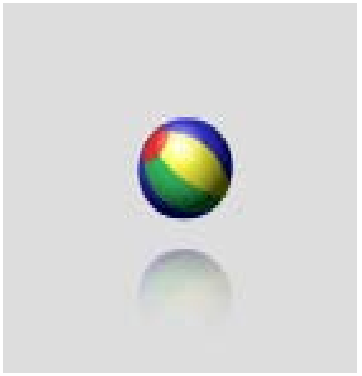
	String		Code
a	**.**	**a**	**1**
b	**.**	**b**	**2**
c	**.**	**c**	**3**
ab	**1**	**b**	**4**
ba	**2**	**a**	**5**
abc	**4**	**c**	**6**
cb	**3**	**b**	**7**
bab	**5**	**b**	**8**
baba	**8**	**a**	**9**

GIF

- Farbtabelle mit bis zu 256 Farben
- Transparenz
- LZW-Komprimierung
- Animation

PNG

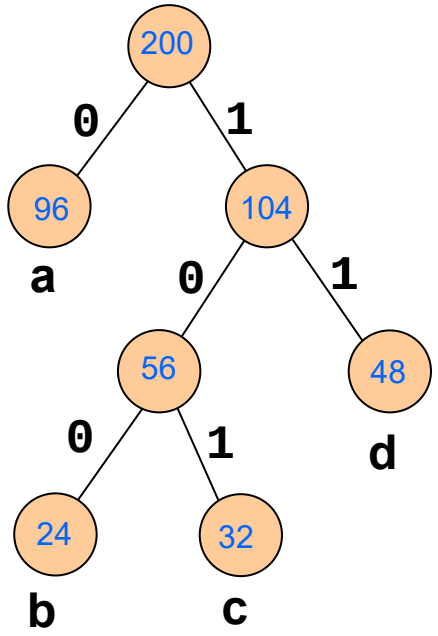
- True-Color-Bild
- Alphakanal mit bis zu 2^{16} Abstufungen
- Animation



Textkompression

aaaaaaaa Lauflängenkodierung: **8a**

a 00	96
b 01	24
c 10	32
d 11	48



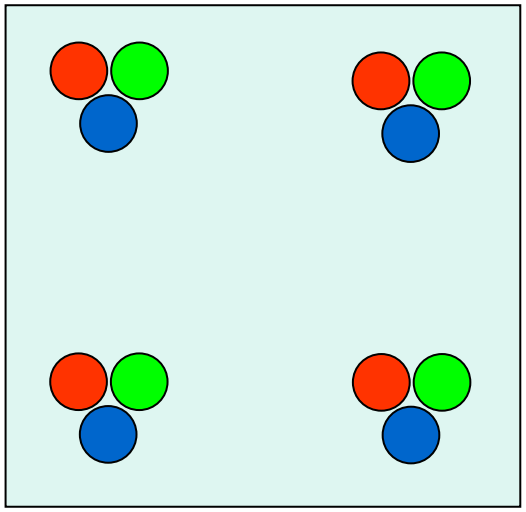
Huffmankodierung:

a 0	
b 100	
c 101	
d 11	

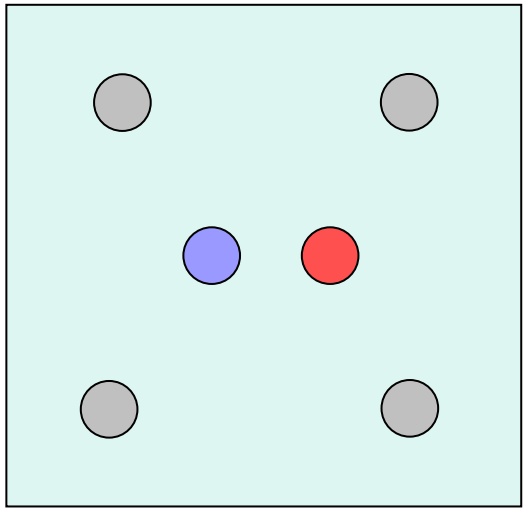
JPEG

- Transformation nach YUV
- Diskrete Cosinus-Transformation
- Quantisierung
- Lauflängenkodierung
- Huffmankodierung

Transformation nach YUV



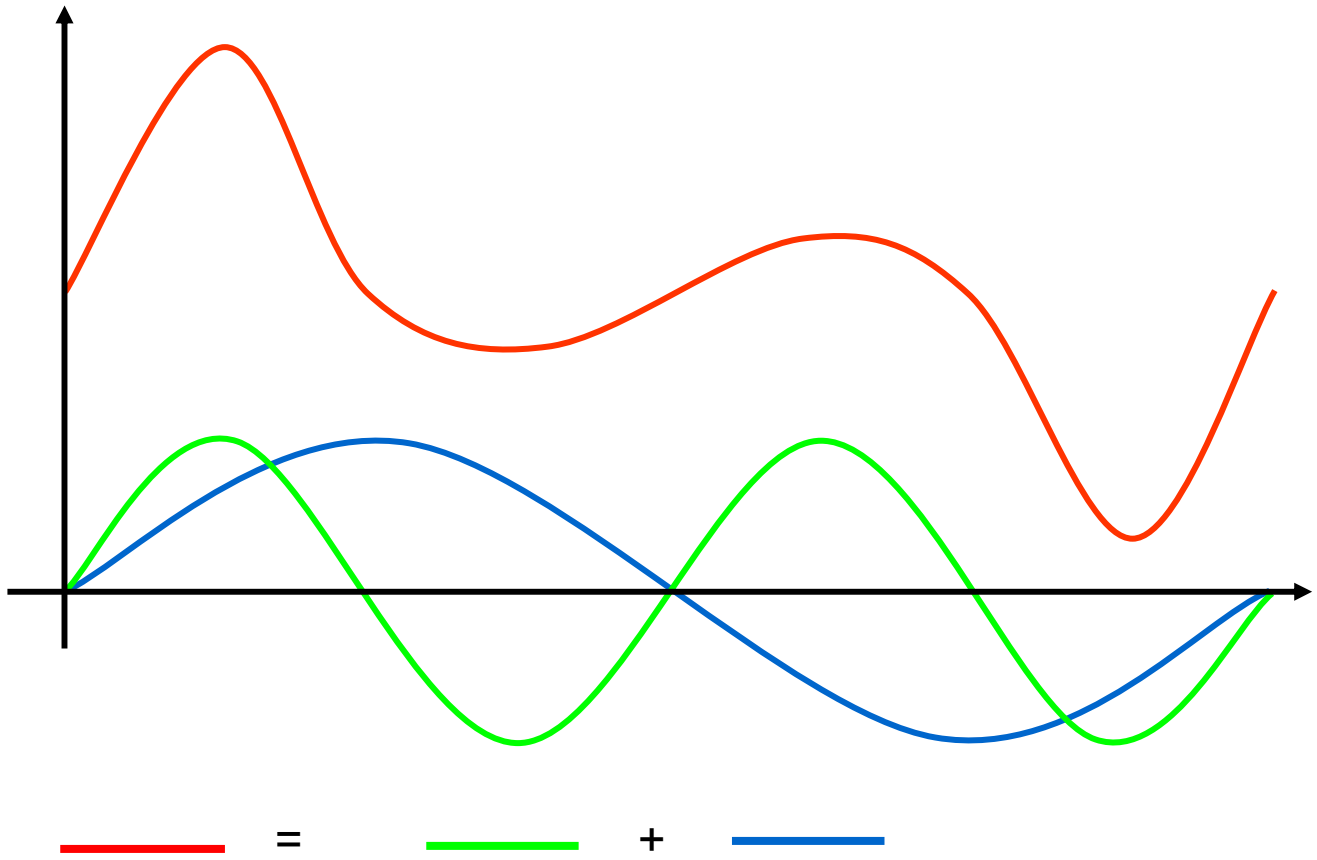
12 Farbwerte



4 Grauwerte
2 Farbdifferenzen

Zerlegung einer Schwingung

127 235 127 112 127 142 127 20 127



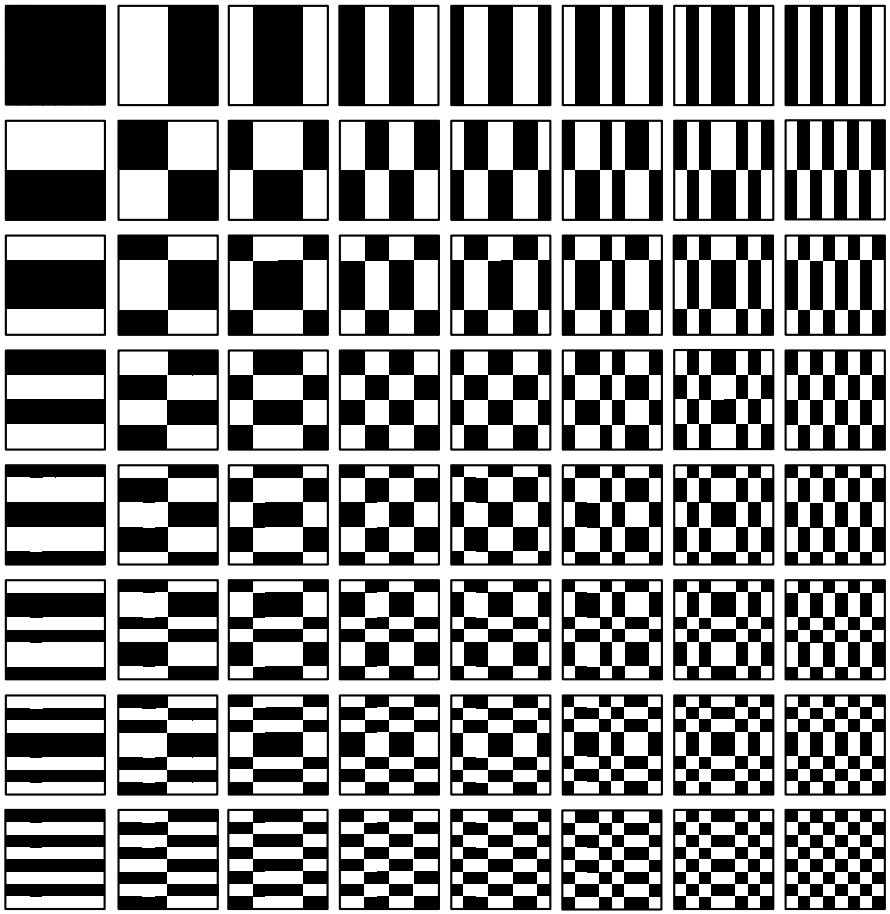
Diskrete Cosinus-Transformation

$$s[u,v] := \frac{1}{4} \cdot c_u \cdot c_v \cdot \sum_{x=0}^{7} \sum_{y=0}^{7} f[x,y]$$

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

$$c_u, c_v := \begin{cases} \frac{1}{\sqrt{2}} & u,v = 0 \\ 1 & \text{sonst} \end{cases}$$

Veranschaulichung der DCT



Ausgangsmatrix

A:

95	88	87	95	88	95	95	95
143	144	151	151	153	170	183	181
153	151	162	166	162	151	126	117
143	144	133	130	143	153	159	175
123	112	116	130	143	147	162	189
133	151	162	166	170	188	166	128
160	168	166	159	135	101	93	98
154	155	153	144	126	106	118	133

verschiebe nach [-128..127]

wende DCT an

M:

```
0.353553  0.353553  0.353553  0.353553  0.353553  0.353553  0.353553  0.353553
0.490393  0.415735  0.277785  0.097545 -0.097545 -0.277785 -0.415735 -0.490393
0.461940  0.191342 -0.191342 -0.461940 -0.461940 -0.191342  0.191342  0.461940
0.415735 -0.097545 -0.490393 -0.277785  0.277785  0.490393  0.097545 -0.415735
0.353553 -0.353553 -0.353553  0.353553  0.353553 -0.353553 -0.353553  0.353553
0.277785 -0.490393  0.097545  0.415735 -0.415735 -0.097545  0.490393 -0.277785
0.191342 -0.461940  0.461940 -0.191342 -0.191342  0.461940 -0.461940  0.191342
0.097545 -0.277785  0.415735 -0.490393  0.490393 -0.415735  0.277785 -0.097545
```

$$B := M \cdot A \cdot M^T$$

Frequenzkoeffzienten

B:

DC

AC

93	2	-8	-7	3	1	1	-2
-38	-58	11	17	-3	5	5	-3
-84	63	-1	-17	2	7	-4	-0
-51	-37	-10	13	-10	5	-1	-4
-85	-42	50	-8	18	-5	-1	1
-63	66	-13	-1	2	-6	-2	-2
-16	14	-37	18	-12	4	3	-3
-53	31	-7	-10	23	-0	2	2

Quantisierungsmatrix

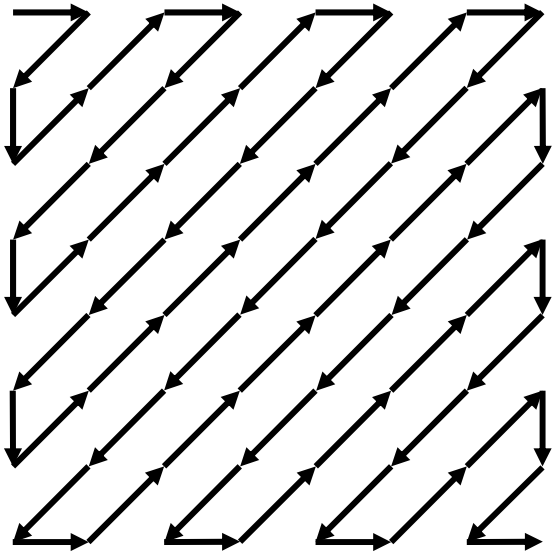
Q:

3	5	7	9	11	13	15	17
5	7	9	11	13	15	17	19
7	9	11	13	15	17	19	21
9	11	13	15	17	19	21	23
11	13	15	17	19	21	23	25
13	15	17	19	21	23	25	27
15	17	19	21	23	25	27	29
17	19	21	23	25	27	29	31

$$r[u,v] := \left\lfloor \frac{b[u,v]}{q[u,v]} \right\rfloor$$

quantisierte Koeffizienten

31	0	-1	0	0	0	0	0
-7	-8	1	1	0	0	0	0
-12	7	0	-1	0	0	0	0
-5	-3	0	0	0	0	0	0
-7	-3	3	0	0	0	0	0
-4	4	0	0	0	0	0	0
-1	0	-1	0	0	0	0	0
-3	1	0	0	0	0	0	0



11,2,-3

i: Zahl der Nullen bis ungleich 0

j: Zahl der erforderlichen Bits

k: Wert mit j Bits kodiert

Huffman-Kodierung

EOB	1010
0/1	00
0/2	01
0/3	100
0/4	1011
0/5	11010
0/6	1111000
0/7	11111000
0/8	1111110110
0/9	1111111110000010
...	
1/1	1100
1/2	11011
1/3	1111001
...	
11/1	1111111001
11/2	1111111111010000

7	111
6	110
5	101
4	100
3	11
2	01
1	1
-1	0
-2	10
-3	00
-4	011
-5	010
-6	001
-7	000

1111111111010000 00

Ausgangsmatrix und Ergebnis

```
01011111 01011000 01010111 01011111 01011000 01011111 01011111 01011111
10001111 10010000 10010111 10010111 10011001 10101010 10110111 10110101
10011001 10010111 10100010 10100110 10100010 10010111 01111110 01110101
10001111 10010000 10000101 10000010 10001111 10011001 10011111 10101111
01111011 01110000 01110100 10000010 10001111 10010011 10100010 10111101
10000101 10010111 10100010 10100110 10101010 10111100 10100110 10000000
10100000 10101000 10100110 10011111 10000111 01100101 01011101 01100010
10011010 10011011 10011001 10010000 01111110 01101010 01110110 10000101
```

DC:

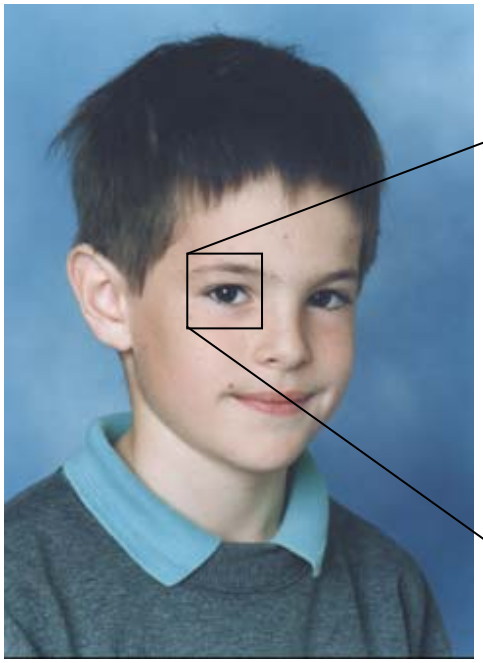
```
00011111
```

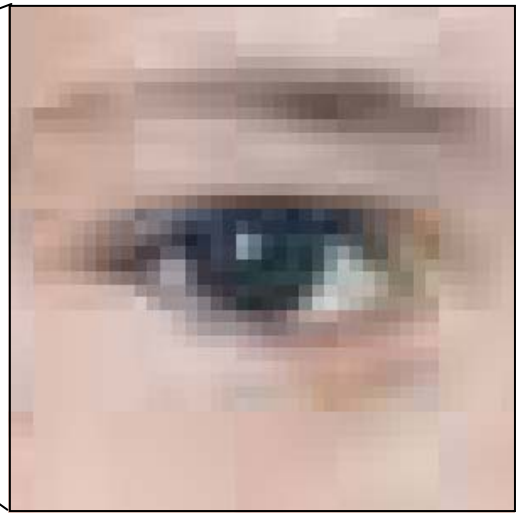
AC:

```
11110010001011001110110111000110011001111000101000000010011001
11101001101100100011000100100011111111111111010000000010001010
```

vorher - nachher

Adobe Photoshop





jan.tif 239 KB jan.jpg 9 KB

Kompressionsfaktor 26