Images

- Licht
- Auge, Retina
- Farbwahrnehmung
- Farbmodelle: RGB, HSB, YUV

- Rasterbilder
- Bildformate PNM, BMP, GIF, PNG
- Filter und elementare Algorithmen

- JPEG-Kodierung
- JPEG-2000
- Wavelets

- (Vektorgraphik, Animationen)

Literatur

Foley, van Dam, Feiner: Computer Graphics: Principles and Practice, Addison-Wesley

B. Jähne, Digitale Bildverarbeitung, Springer 1997, 3-540-61379-X

D. Salomon, Data Compression, Springer 2000,

Skripte Bildverarbeitung, Farbbildverarbeitung, ...

Proceedings of the IEEE, Vol. 86-4, 1998, "multimedia signal processing"

netpbm.sourceforge.net, www.acme.com/software/pbmplus

www.jpeg.org, jj2000.epfl.ch, www.libpng.org

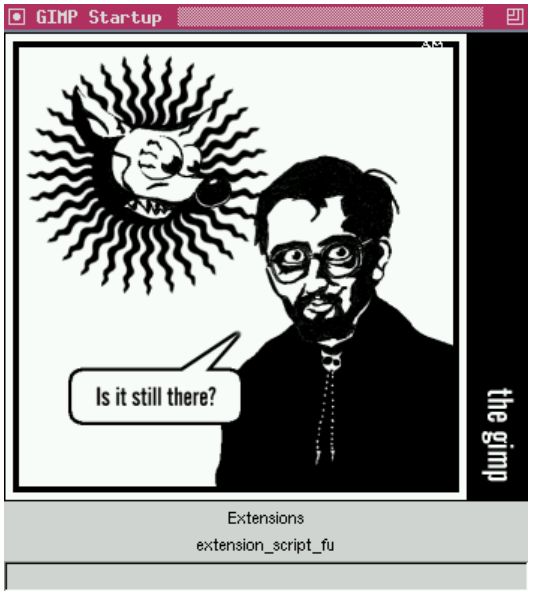
www.gimp.org, www.adobe.com/products/photoshop/main.html

Warnung

- Themen Computergraphik und Bildverarbeitung hier nur angerissen

- Grundlagen der Rastergraphik
- Datenformate
- Grundlage für Videokodierung

- keine mathematischen Grundlagen
- keine Algorithmen, Filter, ...
- keine Computergraphik, ...
- keine Details zu Applikationen



Vertiefung siehe Vorlesungen und Seminare:

- Bildverarbeitung (KOGS, IMA, ...)
- Computergraphik (TIS, ...)

Thema nur angerissen:

Inhaltsverzeichnis aus Foley et.al.: "Computer Graphics" :=

- simple raster graphics
- graphics algorithms for drawing 2D primitives
- geometrical transformations
  - viewing in 3D
  - representing curves and surfaces
  - object hierarchy and PHIGS, solid modeling
  - visual realism, illumination and shading

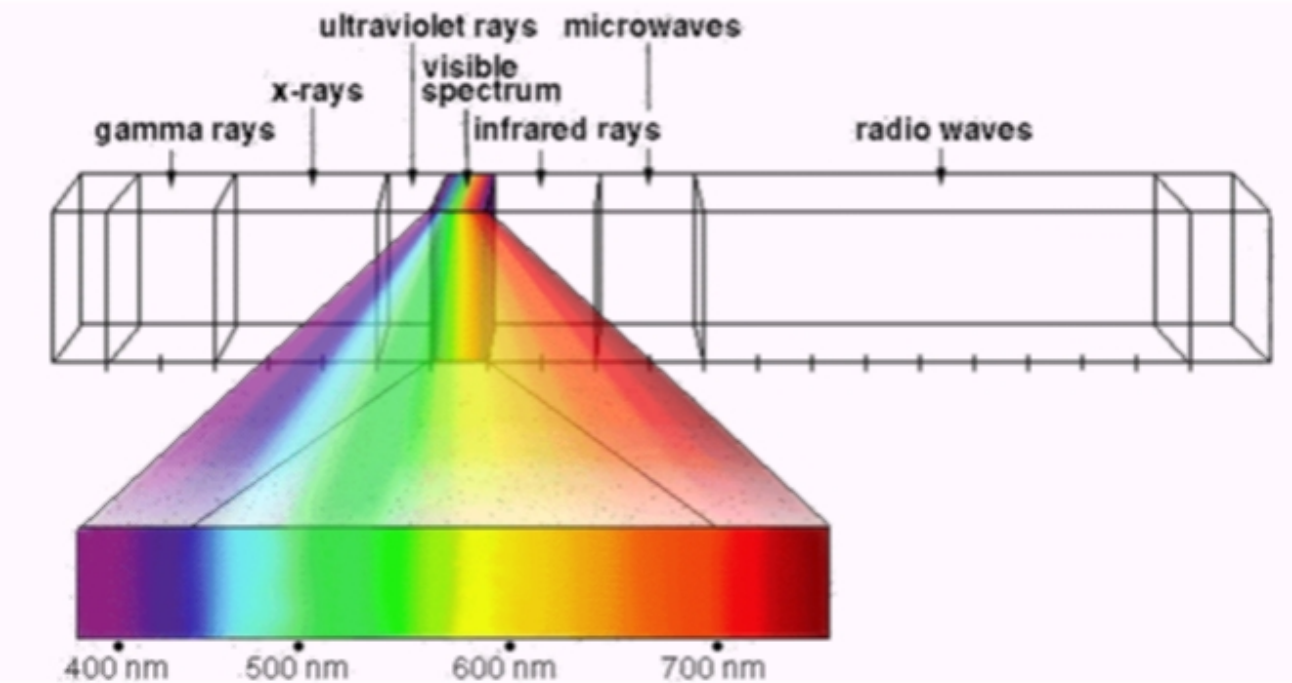
  . . .
- image manipulation and storage
  - user interface design
  - animation
- graphics hardware, input devices

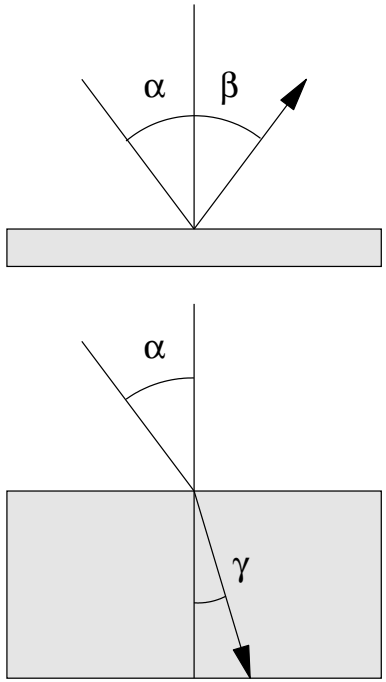
Thema nur angerissen:

Inhaltsverzeichnis aus Jähne, "Digitale Bildverarbeitung" :=

- Beleuchtung und Reflexion
- Abbildung, Perspektive, Digitalisierung
- Unitäre Transformationen und Bildrepräsentation
- Statistik: Bilder als stochastische Prozesse
- Einfache Bildstrukturen und Filter
- Orientierung und adaptive Filterung
- Segmentierung und Klassifizierung
- Rekonstruktion aus Projektionen
- Bewegung, Verschiebungsvektoren
- Bewegung in Orts-Zeit-Bildern
- ...

Licht:

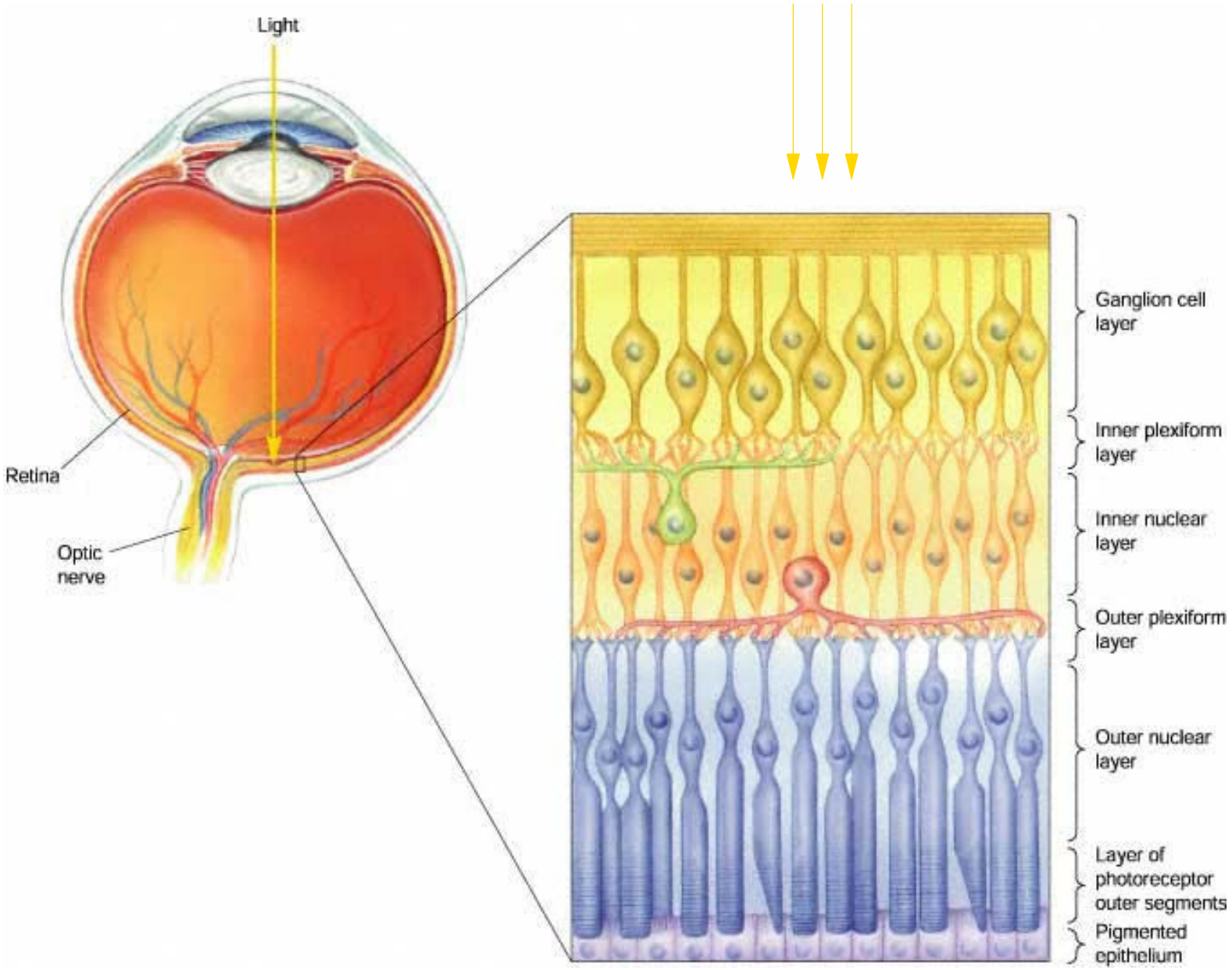




- Licht: elektromagnetische Wellen im Bereich 400 .. 700 nm
- Brechung, Reflektion, Beugung
- Quantenmechanik: Strahlung des "schwarzen Körpers"
- Intensitätsverteilung abhängig von Temperatur: "Farbtemperatur"

  z.B. Sonnenlicht 5500K, Glühbirne 60W 2200K, Leuchtstoffröhre 4400K

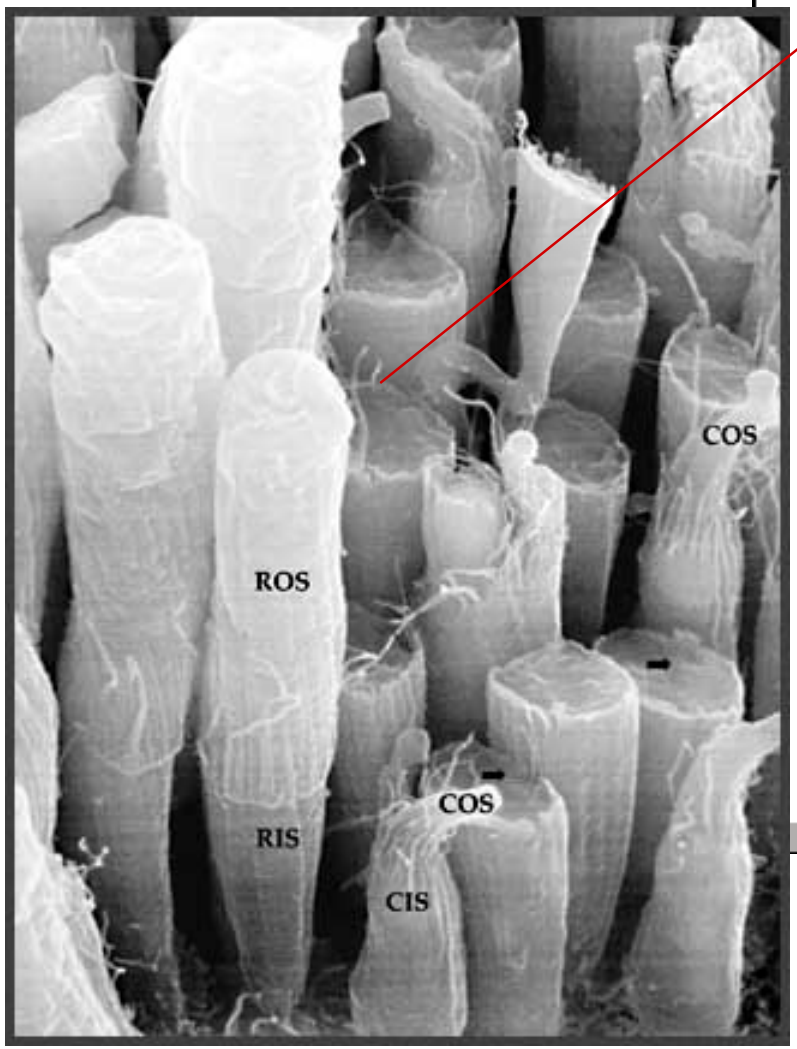
Auge: Aufbau

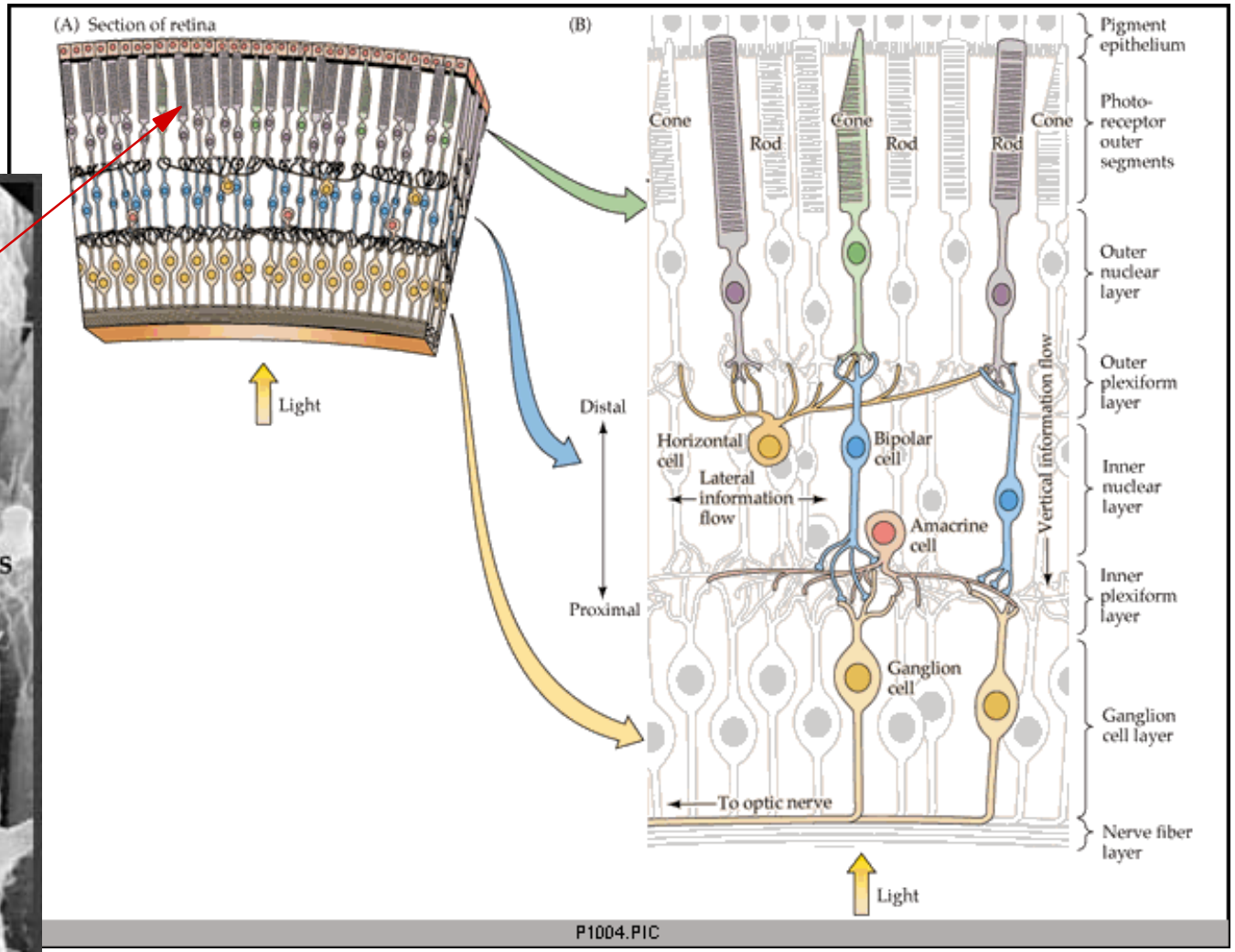

Light
Retina
Optic nerve
Ganglion cell layer
Inner plexiform layer
Inner nuclear layer
Outer plexiform layer
Outer nuclear layer
Layer of photoreceptor outer segments
Pigmented epithelium

Auge: Retina

Stäbchen, "rods"

Zäpfchen, "cones"





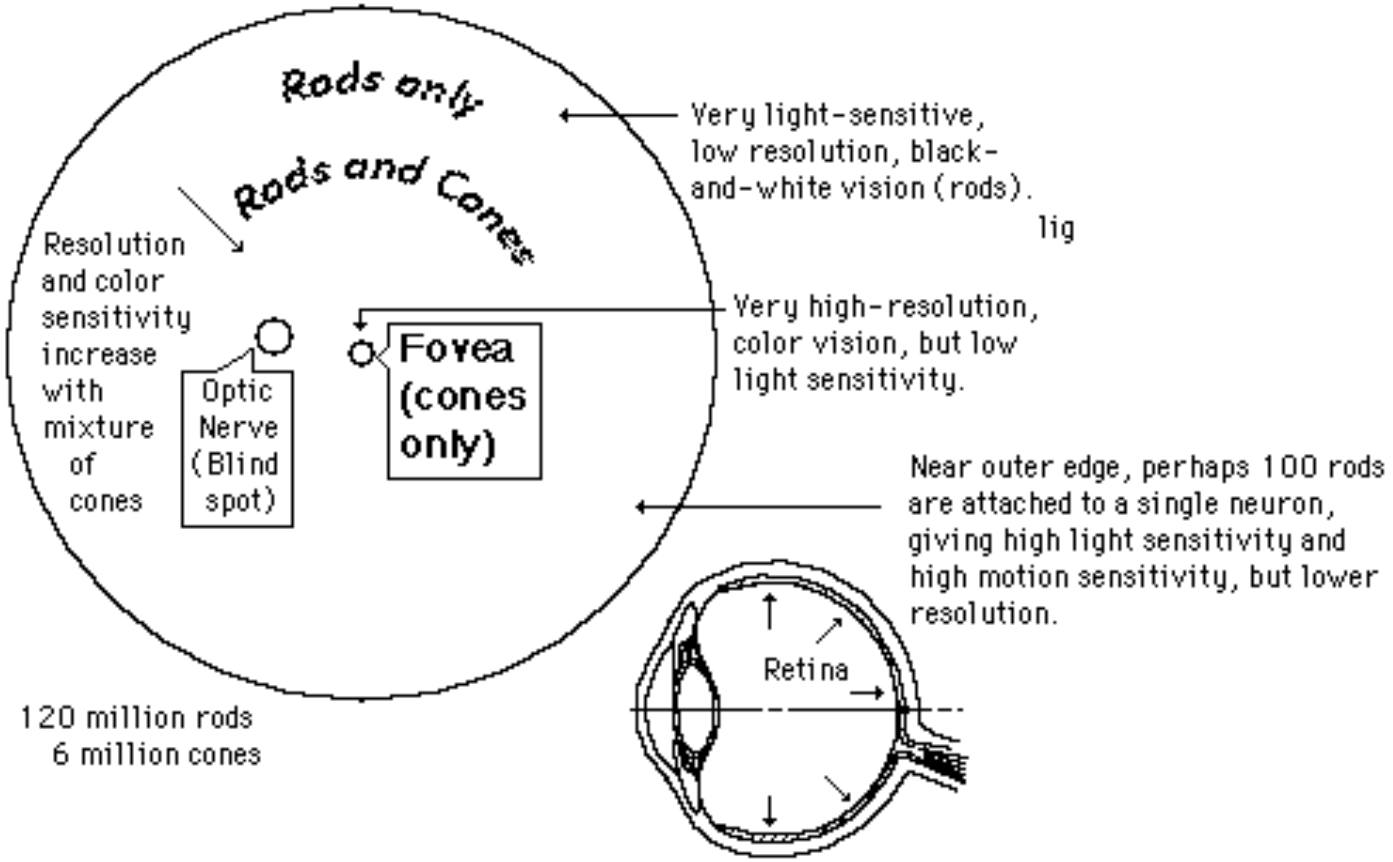
- "festverdrahtete" Vorverarbeitung
- Ecken-, Kanten-, Bewegungserkennung

Retina: Auflösung

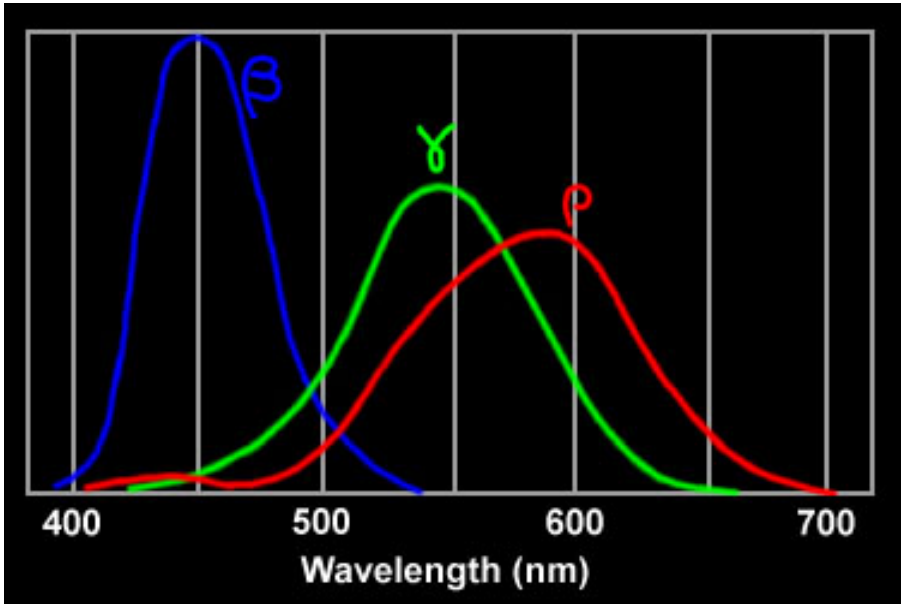
Aufteilung der Zäpfchen:

4% blau (außen), 32% grün, 64% rot

"6 Megapixel"



Auge: Farbempfindung



spektrale Empfindlichkeit der
blauen, grünen, roten Zäpfchen

Farb-Vorverarbeitung in Retina und Sehnerv:

R - G	Differenzsignal aus (rot - grün)
Y = R + G	Helligkeits- und Gelbwahrnehmung
Y - B	Differenz (gelb - blau) zur Gelb/Blau-Unterscheidung

=> ca. 128 Farbtöne, 130 Stufen Farbsättigung, 16..26 Helligkeiten

Farbe: Richtlinien

Aufbau der Retina, Vorverarbeitung: (R-G), Y=(R+G), (Y-B)

- möglichst nicht mehrere gesättigte Farben
- möglichst kein blauer Text oder feine blaue Linien:
  (weil keine blauempfindlichen Zäpfchen in der Fovea)
- möglichst keine roten/grünen Details am Rand von Bildern
  (weil außen an der Retina keine rot/grünempfindlichen Zäpfchen)
- keine roten Zeichen vor blauem Hintergrund
- keine benachbarten Farben, die sich nur im Blauanteil unterscheiden
- Verwendung von Farbe als einziges Mittel zur Kodierung
- usw.

(Henning 4.2.4)

Helligkeitsempfindung

- Helligkeitsempfindung des Auges ist nichtlinear
- experimentelle Resultate:
  gerade noch erkennbare Helligkeitsdifferenz: 1..2 %

- "Weber-Fechnersches-Gesetz" durch Integration:

$$E = k \cdot \log(L / L_0) + c \qquad \text{Leuchtdichte } L$$

- bessere Approximation durch "Stevenssche Potenzfunktion":

$$E = k \cdot (L - L_0)^n \qquad \text{mit } n \approx 0.33$$

- unempfindlich gegen Rauschen bei geringer / hoher Leuchtdichte

(Hentschel)

Visuelles System:

Wahrnehmung komplexer Bildinhalte?

- low-level Experimente liefern nur Einzelaussagen
- aber außerordentliche leistungsfähige Verarbeitung im Gehirn
- "visuelles System"

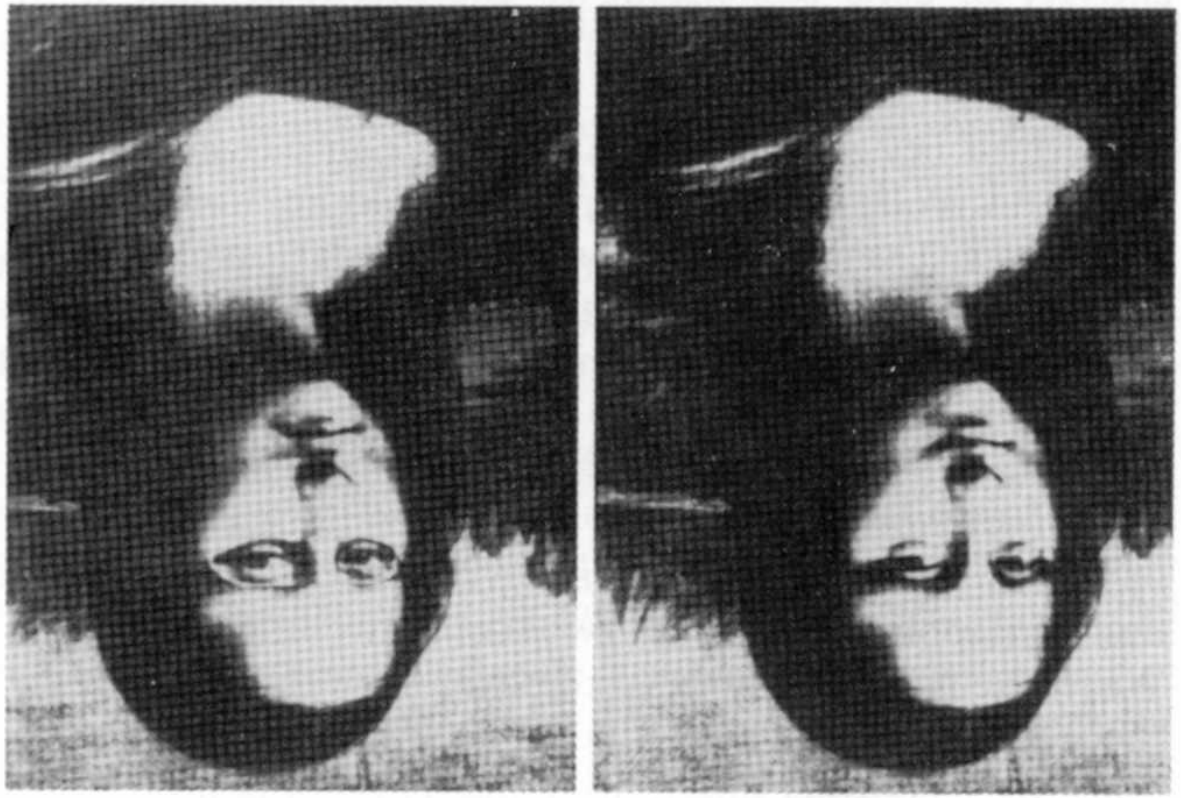
- bisher kein vollständiges Modell des visuellen Systems
- Demonstration z.B. mit 'optischen Täuschungen'
- siehe folgende Seiten

- entsprechend kein Qualitätsmodell für Bildkodierung
- ad-hoc Annahmen
- Beurteilung durch Testpersonen

(Hentschel)

Visuelles System: Vorverarbeitung ...

FIGURE 30–23
These pictures appear similar at first glance because only our preattentive process is active. When the pictures are seen upright, the true detail in the two faces is revealed. (From Julesz, 1986, after an idea of Thompson, 1980.)



(Kandell, Principles of Neural Science)

Visuelles System:

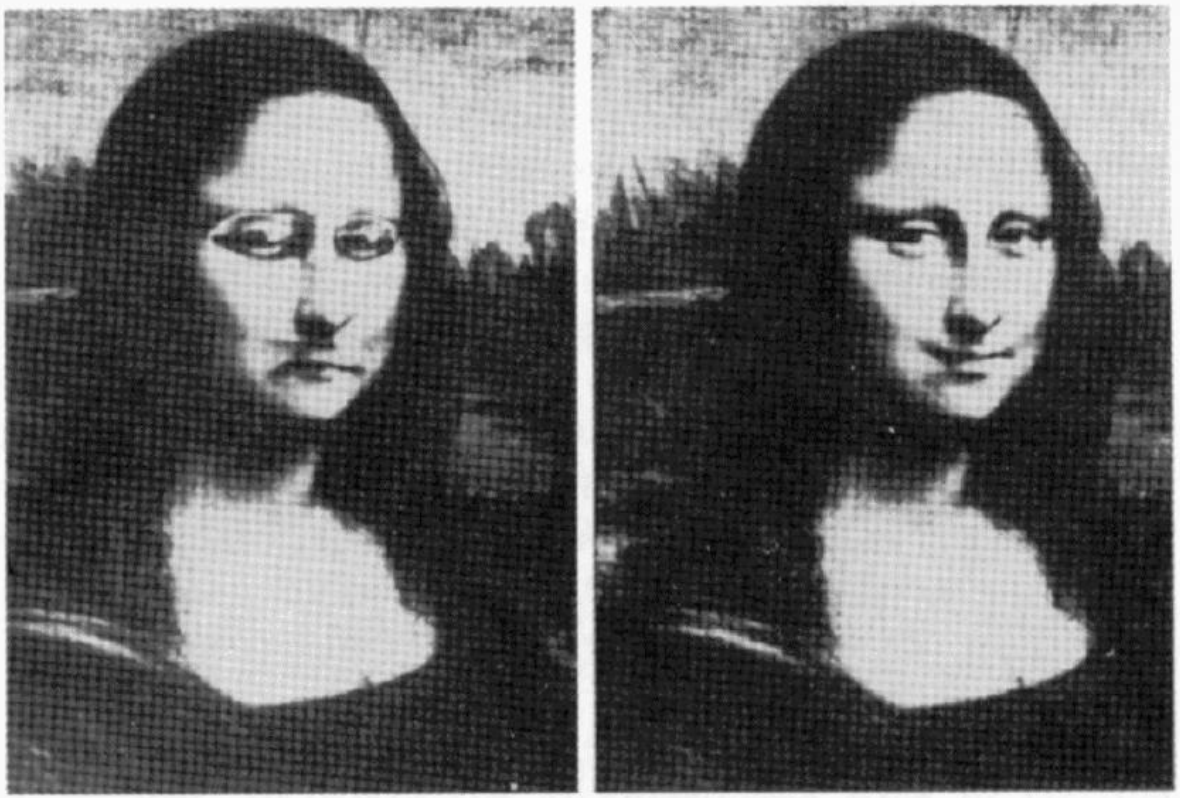
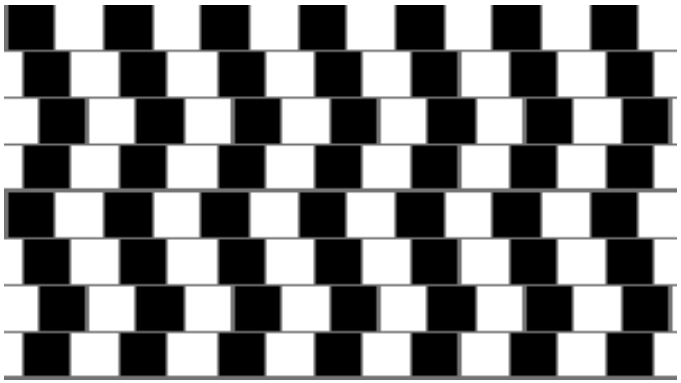
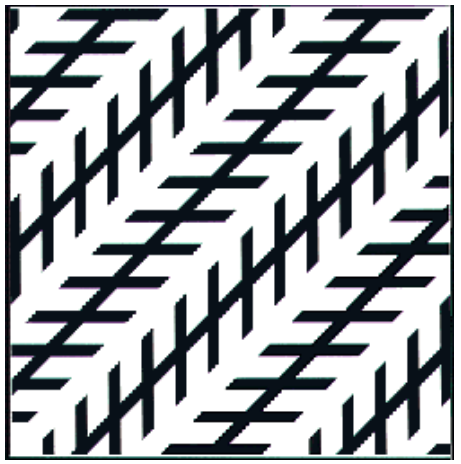


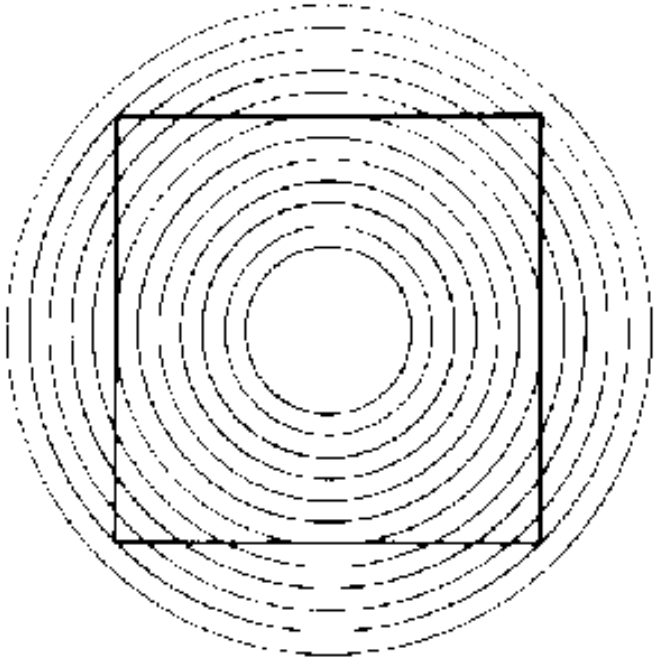
FIGURE 30–53
These pictures appear similar at first glance because only one perceptive process is active. When the pictures are seen superimposed, the true detail in the two faces is revealed. (From Julesz, 1989, after an idea of Thompson, 1980.)

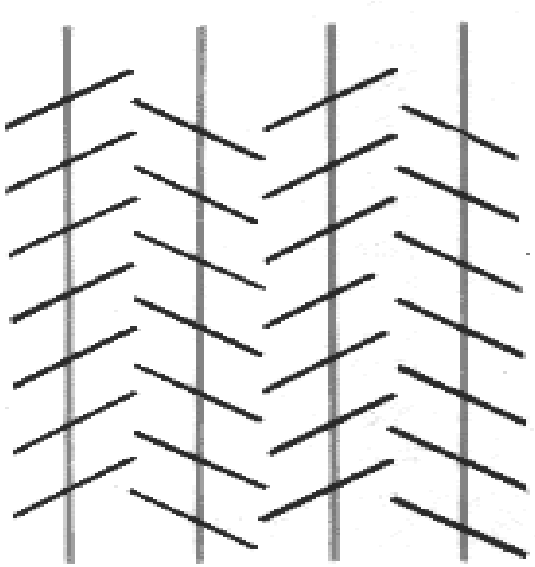
(Kandell, Principles of Neural Science)

Visuelles System: Parallele Linien ...









Visuelles System: Kantendetektion

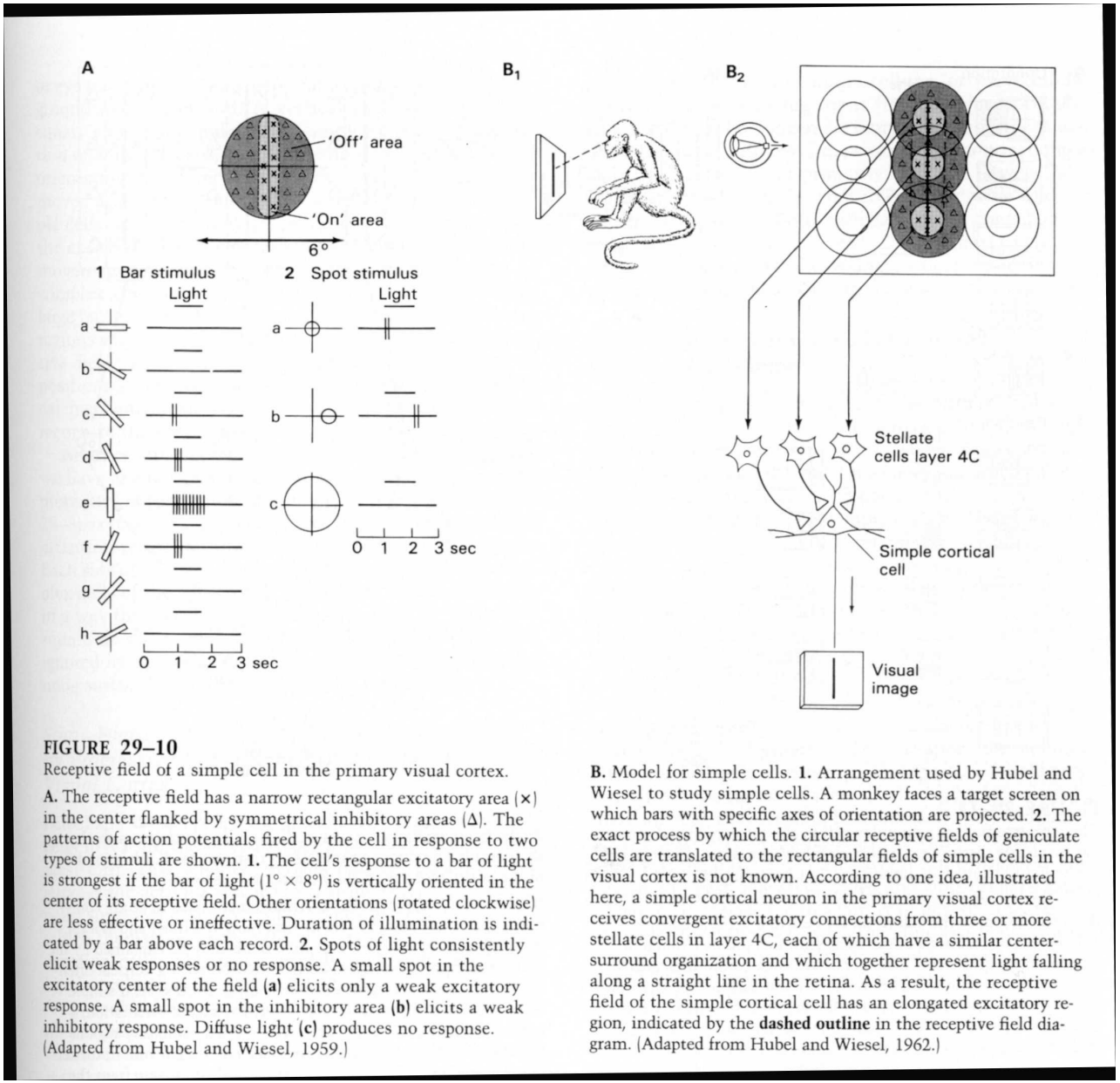


FIGURE 29–10
Receptive field of a simple cell in the primary visual cortex.

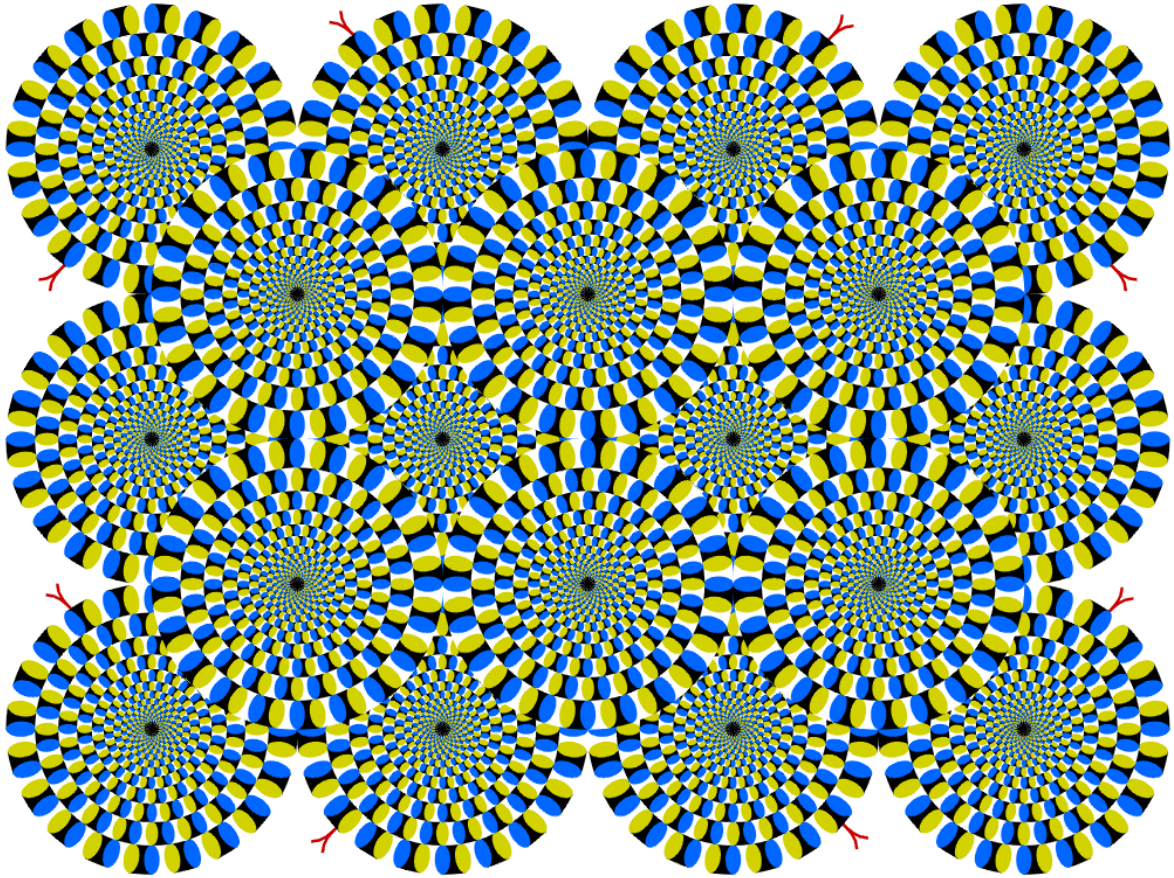
A. The receptive field has a narrow rectangular excitatory area (×) in the center flanked by symmetrical inhibitory areas (Δ). The patterns of action potentials fired by the cell in response to two types of stimuli are shown. **1.** The cell's response to a bar of light is strongest if the bar of light (1° × 8°) is vertically oriented in the center of its receptive field. Other orientations (rotated clockwise) are less effective or ineffective. Duration of illumination is indicated by a bar above each record. **2.** Spots of light consistently elicit weak responses or no response. A small spot in the excitatory center of the field **(a)** elicits only a weak excitatory response. A small spot in the inhibitory area **(b)** elicits a weak inhibitory response. Diffuse light **(c)** produces no response. (Adapted from Hubel and Wiesel, 1959.)

B. Model for simple cells. **1.** Arrangement used by Hubel and Wiesel to study simple cells. A monkey faces a target screen on which bars with specific axes of orientation are projected. **2.** The exact process by which the circular receptive fields of geniculate cells are translated to the rectangular fields of simple cells in the visual cortex is not known. According to one idea, illustrated here, a simple cortical neuron in the primary visual cortex receives convergent excitatory connections from three or more stellate cells in layer 4C, each of which have a similar center-surround organization and which together represent light falling along a straight line in the retina. As a result, the receptive field of the simple cortical cell has an elongated excitatory region, indicated by the **dashed outline** in the receptive field diagram. (Adapted from Hubel and Wiesel, 1962.)

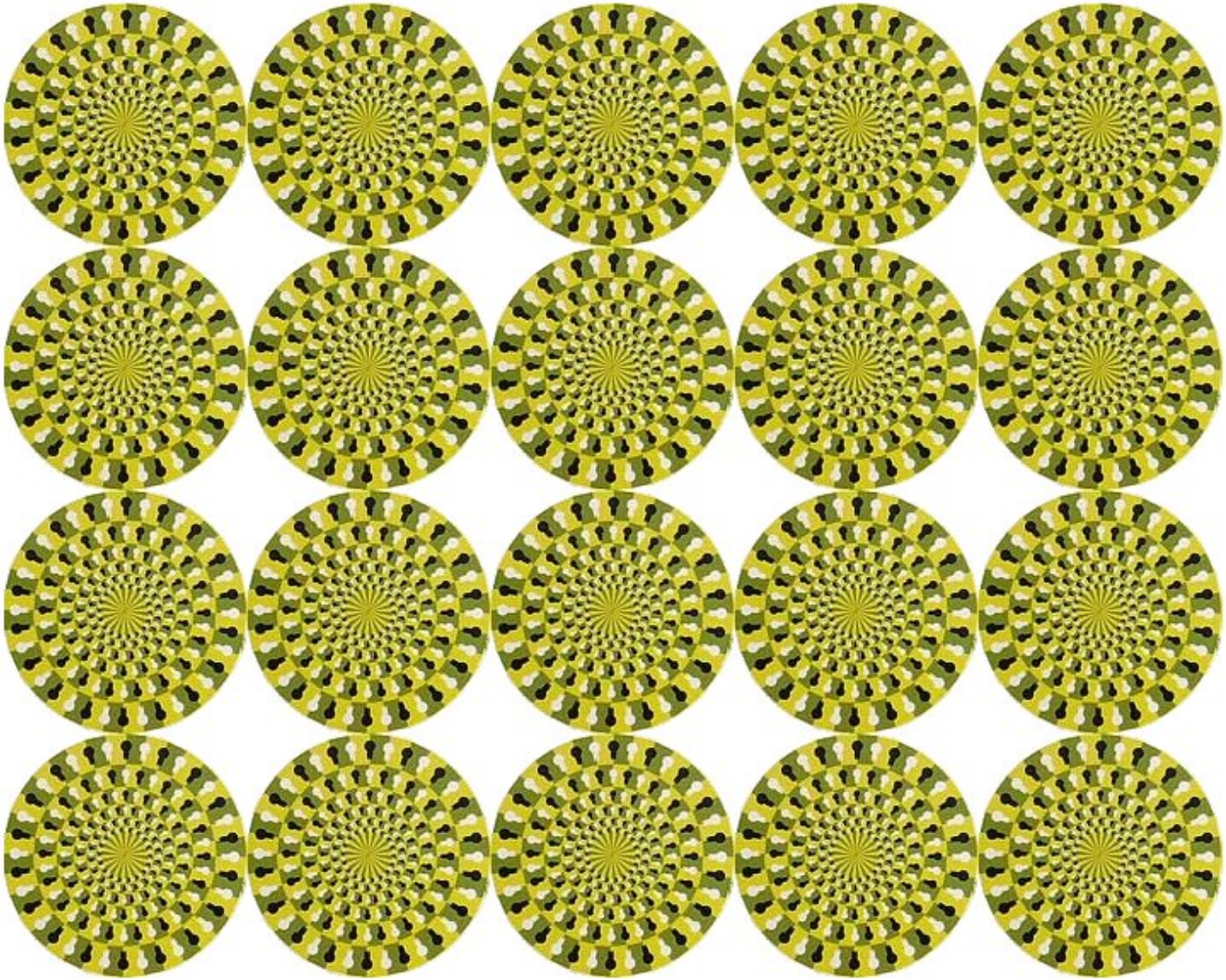
"receptive fields"

(Kandell, Principles of Neural Science)

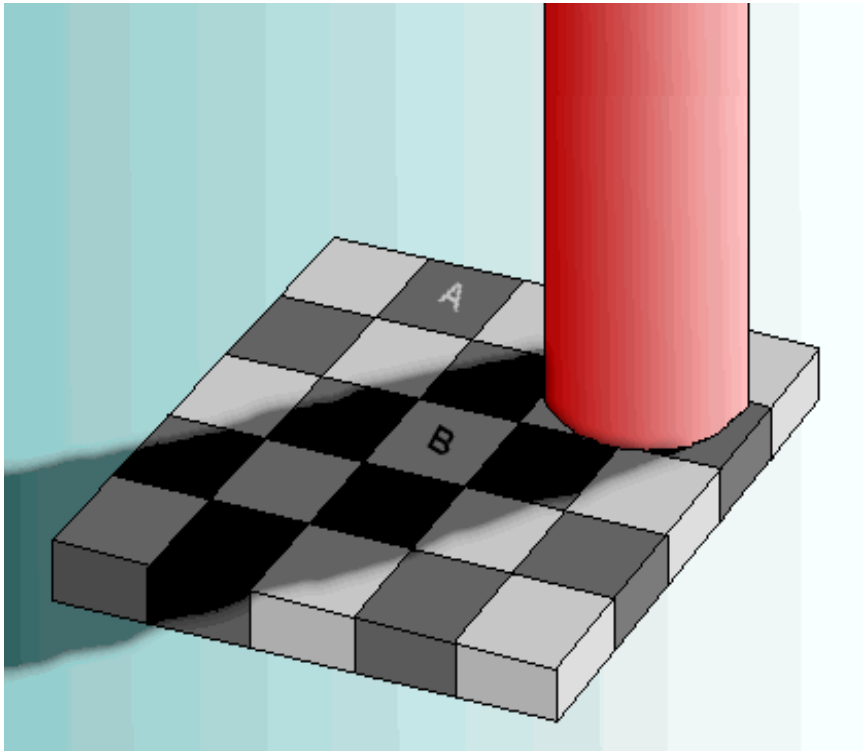
Visuelles System:

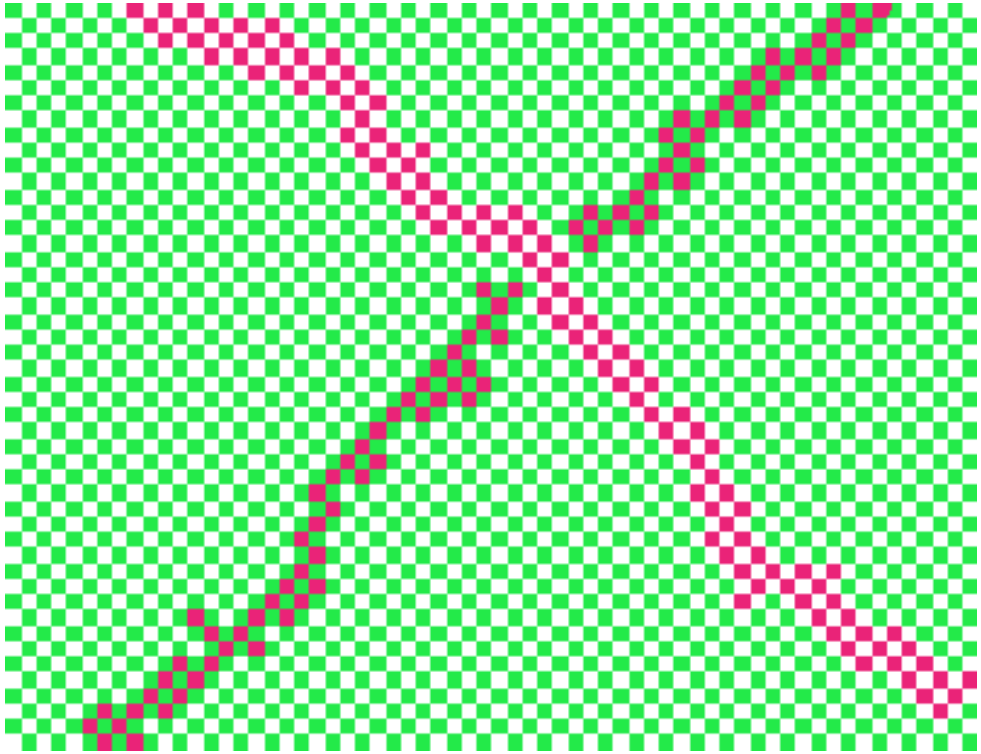


Visuelles System



Visuelles System: Helligkeit ...

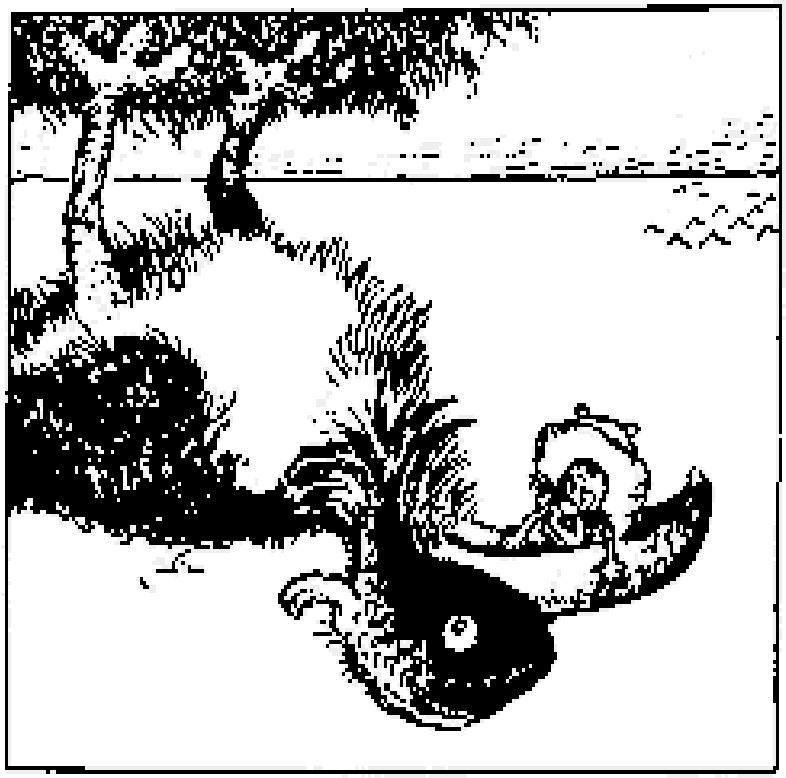




Bereiche "A" und "B" haben denselben Grauwert

aber Analyse und Interpretation durch das Gehirn ("Schatten")

Visuelles System: Interpretation ...



Visuelles System: Größentäuschung

FIGURE 30–5
To judge size, we unconsciously compare the various objects in the visual field. In the picture on the **left** the nearer woman is 9 feet from the camera while the farther one is 27 feet away. Both appear to be the same size. The picture on the **right** was taken with the nearer woman in the same place but without the farther woman. The farther woman in the left picture was then cut out and pasted on the picture at right. To convince yourself that she is the same size as the one at left you will probably need to measure her. In the right picture she appears as small, not as being far away, because the corridor and tiles around her are not proportional, as they are in the left picture. (From Brown and Herrnstein, 1975.)

(Kandell, Principles of Neural Science)

Visuelles System: Konflikt ...

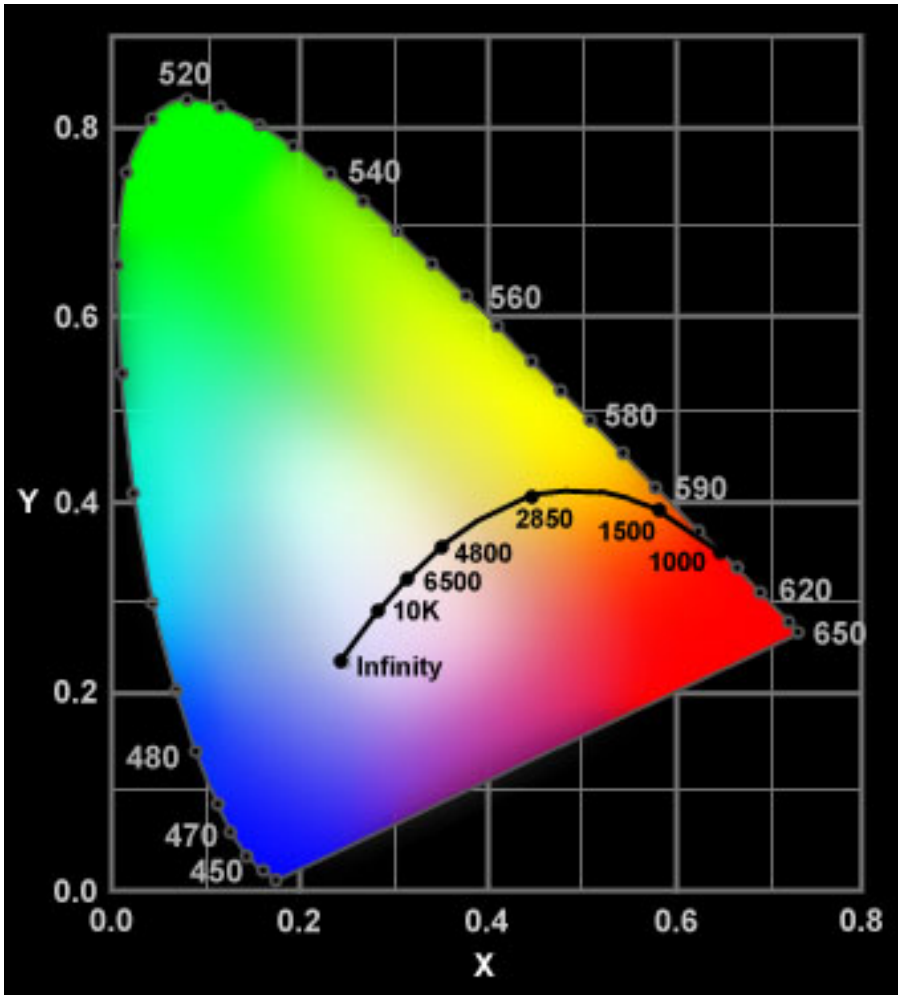
Look at the chart and say the COLOUR not the word

YELLOW BLUE ORANGE
BLACK RED GREEN
PURPLE YELLOW RED
ORANGE GREEN BLACK
BLUE RED PURPLE
GREEN BLUE ORANGE

Left – Right Conflict

Your right brain tries to say the colour but your left brain insists on reading the word.

Farbmodelle



CIE = comm. int. de l'elcairage

non-profit, gegründet 1913

diverse Farbmodelle / -Standards

Grundkoordinaten X, Y, Z

aufbauend auf Lichtempfindlichkeit

der blauen/grünen/roten Zäpfchen

"normierte" Koordinaten (x,y,z):

$x = X / (X+Y+Z)$

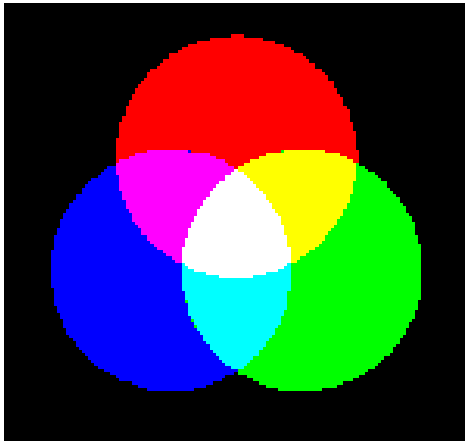
$y = Y / (X+Y+Z)$

$z = 1-x-y$

monochromatisches Licht:	liegt im CIE-Diagramm am Rand
schwarzer Körper (weiß):	im Zentrum, abhängig von Farbtemperatur

RGB: Additive Farbmischung

"Red Green Blue"



- Modell der additiven Farbmischung
- aus den Grundfarben rot, grün, blau

  zyan = blau + grün

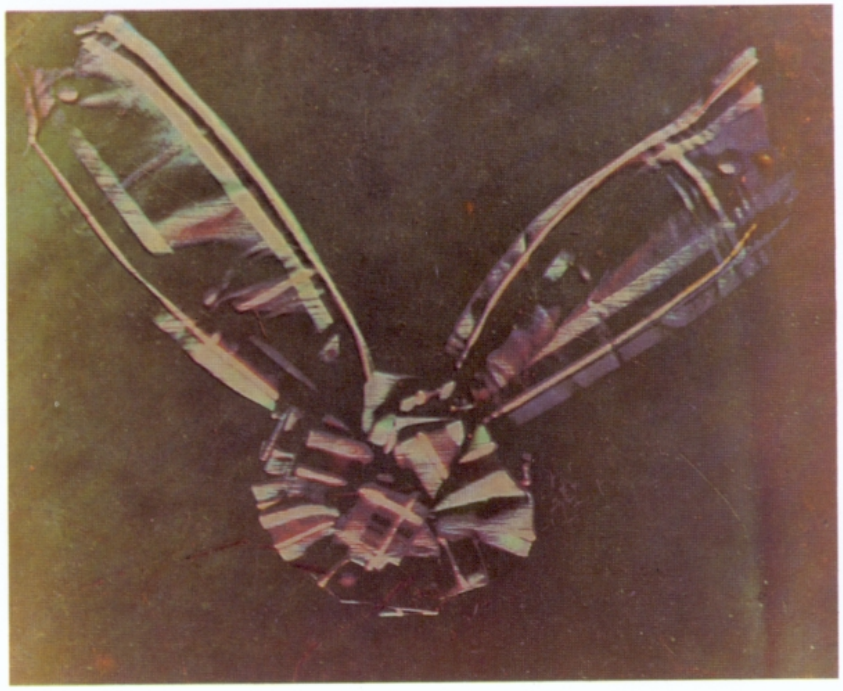
  gelb = rot + grün

  magenta = rot + blau

  weiß = rot + blau + grün

- Anwendung beim Fernseher / Farbmonitor / Drei-Farb-LCDs
- direkte Bearbeitung mit Algorithmen möglich

- deshalb meistens als Datenstruktur verwendet
- üblicher Wertebereich 0 .. 255 für jede Grundfarbe (8 bit)
- Spezialanwendungen (u.a. Medizin) bis ca. 16bit/Farbe

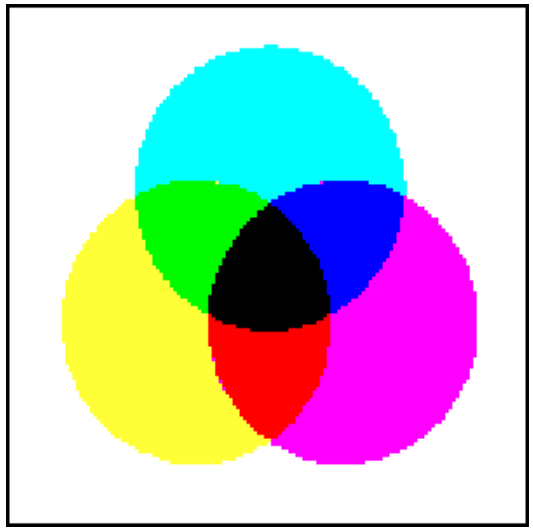


1861 stellte James Clerk Maxwell drei Diapositive eines Ordensbandes mit Streifenmuster her, photographiert durch rote, grüne und blaue Filter. Diese drei Diapositive wurden mit drei Laterna magica-Projektoren durch die gleichen roten, grünen und blauen Filter projiziert. Sich genau überdeckend ergaben die drei Teilbilder eine beachtliche Farbreproduktion des Originals. Dieses war die erste farbige Photographie.

Rechts oben: Ein Ives Kromogram (um 1897), direkt durch den Kromskop-Betrachter photographiert, ein Gerät, in dem die separaten roten, grünen und blauen Teilbilder zu einem Bild mit allen Farben kombiniert wurden.
Rechts unten: Ein 1892 mit dem Ives Photochromiscope-Projektor vorgeführtes Bild, reproduziert als Kodak Dye Transfer Vergrößerung nach den Originaldiapositiven.

(additive Farbmischung, Maxwell, 1861)

CMYK: Subtraktive Farbmischung



"Cyan Magenta Yellow blacK":

- Herausfiltern von Farben aus weissem Licht
- Grundfarben Zyan, Magenta, Gelb

- im Prinzip komplementär zu RGB
- aber unterschiedlicher Farbumfang (mit verfügbaren Farbstoffen)

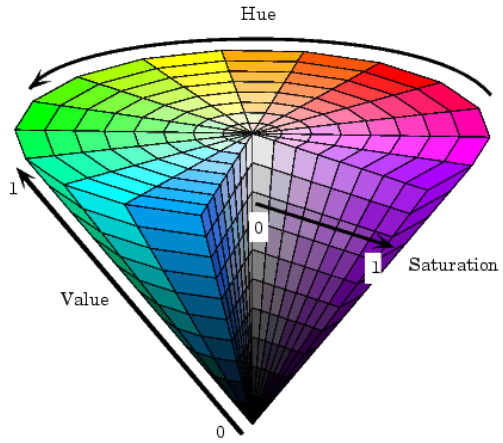
- Anwendung bei reflektierenden Medien (z.B. Papier)
- Farbdruck, "drei-Schichten" Farbfilme, ...

- weitere Farben für größeren Farbumfang
- bzw. um Farben zu sparen (insb. schwarz bei CMYK)

(subtraktive Mischung, 3 Folien, Banfield 1907)

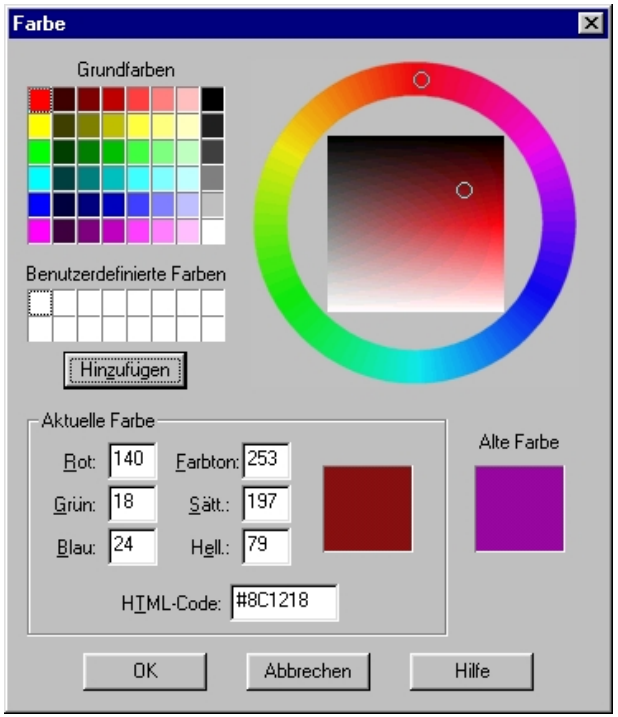
HSB: Hue Saturation Brightness

oder "HSV": hue saturation value



- Auswahl einer Grundfarbe
  (Darstellung als Farbkreis / Pyramide)
- Sättigung
- Helligkeit

- oft als GUI zur Farbauswahl



- Demo (Paint Shop Pro)

YUV

Y = Luminanzwert der CIE-Koordinaten, Wert [0 .. 1]

UV = Farbwerte ("Chrominanz"), Werte [-0.5 .. 0.5]

```
Y  =  0.299 R + 0.587 G + 0.144 B
U  = -0.169 R - 0.331 B + 0.500 B
V  =  0.500 R - 0.419 B - 0.081 B

Cb =  B - Y          U = 0.577 Cb
Cr =  R - Y          V = 0.713 Cr
```

- verwendet beim Farbfernsehen und für digitales Video
- abwärtskompatibel zum S/W-Fernsehen (nur Y)
- Details später, siehe Video

- andere Skalierung von UV: YCbCr-Modell

Umrechnung

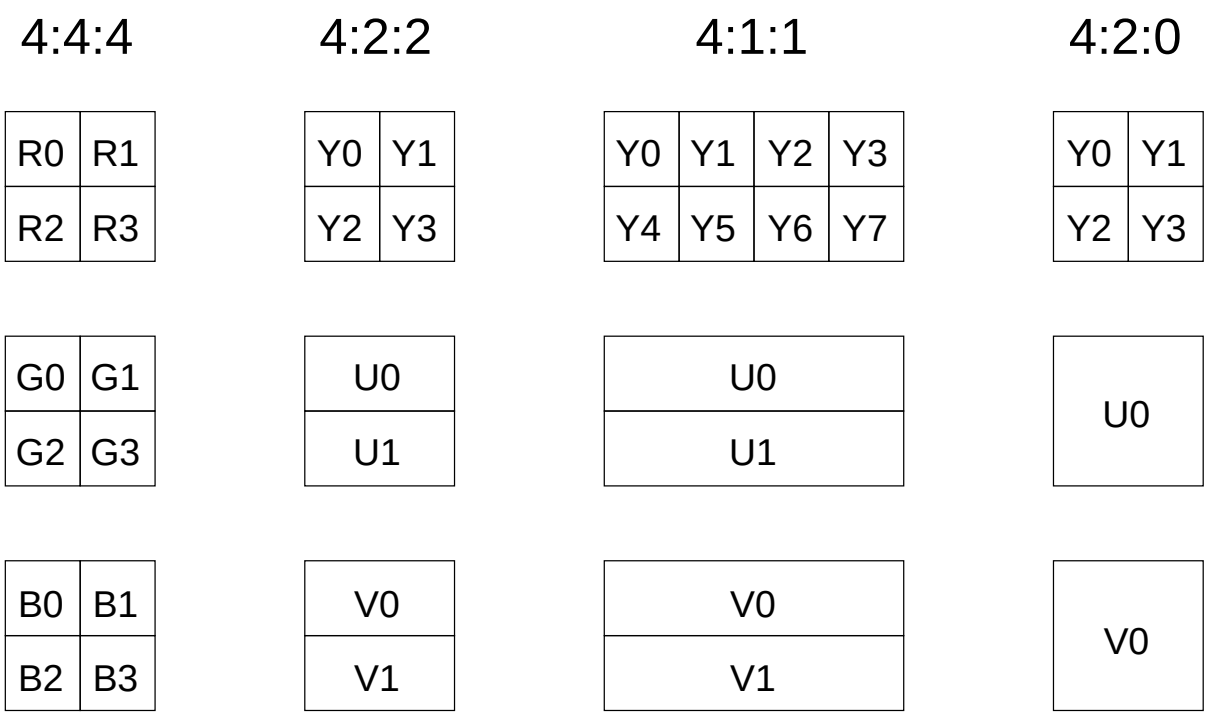
```
C  =  1 - R
M  =  1 - G
y  =  1 - B


K  =  min( C, M, Y )
C' =  C - K
M' =  M - K
Y' =  Y - K



Y  =  0.299 R + 0.587 G + 0.144 B
U  = -0.169 R - 0.331 B + 0.500 B
V  =  0.500 R - 0.419 B - 0.081 B

Cb =  B - Y          U = 0.577 Cb
Cr =  R - Y          V = 0.713 Cr
```

(weitere Details: siehe Henning 4.4.3)

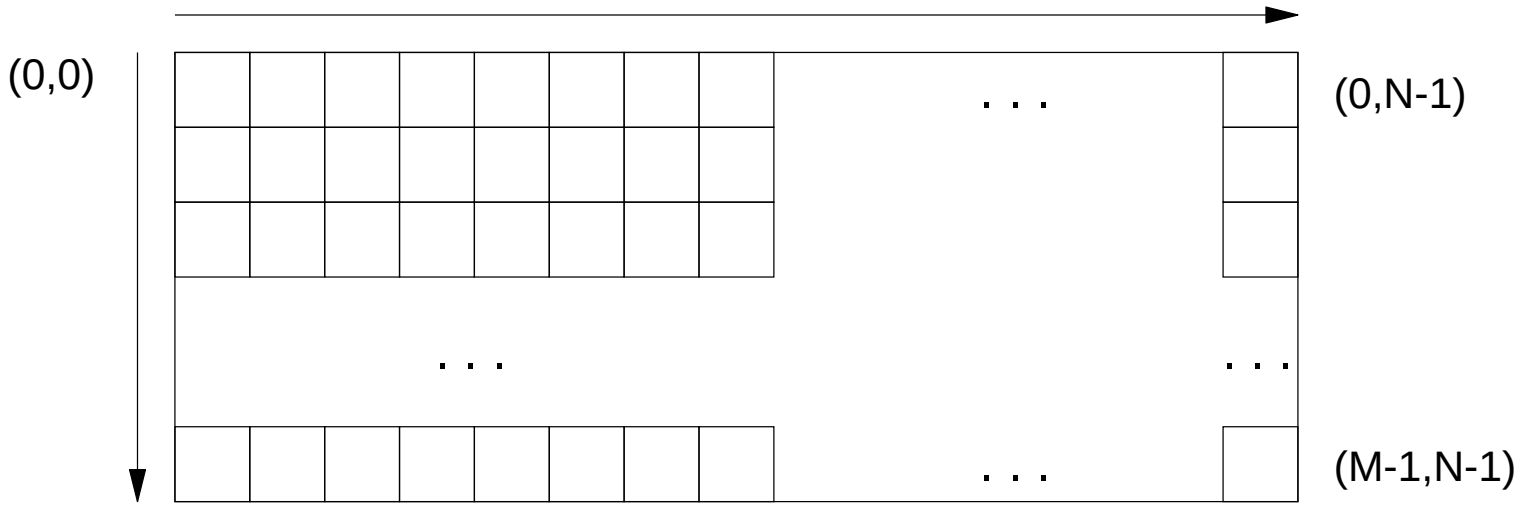
Farb-Subsampling

- Stäbchen und Zäpfchen liefern Helligkeitsinformation
- aber Farbinformation nur aus Zäpfchen

=> in der Bildkodierung und -kompression ausnutzen



Raster-Bildformate



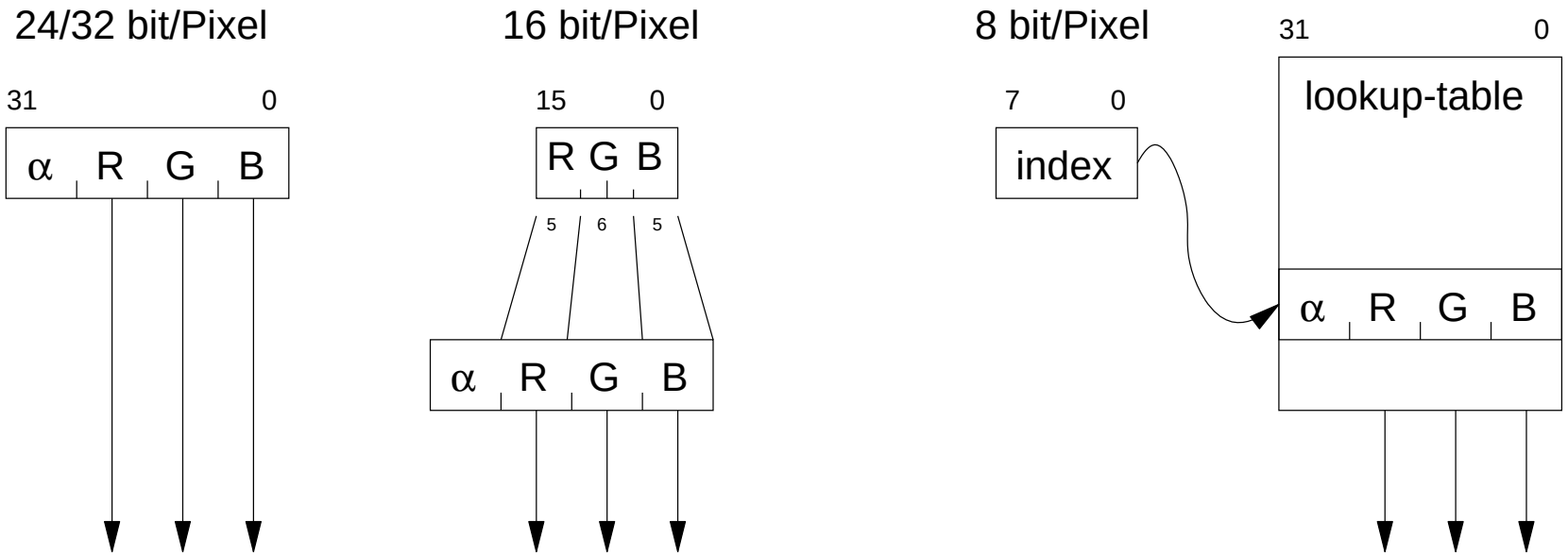
Digitalisierung von Bilddaten:

- rechteckiges Raster, "Pixel" (picture element, picture cell)
- Auflösung gemessen in dpi (dots per inch)
- Farbtiefe (bits per pixel)
- andere Rasterung möglich (z.B. hexagonal: Insekten)

VGA:	640 x 480 x 8	300 KByte
XGA:	1024 x 768 x 24	2.3 MByte
UXGA:	1600 x 1200 x 24	5.7 MByte

- Speicherung als 2D-Array, aber auch 1D-Array möglich

Pseudo-, High-, True-Color



Speicherung der Helligkeits- / Farbwerte eines Pixels?

direkt	"true color", üblich 24 bit 888 RGB
	"high color", z.B. 16 bit 565 RGB
indirekt über LUT	"indexed color" / "pseudo color"
alpha-Kanal	"Transparenz"-Information

Rasterbilder: Beispiel Java

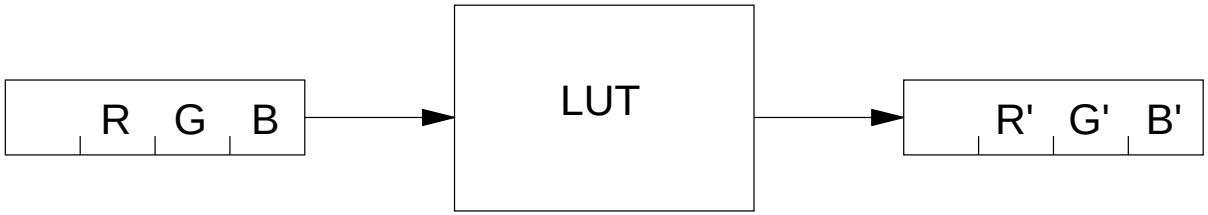
```
import java.awt.image.*;
import javax.imageio.*;

public class ImageDemo {

  public BufferedImage loadImage( InputStream is ) throws Exception {
    return ImageIO.read( is );
  }

  public BufferedImage createRGBImage( int w, int h ) { // 24bit per pixel
    return new BufferedImage( w, h, BufferedImage.TYPE_INT_RGB );
  }

  public BufferedImage redFilter( BufferedImage inputImage ) {
    int w = inputImage.getWidth( null );
    int h = inputImage.getHeight( null );
    BufferedImage BI = createRGBImage( w, h );

    for( int y=0; y < h; y++ ) {                // for all rows y
      for( int x=0; x < w; x++ ) {              // and all columns x
        int pixel = inputImage.getRGB( x, y );  // get pixel value
        BI.setRGB( x, y, pixel&0xFFFF0000 );    // mask red: aaRRggbb
      }
    }
    return BI;
  }

  ...
}
```

Gammakorrektur

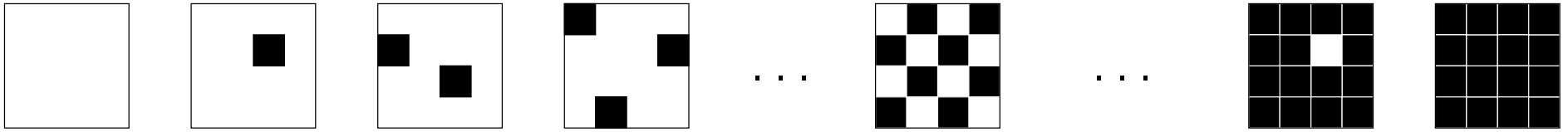
- Farbmonitore / Fernsehen haben nichtlineare Kennlinien
- Helligkeit als Funktion der Eingangsspannung:

$$Y = U^{\gamma} \qquad \gamma = 2.5 \,..\, 3.5$$

- Einsatz einer Lookup-Tabelle für (RGB) -> (R'G'B')
- "Gammakorrektur"



Rasterung und Dithering



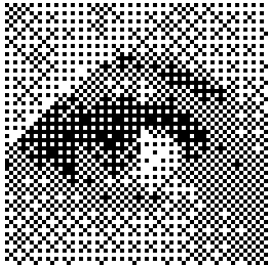
Darstellung von Bildern auf "minderwertigen" Geräten?!

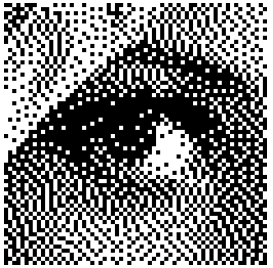
- Reduktion der Farbtiefe — True-Color -> High-Color -> 256 Farben
- Reduktion auf S/W — S/W-Monitor, Laserdrucker, ...
- Vierfarbdruck — Tintenstrahldrucker, Offsetdruck, ...

"Dithering"

- höhere Farbauflösung auf Kosten der räumlichen Auflösung
- regelmässige, periodische Rasterung
- oder "Error-Diffusion", diverse Algorithmen

Dithering: Beispiel Grau nach SW

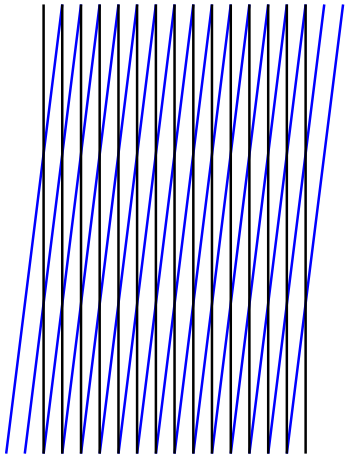


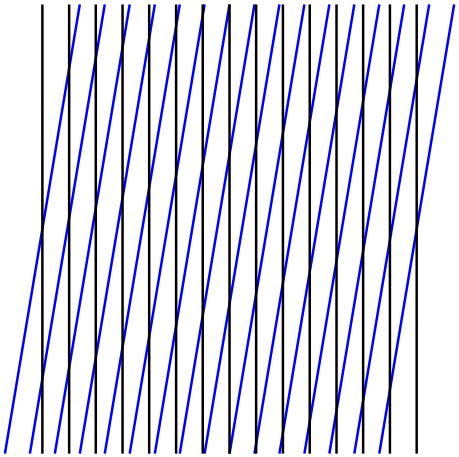


- "nearest neighbor"
- geordnetes, regelmässiges Raster
- Dithering, Floyd-Steinberg Algorithmus
- evtl. schwere Alias-Probleme, s.u.

Dithering: Alias-Effekte

mehrfache Rasterung:





- Überlagerung der einzelnen Raster
- period. Verstärkung/Auslöschung

Alias- bzw. "Moire"-Effekte

- Darstellung eines gerasterten Bildes auf einem Monitor
  (Auflösung des Bildes vs. Auflösung des Monitors)
- Einscannen eines gedruckten Bildes
  (Druckauflösung vs. Auflösung des Scanners)
- usw.

- Verwendung "kompatibler" Raster (ganzzahliges Verhältnis)
- komplizierte Algorithmen zur "Entrasterung"

Netpbm: PBM, PGM, PPM

```
P3
# width height maxval r g b r g b r g b . . .
512 512
255
226 137 125 226 137 125 223 137 133 223 136 128 226 138 120
226 129 116 228 138 123 227 134 124 227 140 127 225 136 119
228 135 126 225 134 121 223 130 108 226 139 119 223 135 120
221 129 114 221 134 108 221 131 113 222 138 121 222 139 114
...
```

"Portable Bitmap / Graymap / Pixmap / Anymap":

- einfache, portable Dateiformate für S/W-, Grau-, Farbbilder
- "raw" oder ASCII-Format
- Format, Größe, max. Farbwert, Pixeldaten
- unkomprimiert, aber leicht zu bearbeiten (z.B. eigene Parser/Writer)
- Jef Poskanzer (1991), jetzt als SourceForge-Projekt

(netpbm.sourceforge.net / www.acme.com/software/pbmplus)

Netpbm: Tools

Dutzende Format-Konverter:

BMP, GIF, JPEG, PNG, TIFF, usw.

Palm Pixmap, Nokia SMF, Windows Icons, WAP bitmap, ...

Beispiele für Transformationen und Filter:

ppmlabel	Text
ppmbrighten, ppmdim, ppmgamma	Helligkeit
ppmnorm	Histogrammausgleich
ppmscale, ppmrotate, ppmshear	Skalierung, Rotation, ...
ppmdither, ppmquant, ...	Dithering, Quantisierung
ppmmix, ppmarith, ...	Überlagerung, Verknüpfung
ppmconvol	allg. Filter
ppmshadow, ppmrainbow, ...	diverse Spezialfunktionen

(netpbm.1 Manpage)

Netpbm: Anwendungsbeispiel

- Verkettung von Filtern / Operatoren via Unix-Pipes
- auch per Skript: Automatisierung / Batch-Verarbeitung
- Beispiel: mehrere Bilder konvertieren:

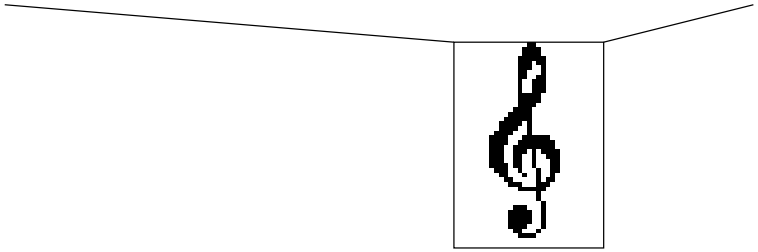
```
for i in *.png; do pngtopnm $i | ppmtojpeg >`basename $i .png`.jpg; done
```

- Beispiel: Scans (TIFF) für dieses Skript aufbereiten:

```
tifftoppm scan.tif |                    von TIFF nach PPM
   ppmbrighten -value +10 |             Bild aufhellen
      pnmscale -width 800 |             neue Bildgrösse
         ppmquant 31 |                  Anzahl der Farben
            ppmtogif > scan.gif         Umwandlung nach GIF
```

XBM-Format

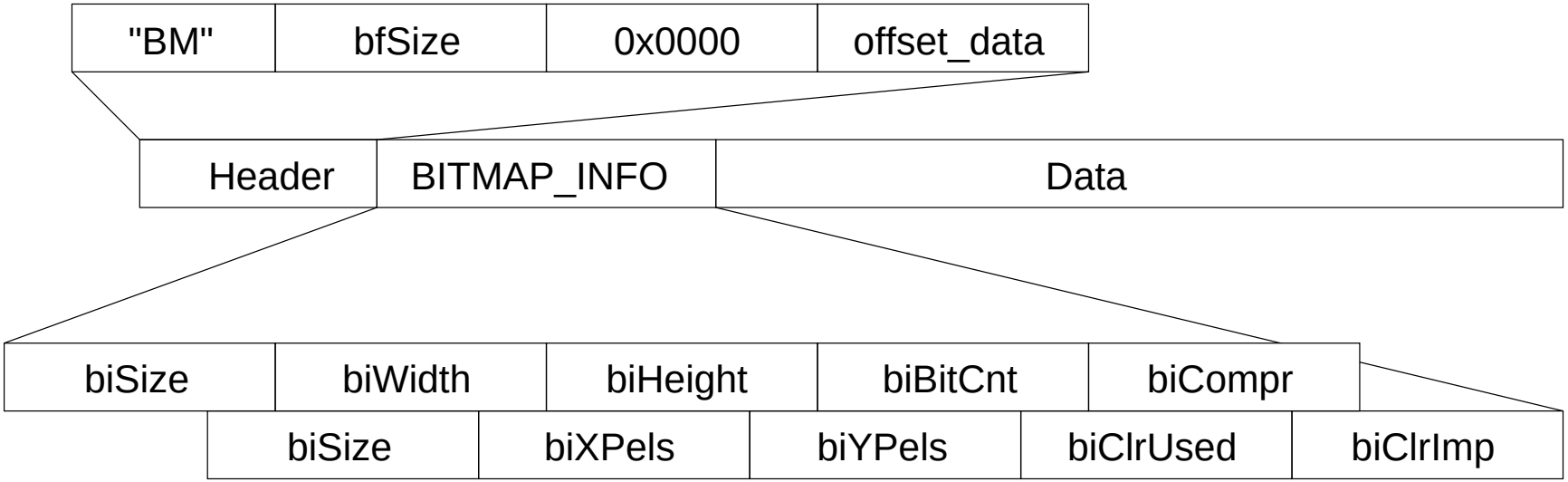
```
#define key0_width 32
#define key0_height 42
static char key0_bits[] = {
 0x00,0x00,0x03,0x00,0x00,0x80,0x07,0x00,0x00,0x80,0x07,0x00,0x00,0xc0,0x0f,
 0x00,0x00,0xc0,0x0d,0x00,0x00,0xc0,0x09,0x00,0x00,0xc0,0x08,0x00,0x00,0xc0,
 0x0c,0x00,0x00,0x40,0x0e,0x00,0x00,0x40,0x0e,0x00,0x00,0x40,0x0e,0x00,0x00,
 0xc0,0x07,0x00,0x00,0xc0,0x07,0x00,0x00,0xc0,0x03,0x00,0x00,0xe0,0x01,0x00,
 0x00,0xf0,0x01,0x00,0x00,0xf8,0x01,0x00,0x00,0x7c,0x01,0x00,0x00,0x3c,0x01,
 0x00,0x00,0x1e,0x01,0x00,0x00,0x8f,0x1f,0x00,0x00,0xcf,0x3f,0x00,0x00,0xe7,
 0x3f,0x00,0x00,0xe7,0x7a,0x00,0x00,0x67,0x72,0x00,0x00,0x67,0x62,0x00,0x00,
 0x6f,0x62,0x00,0x00,0x4e,0x64,0x00,0x00,0x8c,0x24,0x00,0x00,0x18,0x34,0x00,
 0x00,0x70,0x1c,0x00,0x00,0xc0,0x0f,0x00,0x00,0x00,0x04,0x00,0x00,0x00,0x04,
 0x00,0x00,0x00,0x08,0x00,0x00,0xe0,0x08,0x00,0x00,0xf0,0x09,0x00,0x00,0xf0,
 0x09,0x00,0x00,0xf0,0x09,0x00,0x00,0xf0,0x08,0x00,0x00,0x60,0x04,0x00,0x00,
 0xc0,0x03,0x00};
```



- einfaches Format für S/W-Bilder
- als C-Quelltext, unkomprimiert

- ähnliches Prinzip für XPM (X11 Pixmap)

BMP: Bitmap-Format

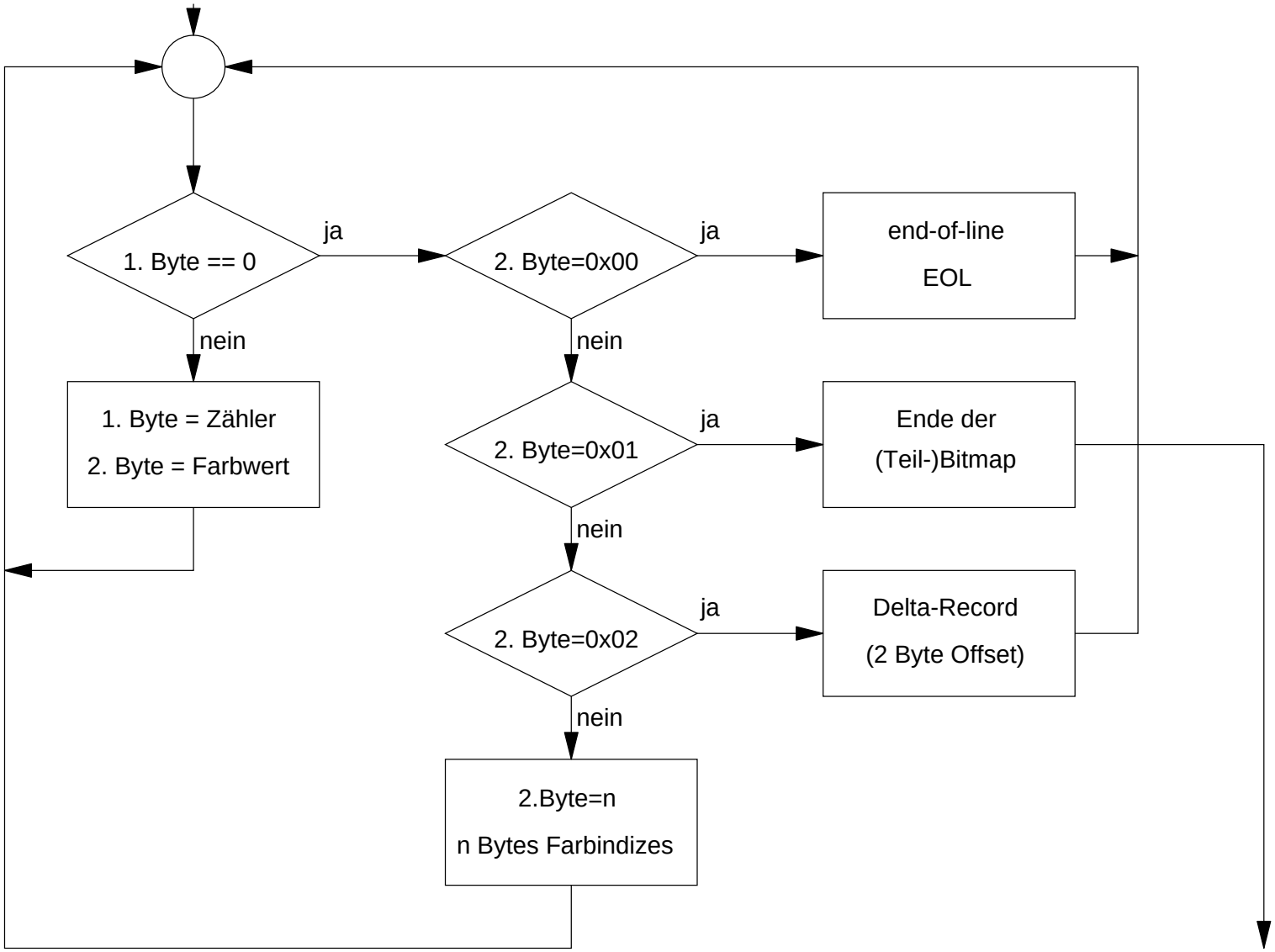


ab Offset 0x36 die Daten:

...	R	G	B	-	R	G	B	-	...

- unkomprimiert als RGB_QUAD:
- unkomprimiert mit "Maske", z.B. für 16-bit 5-6-5 Kodierung
- Lauflängenkodiert

BMP: Runlength Encoding



GIF

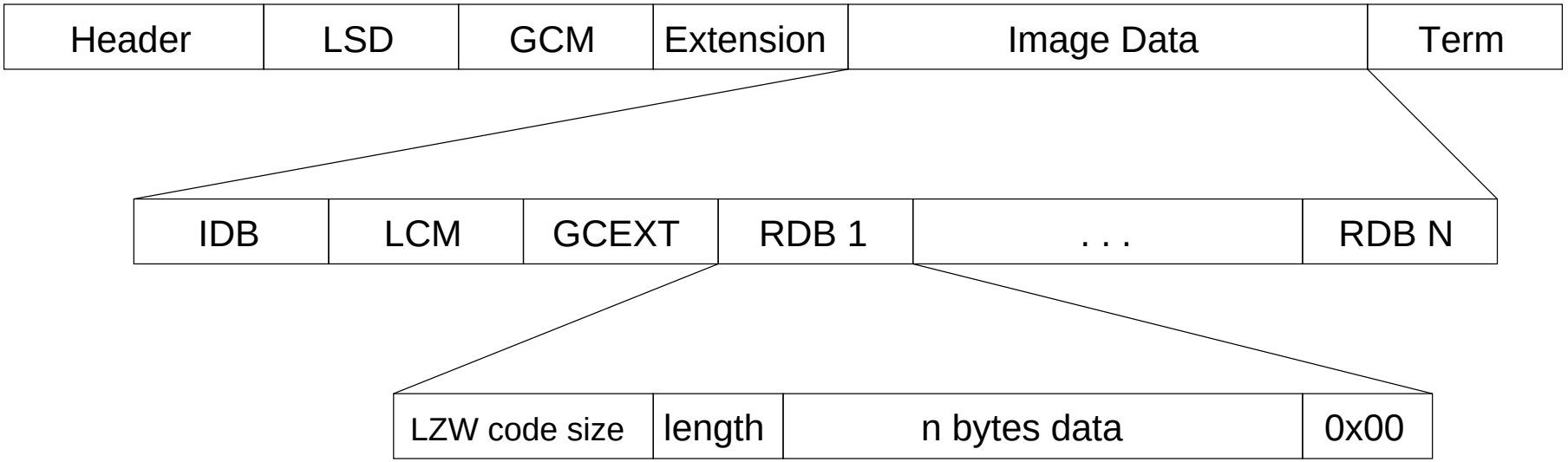
- flexibles Bildformat für Grau- und Farbbilder
- nutzt LZW-Kompression
- weite Verbreitung dank Einsatz in HTML

- Unisys / Compuserve, 1987
- Verfahren ist patentiert, Einsatz im Web lizenzfrei
- aber Lizenz erforderlich für Programme, die GIFs erzeugen
  (Patent läuft in Deutschland im Juni 2004 aus)

- Index-Farbmodell, max. 256 Farben
- max. 16.000 x 16.000 Pixel

  erweiterte Version (GIF 89a):
- transparente Hintergrundfarbe
- "animated GIFs": mehrere Teilbilder in einer Datei

GIF 89a: Dateiformat



LSD	logical screen descriptor: width, height, background, ...
GCM, LCM	global/local color map, 256 entries
EXT	extension block (user defined data)
IDB	image descriptor block
RDB	raster data block: LZW-encoded image data
GCEXT (GIF89a)	graphics control extension, u.a. animation delay
CEXT (GIF89a)	commect extension block

(members.aol.com/royalef/gif89a.txt)

GIF 89a: Interlacing

The rows of an Interlaced images are arranged in the following order:

Group 1 : Every 8th. row, starting with row 0. (Pass 1)
Group 2 : Every 8th. row, starting with row 4. (Pass 2)
Group 3 : Every 4th. row, starting with row 2. (Pass 3)
Group 4 : Every 2nd. row, starting with row 1. (Pass 4)

Row Number	Interlace Pass
0	1
1	4
2	3
3	4
4	2
5	4
6	3
7	4
8	1
9	4
10	3
11	4
12	2
13	4
14	3
15	4
16	1
17	4
18	3
19	4

(members.aol.com/royalef/gif89a.txt)

GIF: LZW-Kompression

- zeilenweise Kodierung der Bilddaten
- nur ein-dimensionale Korrelationen
- gut geeignet für Zeichnungen etc.

LZW-Variante:

- Dictionary wird während der Kodierung gefüllt
- Startgröße ist b=2 (S/W) bzw. b=8 (Grau-/Farbbilder)
- max. 4096 Einträge
- bei Bedarf Anlegen eines neuen Dictonary

  1 byte block size
  data (pointers into dictionary into color map)
  1 byte end marker (0x00)

GIF: Farbphotos

GIF, 256 Farben

GIF

JPEG

GIF-Format nur bedingt für Farbphotos geeignet:

- Beschränkung auf 256 Farben
- typische "körnige" Struktur durch das "Farbrauschen"
- selbst bei Einsatz guter Dithering-Verfahren

- andere Verfahren (PNG, JPEG, TIFF) besser

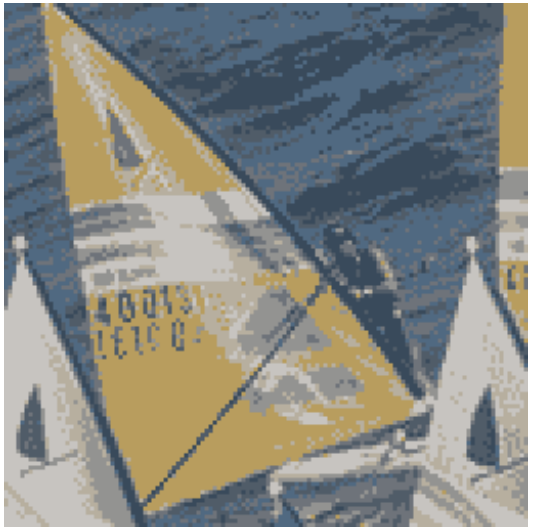
GIF: Dithering

GIF; 256 Farben

32 Farben, Floyd-Steinberg

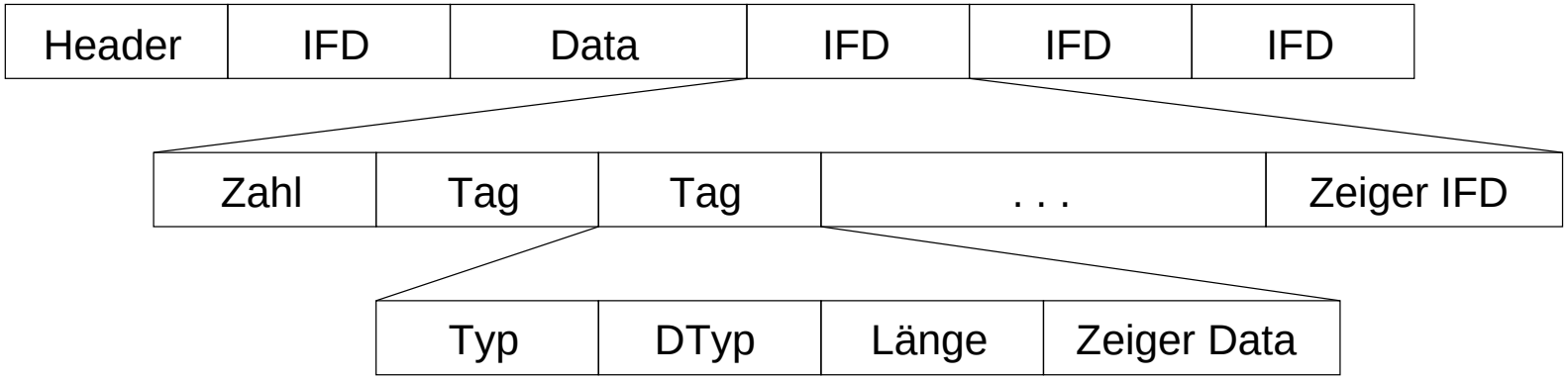
32 Farben, nearest

8 Farben, Floyd-Steinberg



8 Farben, nearest

TIFF



"Tagged Image File Format"

- Aldus Corporation, Hewlett Packard, Microsoft, 1980
- hierarchischer Aufbau, Header und var. Anzahl von Datenblöcken
- Verkettung der Blöcke über Zeiger (Image File Directories)
- Zeiger kodiert in 12-Byte "Tag"-Strukturen

- Vielzahl von Tag-Varianten definiert
- für Bildformat, -auflösung, -größe, -kompression, ...

PNG / MNG:

"Portable Network Graphics" / "Multiple Network Graphics" Format

- entwickelt 1995 als lizenzfreier GIF-Ersatz
- http://www.libpng.org

- Grau- oder Farbbilder, bis zu 48 bit Farbtiefe
- bessere Datenkompression als GIF (deflate-Algorithmus)
- alternativ auch verlustfreie Speicherung
- verlustfreie Konvertierung PNG <-> GIF
- Interlacing und Animation ähnlich wie bei GIF
- aber vollwertige 8-bit Transparenz

- für Fotos schlechter als JPEG (s.u.)
- immer noch nicht vollständig im Internet Explorer implementiert

PNG: Filter

- Vorverarbeitung zum Erleichtern der Kompression
- hier: Vorhersage von Pixelwerten aus Nachbarpixeln
- PNG definiert fünf Filteroperationen
- Anwendung jeweils separat für alle Farbkanäle

None: `none[x,y] = img[x,y]`

Sub: `sub [x,y] = img[x,y] - img[x-1,y]`

Up: `up [x,y] = img[x,y] - img[x,y-1]`

Avg: `avg [x,y] = img[x,y] -(img[x-1,y]+img[x,y-1])/2`

Paeth:

```
a = img[x-1,y] b= img[x,y-1] c = img[x-1,y-1]
paeth[x,y] = img[x,y] + praediktor(a,b,c)
praediktor(a,b,c):
    e  = a + b - c;  // inital estimate
    da = abs(e-a); db = abs(e-b); dc = abs(e-c);
    if ((da <= db) && (da <= dc)) return a;
    else if (db <= dc) return b;
    else return c;
```

PNG: Alpha-Transparenz

(Schatten beachten!)

- Demos z.B. unter http://www.libpng.org/pub/png/pngs-img.html
- z.B. für 'randlose' Einbettung von Fotos vor Hintergründen
- unterstützt von Mozilla, Opera, ... - nicht vom MSIE

Bildverarbeitung: Filter

- Grundoperatoren: lineare, verschiebungsinvariante Filter
- Beispiel 3x3 Filter zur Glättung:

$$\begin{array}{ccccc} & \ldots & \ldots & \ldots & \ldots \\ \ldots & 0 & 0 & 1 & 1 \quad \ldots \\ \ldots & 0 & 0 & 1 & 1 \quad \ldots \\ \ldots & 0 & 0 & 1 & 1 \quad \ldots \\ & \ldots & \ldots & \ldots & \ldots \end{array} \; * \; 1/9 \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} = \begin{array}{ccccc} & \ldots & \ldots & \ldots & \ldots \\ \ldots & 0 & 1/3 & 2/3 & 1 \quad \ldots \\ \ldots & 0 & 1/3 & 2/3 & 1 \quad \ldots \\ \ldots & 0 & 1/3 & 2/3 & 1 \quad \ldots \\ & \ldots & \ldots & \ldots & \ldots \end{array}$$

- für jeden Bildpunkt: Auslesen der Nachbarpixel
- Multiplikation mit den entsprechenden Filterkoeffizienten
- Summation

- alternative Berechnung via Fouriertransformation

Filter:

- große Auswahl wichtiger Algorithmen
- typ. Matrixgröße von 3x3 bis 15x15 bis ..
- Limit wegen Rechenaufwand: 9 .. 225 .. Operationen / Pixel

  Glättung, Weichzeichner (Gauß), Schärfung,
  Kantendetektion, ...

- Verbesserung durch "separierbare" Filter
- Auftrennung der Matrix in zwei Vektoren
- Anwendung erst zeilen- dann spaltenweise (oder umgekehrt)
- lineare Abhängigkeit von der Filtergröße

- entsprechendes Prinzip auch für nichtlineare Op. (z.B. Median)

Filter:

Beispielmatrizen: Weichzeichner 3x3 und 5x5:

Koeffizienten aus Binomialverteilung

$$\frac{1}{16}\begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix} \qquad \frac{1}{256}\begin{pmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{pmatrix}$$

- Herleitung und weitere Beispiele: s. Jähne

Bildverarbeitung: Filter-Pipelines

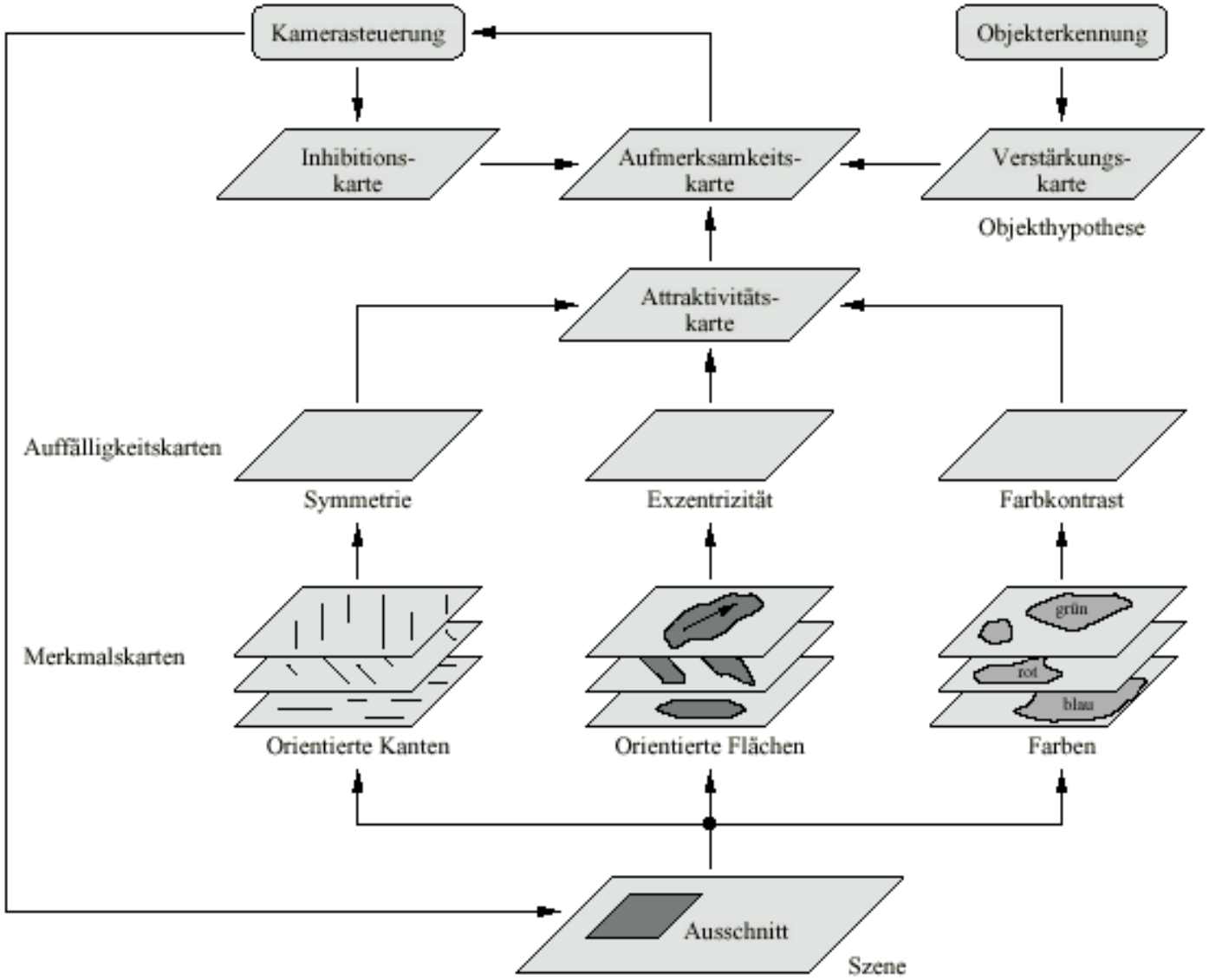


Abbildung 4.1: Schema der Aufmerksamkeitssteuerung in NAVIS

"high level" Verarbeitung

weitere Filter zur Kombination mehrere Merkmale

diverse Filterbänke, Merkmalsextraktion

Eingabedaten

(Bollmann 2000)

Vektorgraphik

- Aufwand bei Rasterbildern: (etwa) quadratisch mit Bildgrösse
- auflösungsunabhängige Formate?

=> "Vektorgraphik"

- Grundobjekte Linien, Rechtecke, usw.
- Angabe der Koordinaten und Attribute
- einfache Manipulation
- Darstellung via Koordinatentransformation und Rasterung

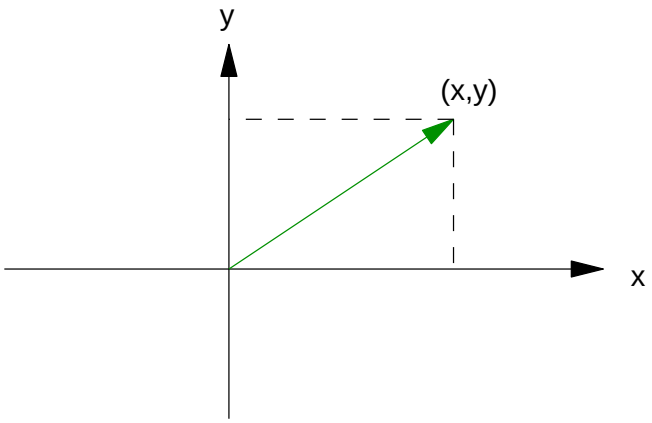
- ideal für Zeichnungen, Diagramme, etc.
- aber nur bedingt für "photorealistische" Darstellungen
- diverse Editoren: Corel Draw & Co.

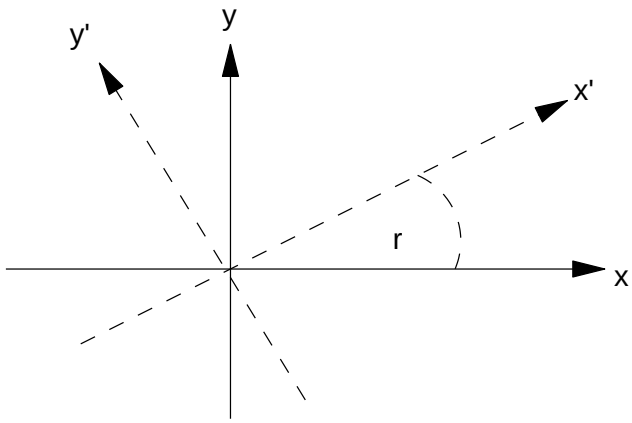
Vektorgraphik: Beispiel FIG-Format

```
#FIG 3.2                    Dateiheader, globale Attribute
Landscape
Center
...

0 34 #4193ff                0=benutzerdefinierte Farbe (RGB)

4 0 0 100 ... 2250 8100 Vektorgraphik
...                         4=Textobjekt,0=left justified,0=black...

2 2 0 1 7 0 ... 5
   0 0 13500 0 13500 0 9450 0 9450 0 0
...                         2=Polyline,2=Box,0=solid,1=thickness...
                            ... 5=number of points
                            ... 5 points (x1,y1) (x2,y2) ...
3 1 0 ... 45
...                         3=Spline,1=closed,0=solid... 45=npoints.
```

(www.xfig.org, ftp.x.org/contrib/applications/drawing_tools/xfig)

Vektorgraphik:





einfache Transformationen: Skalierung, Translation, Rotation

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} ax \\ ay \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} x0 \\ y0 \end{pmatrix}$$

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos r & \sin r \\ -\sin r & \cos r \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

Rasterung via "Bresenham"-Algorithmus

Fonts: TrueType

- entfällt aus Zeitgründen, siehe Henning 3.2