

# Interaktive Medien



Human-Computer Interaction  
Fachbereich Informatik  
Universität Hamburg



# Interaktive Medien

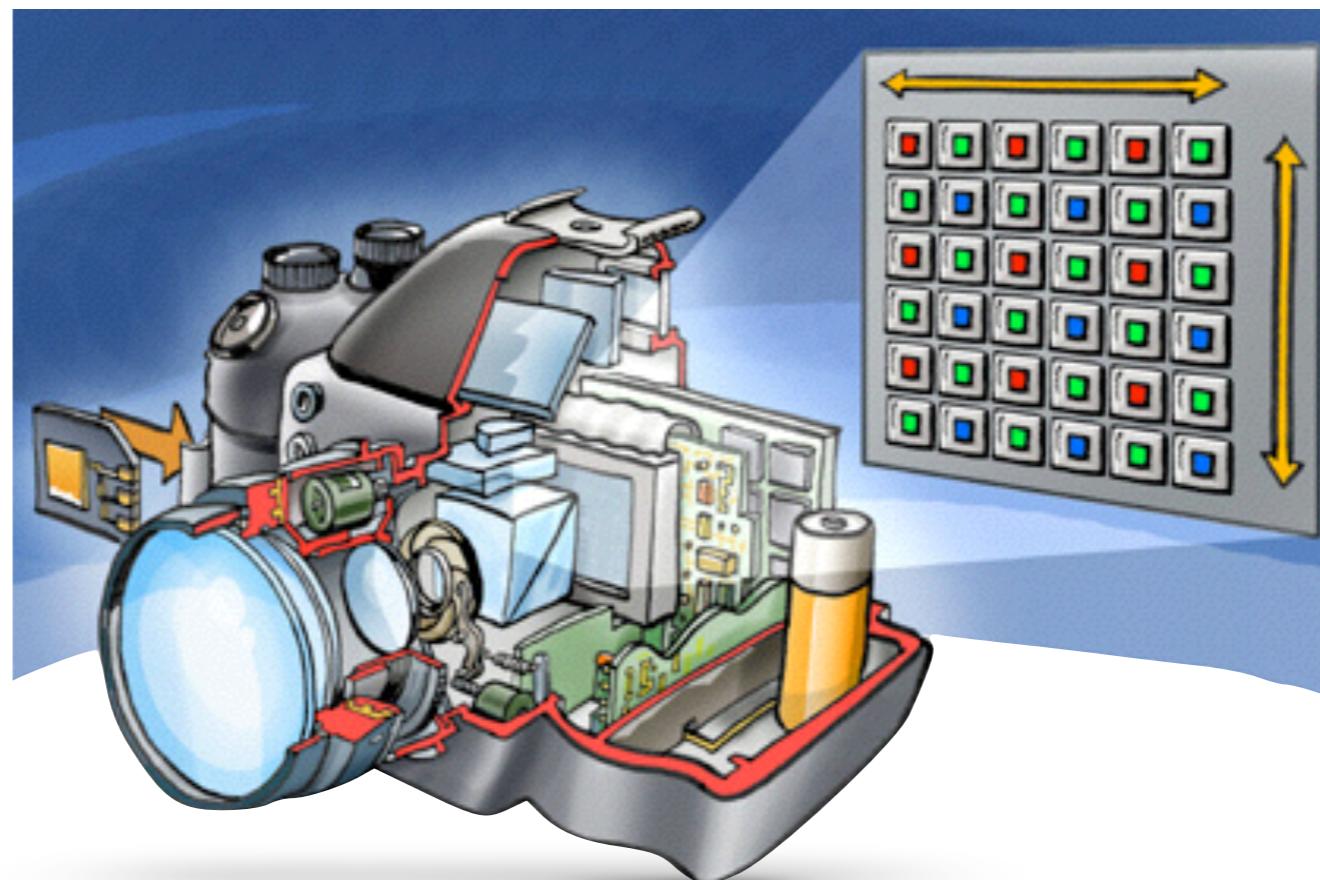
## Kapitel Bilder

**Susanne Schmidt**

Human-Computer Interaction, Universität Hamburg

# Digitalbilder

- **Digitalkamera** nutzt anstatt eines analogen Films einen elektronischen Bildwandler (**Bildsensor**) und digitales Speichermedium



# Inhalt

- Licht und Farben
- Farbwahrnehmung
- Digitale Bilder
- Farbmodelle (RGB, CMY(K), HSV, CIE XYZ)
- Kompression von Rastergrafiken (JPEG, GIF, PNG)
- *Praxis: Bildverarbeitung*



# Interaktive Medien

## Kapitel Bilder

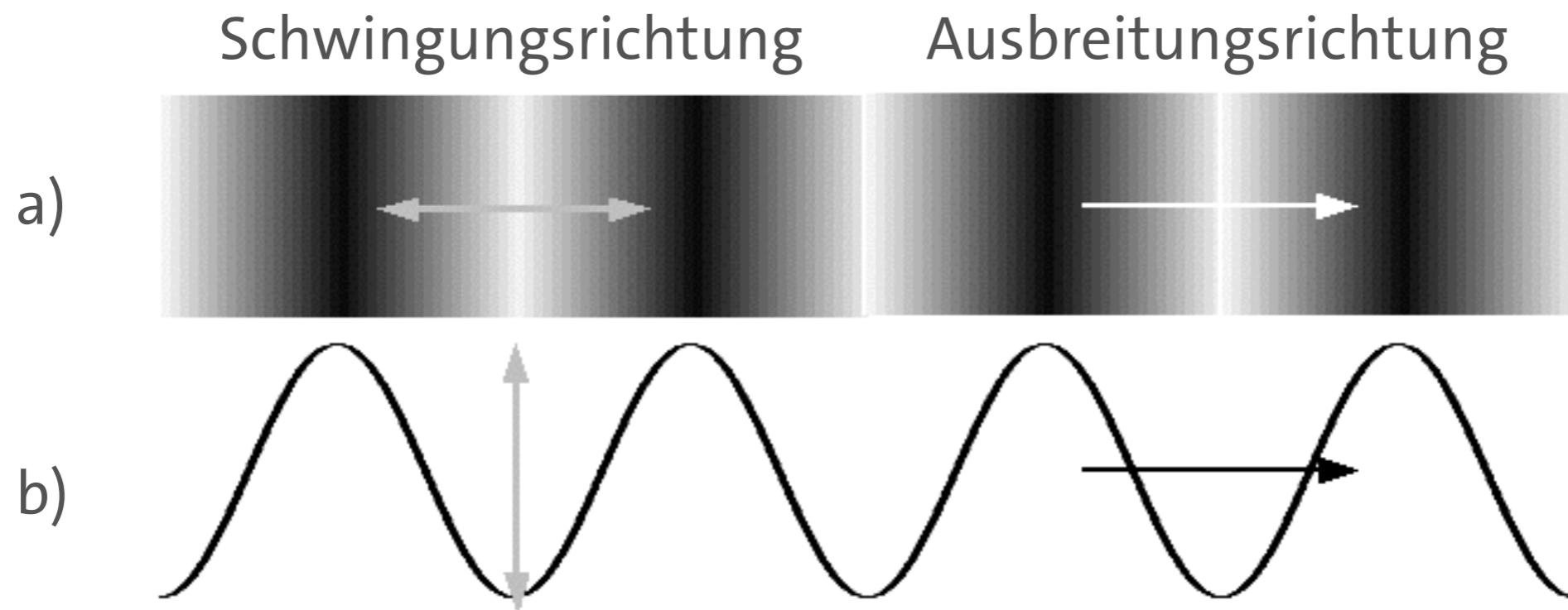
### Licht und Farben

# Welle oder Teilchen?

- Licht ist sichtbarer Bereich des **elektromagnetischen Spektrums**
  - **Wellenoptik:** Welle mit Frequenz, Amplitude und Länge
  - **Quantenphysik:** Menge von einzelnen diskreten Energiequanten (**Photonen**)

# Wellen

## Longitudinal- und Transversal

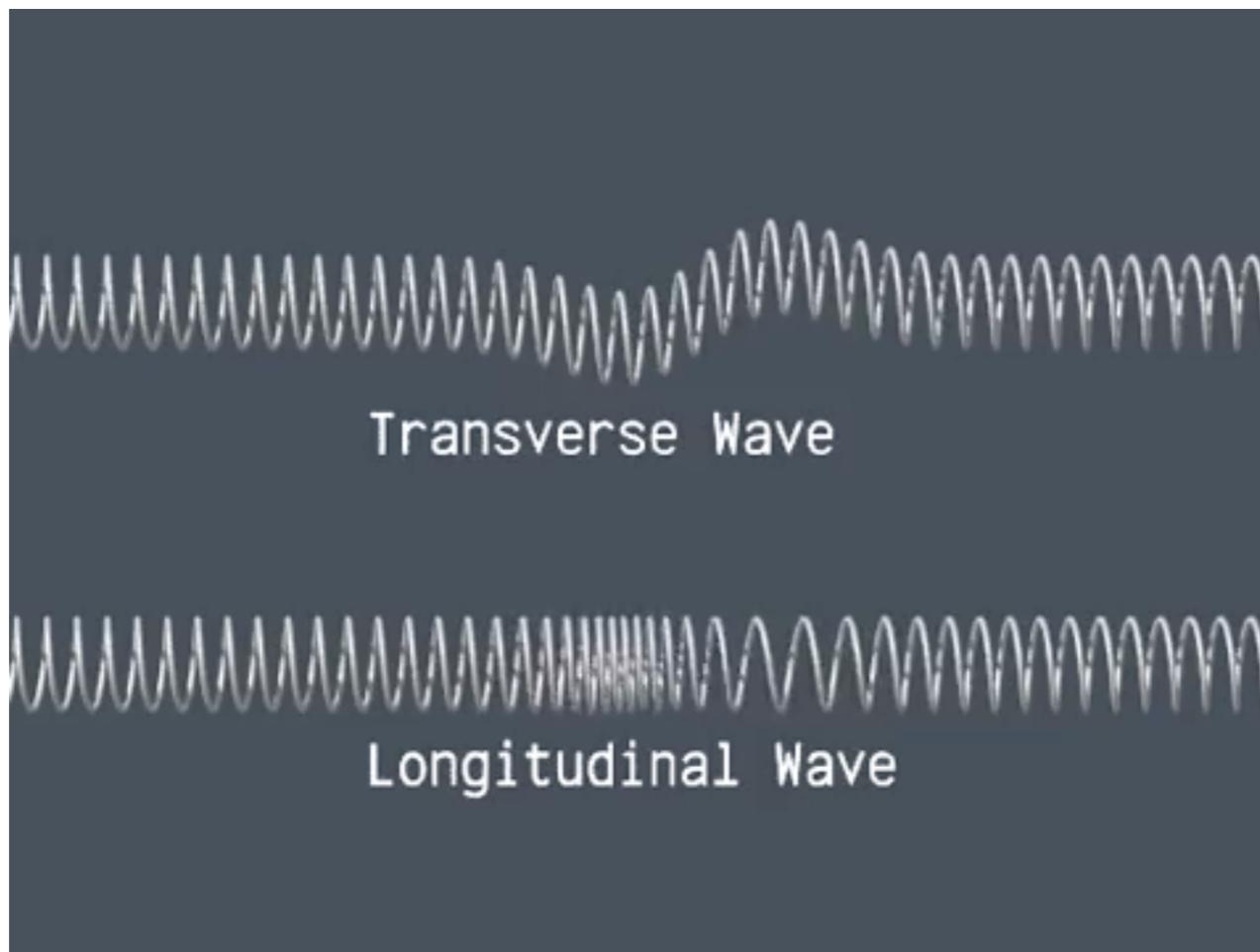


a) **Longitudinalwelle:** Oszillatoren schwingen  
in der Ausbreitungsrichtung

b) **Transversalwelle:** Oszillatoren schwingen  
quer zur Ausbreitungsrichtung

# Wellen

## Longitudinal- und Transversal

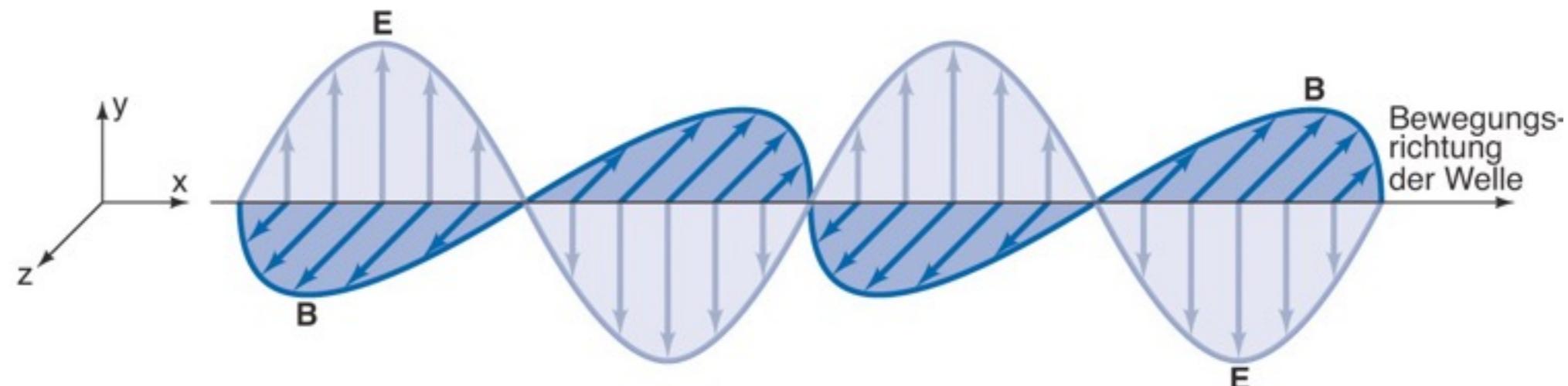


z.B. Licht

z.B. Schall

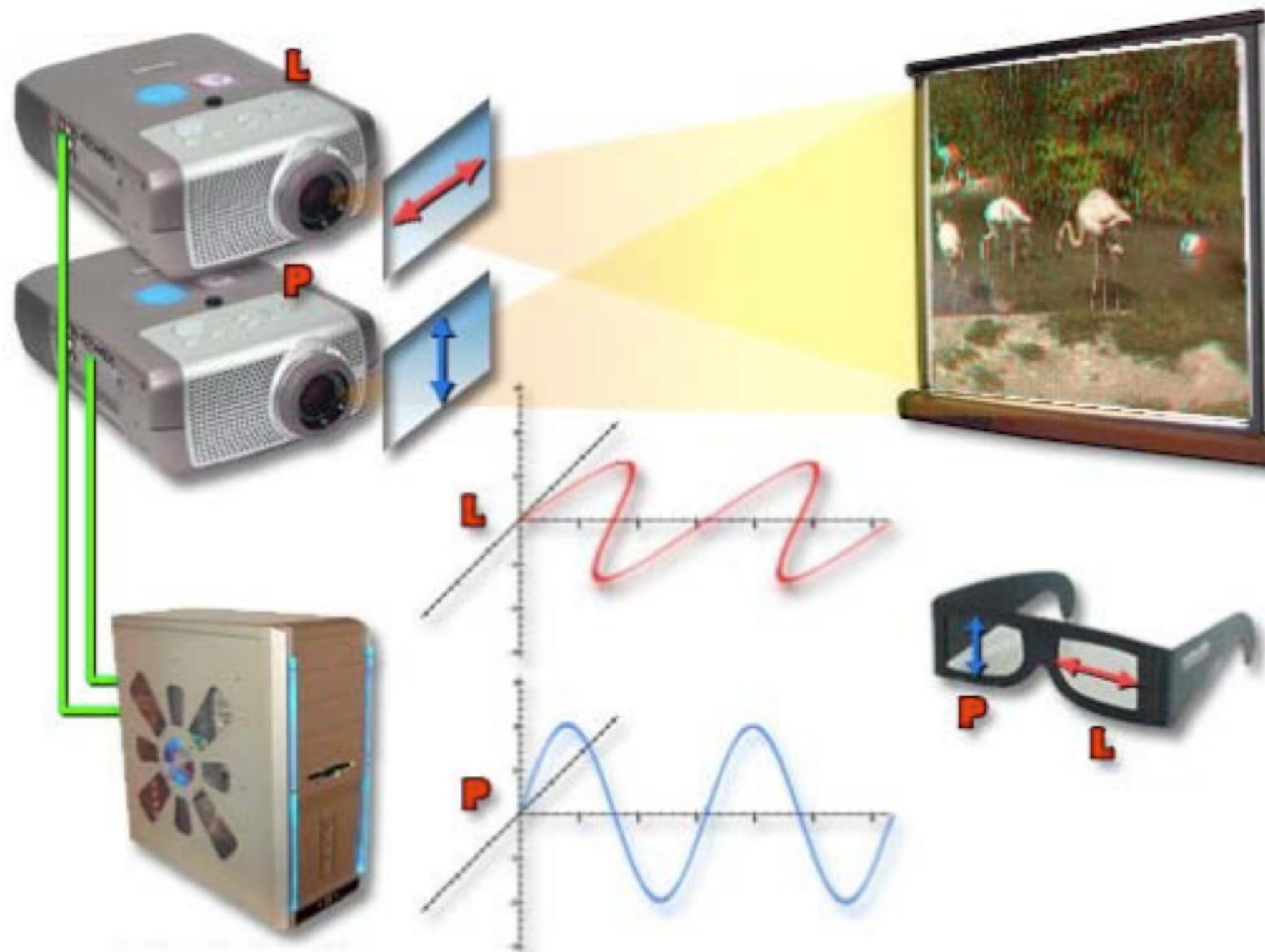
# Polarisation

- Wellen E und B haben zwar gleiche **Amplitude** und **Wellenlänge**, sind aber unterschiedlich gedreht, d.h. **polarisiert**
  - wichtig z.B. für stereoskopische Verfahren



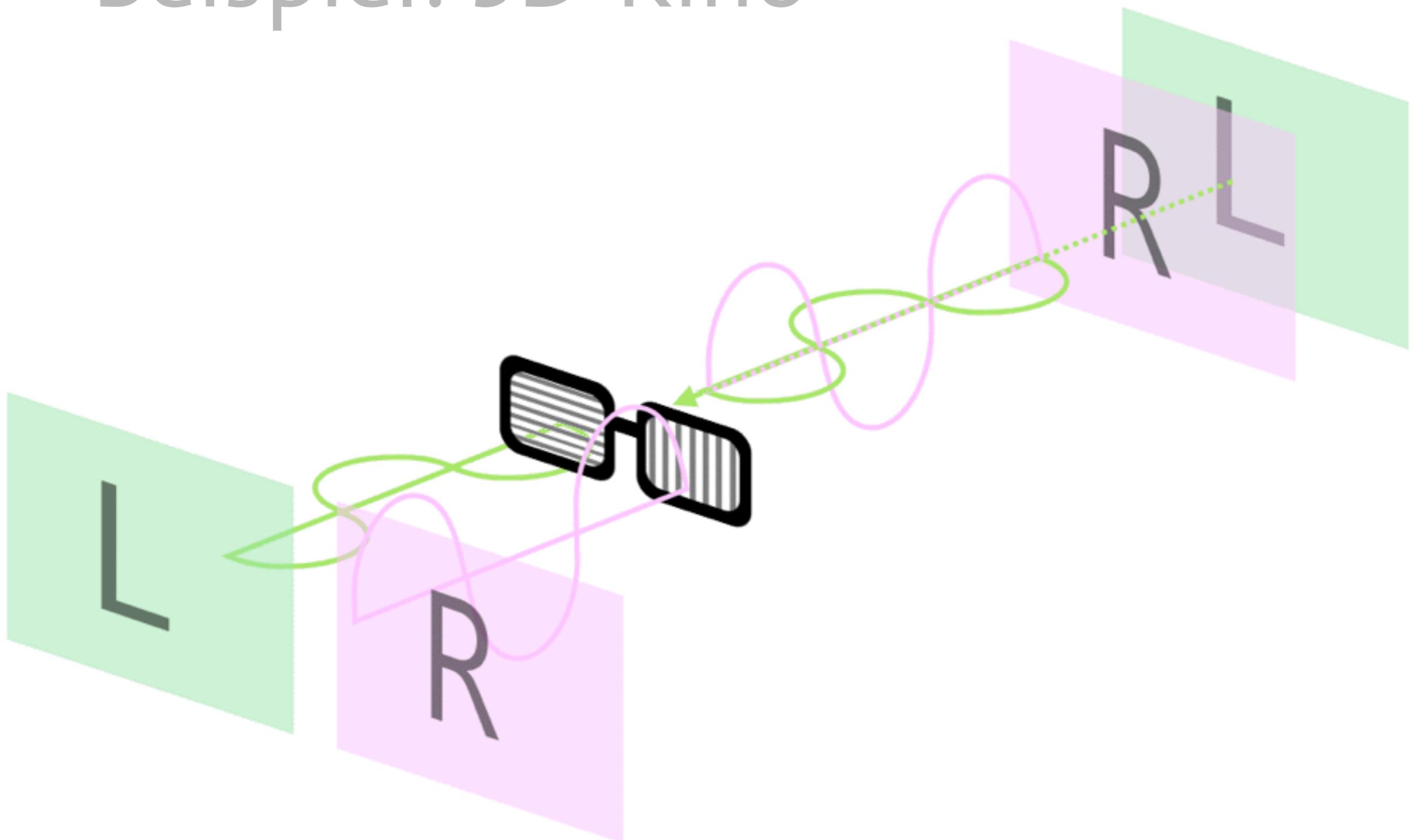
# Polarisation

## Beispiel: 3D-Kino



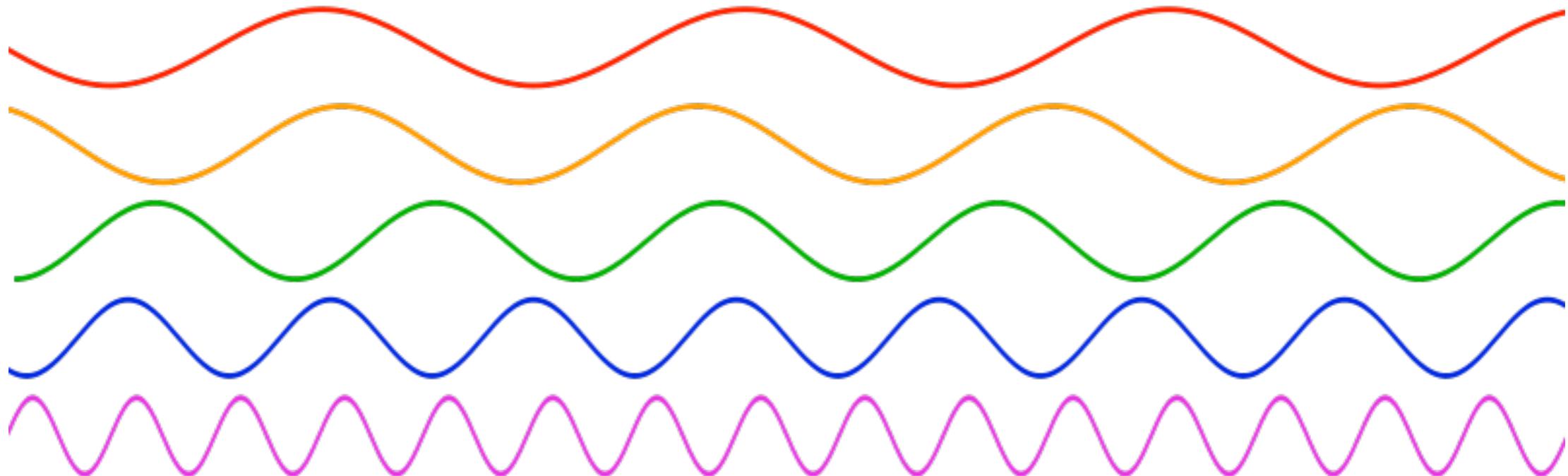
# Polarisation

## Beispiel: 3D-Kino



# Licht als Welle

- Frequenz definiert Farbe
- Amplitude definiert Helligkeit





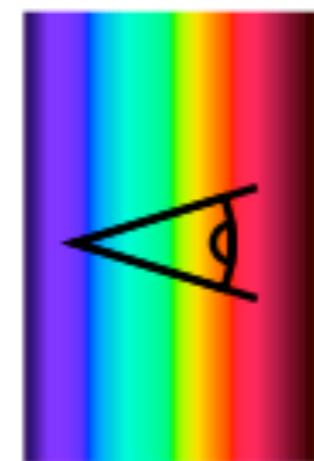
0.01nm



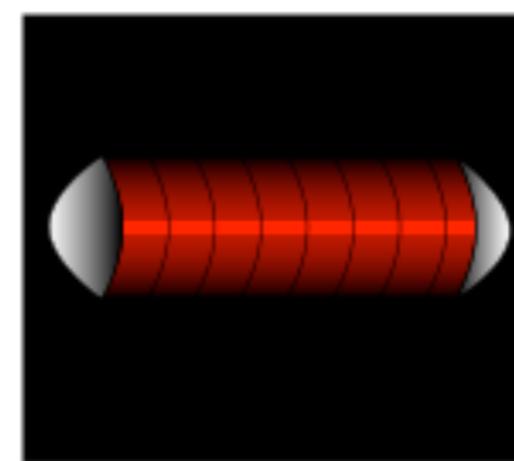
1nm



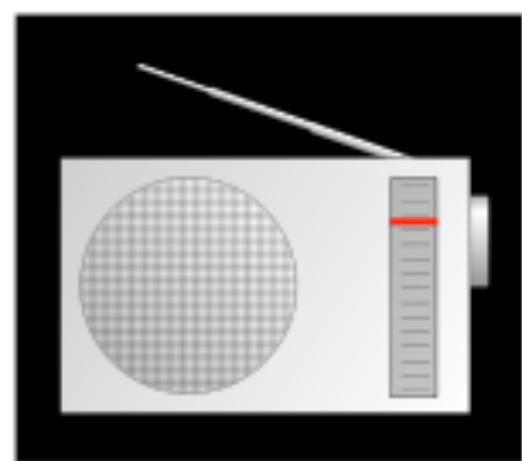
100nm



1mm



1cm

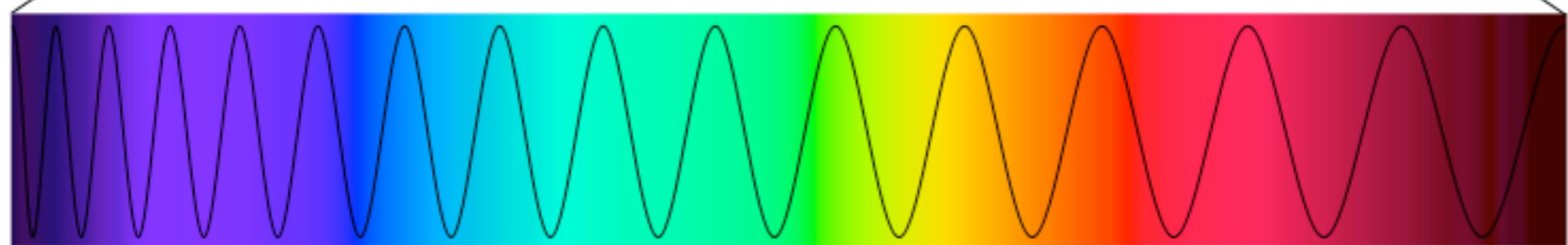


1m

1km

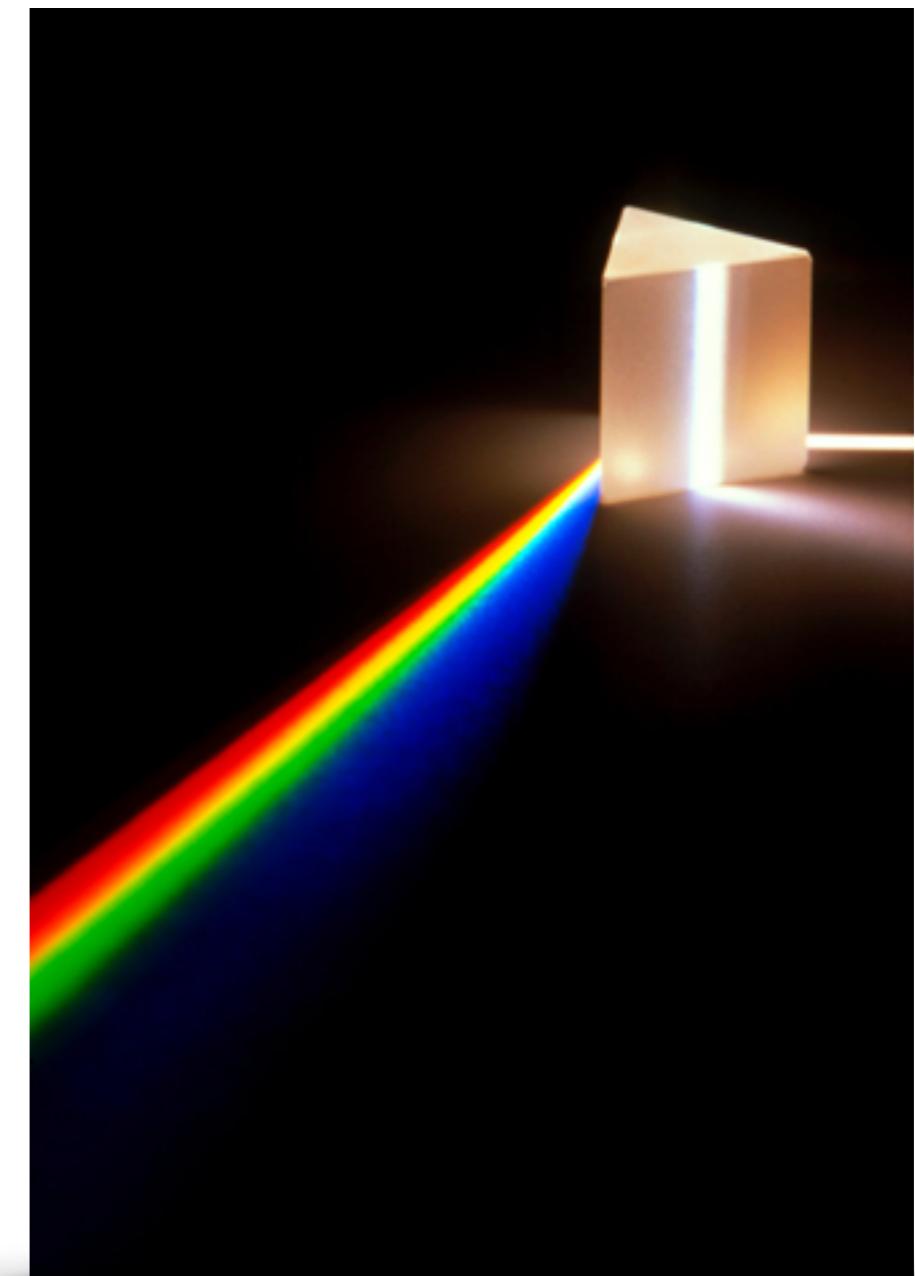
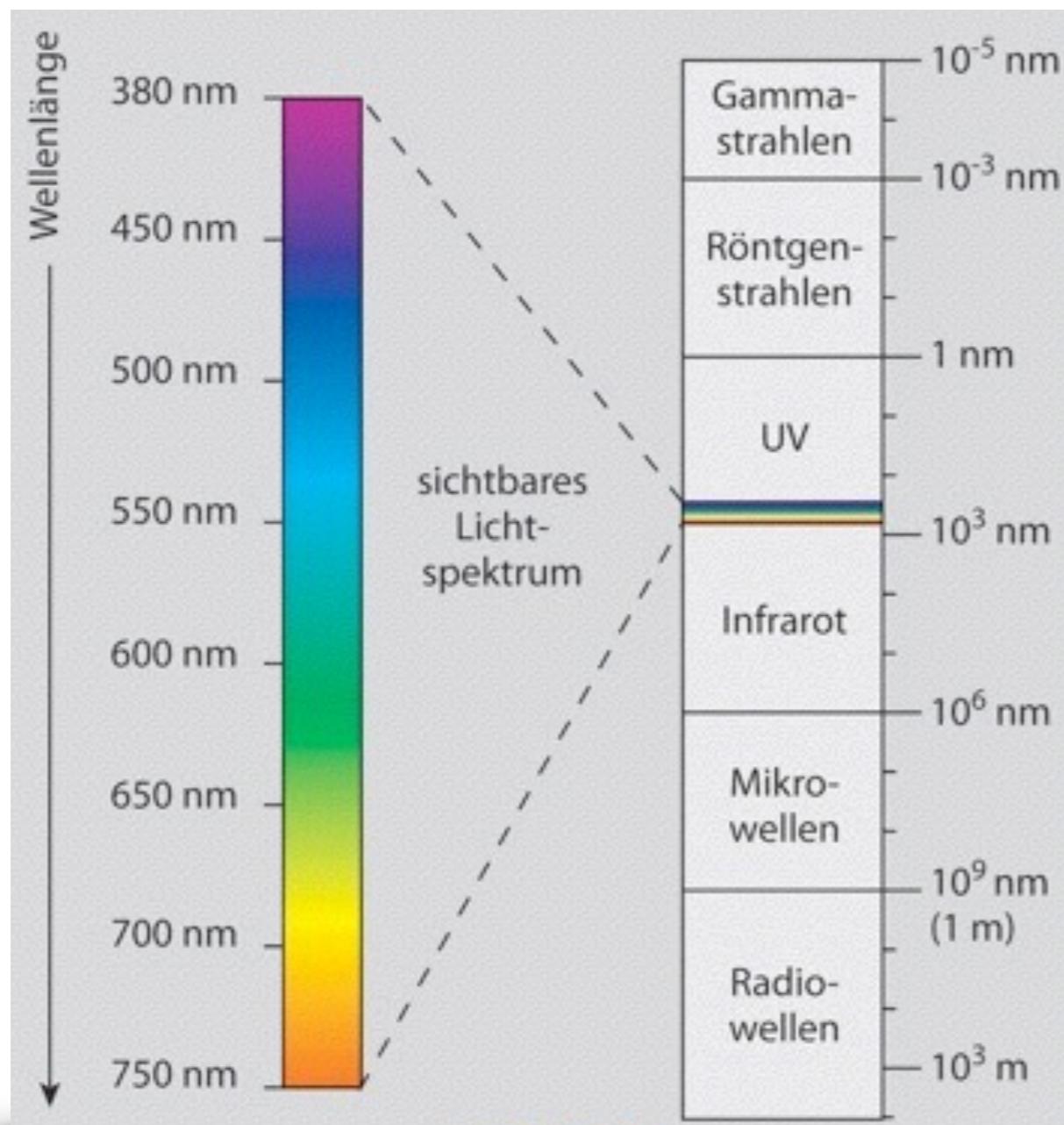
400nm

700nm



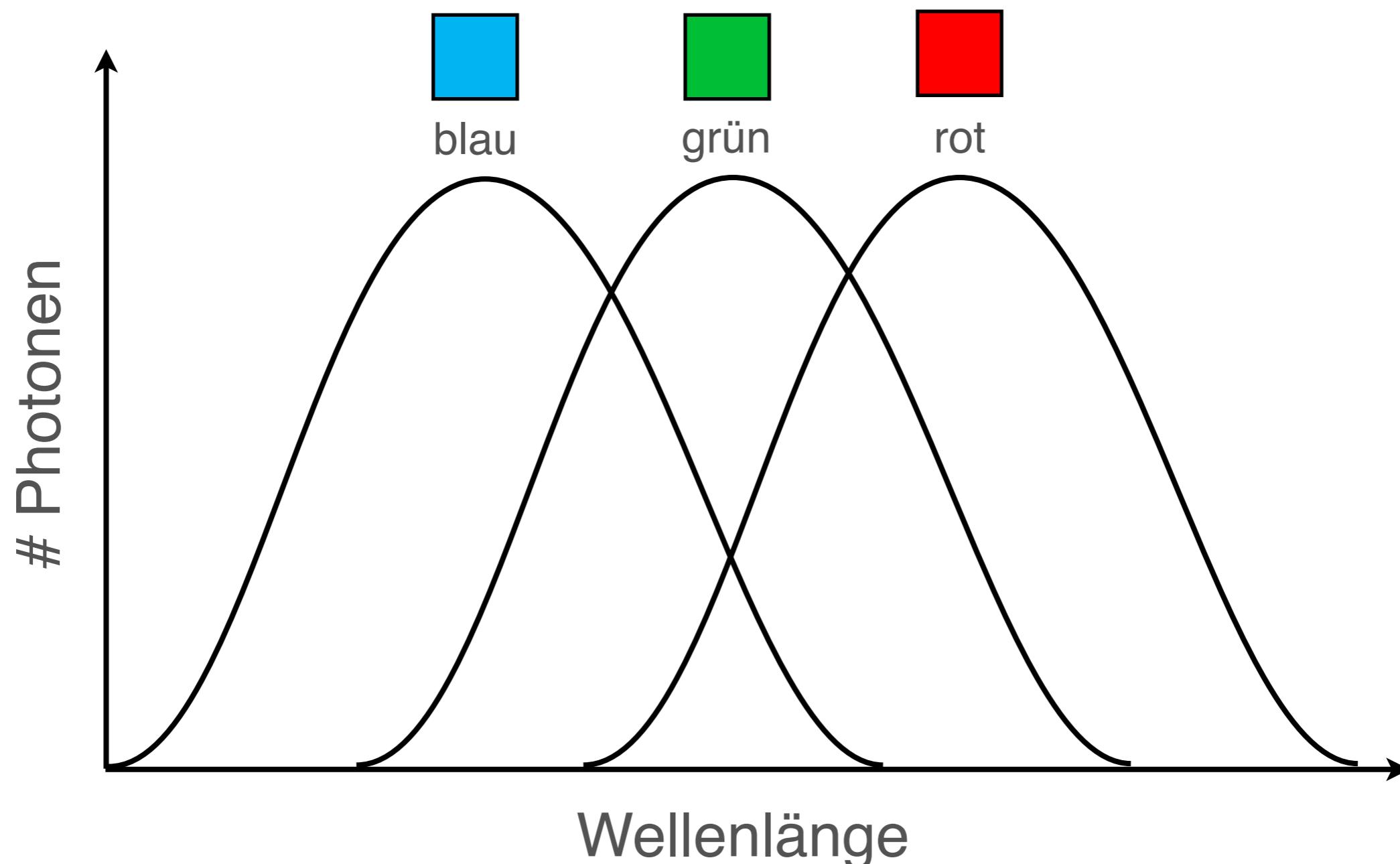
# Lichtspektrum

## Bsp: Sichtbare Anteil



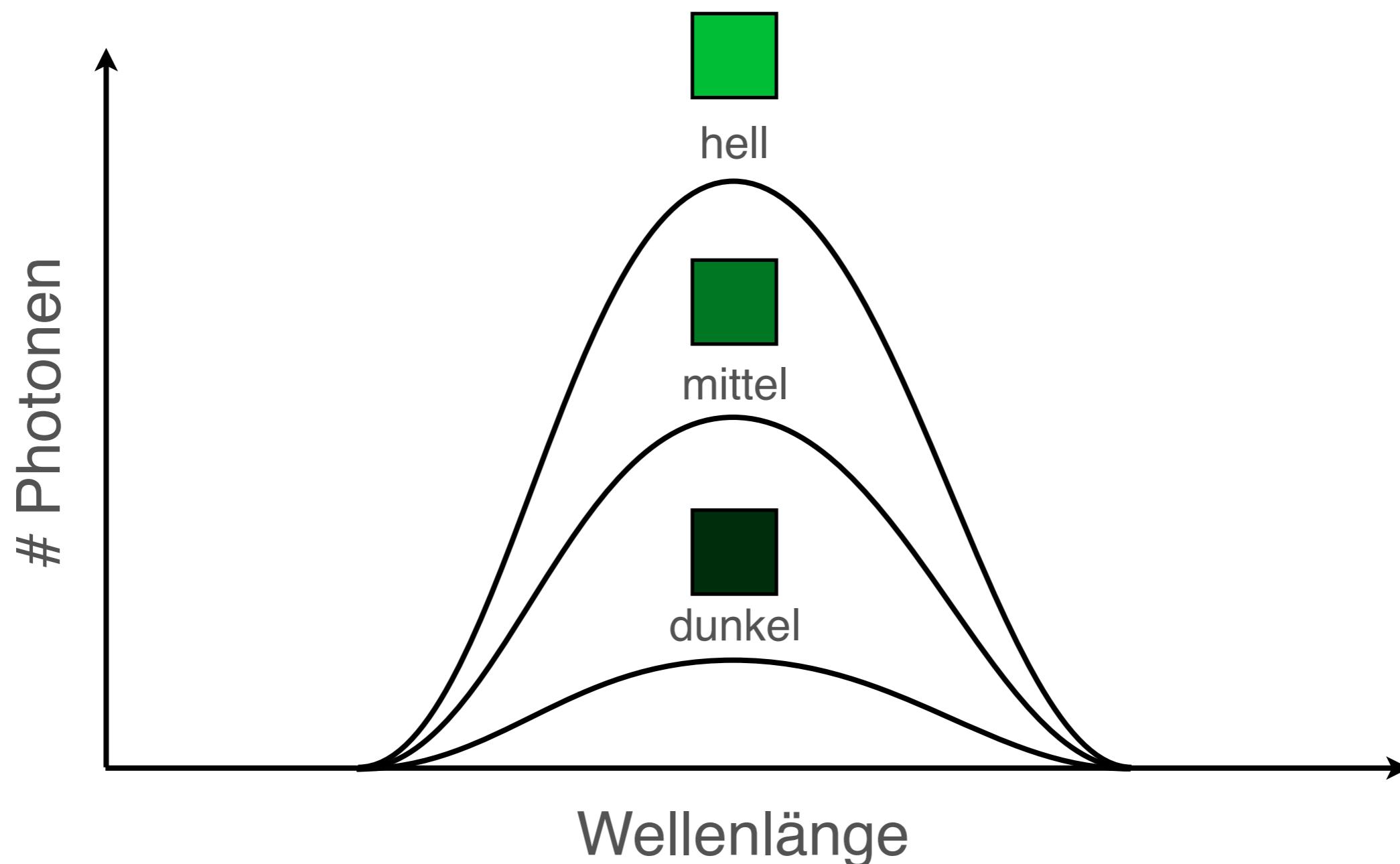
# Visuelle Psychophysik

## Wahrnehmung: Farbton



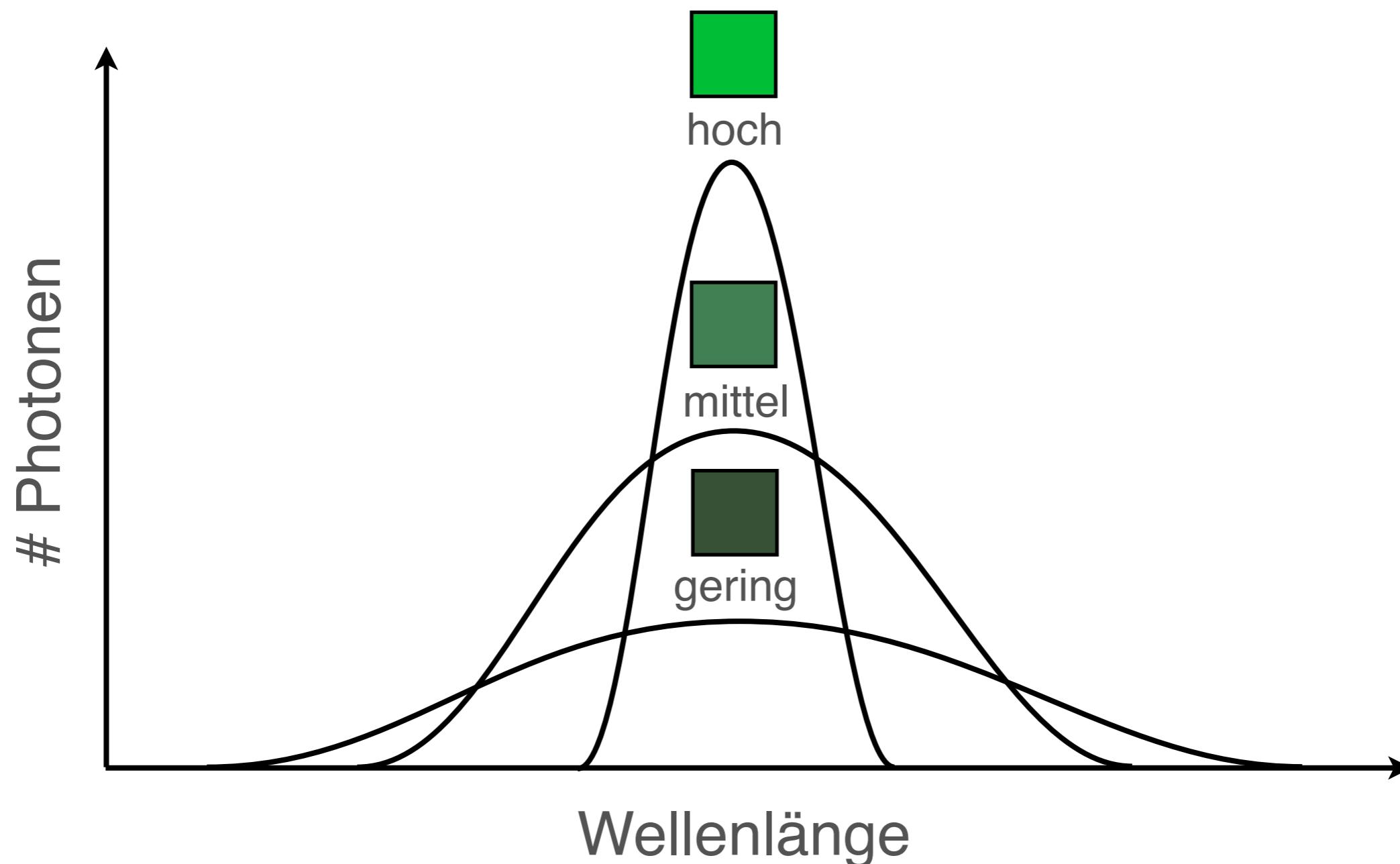
# Visuelle Psychophysik

## Wahrnehmung: Helligkeit



# Visuelle Psychophysik

## Wahrnehmung: Sättigung

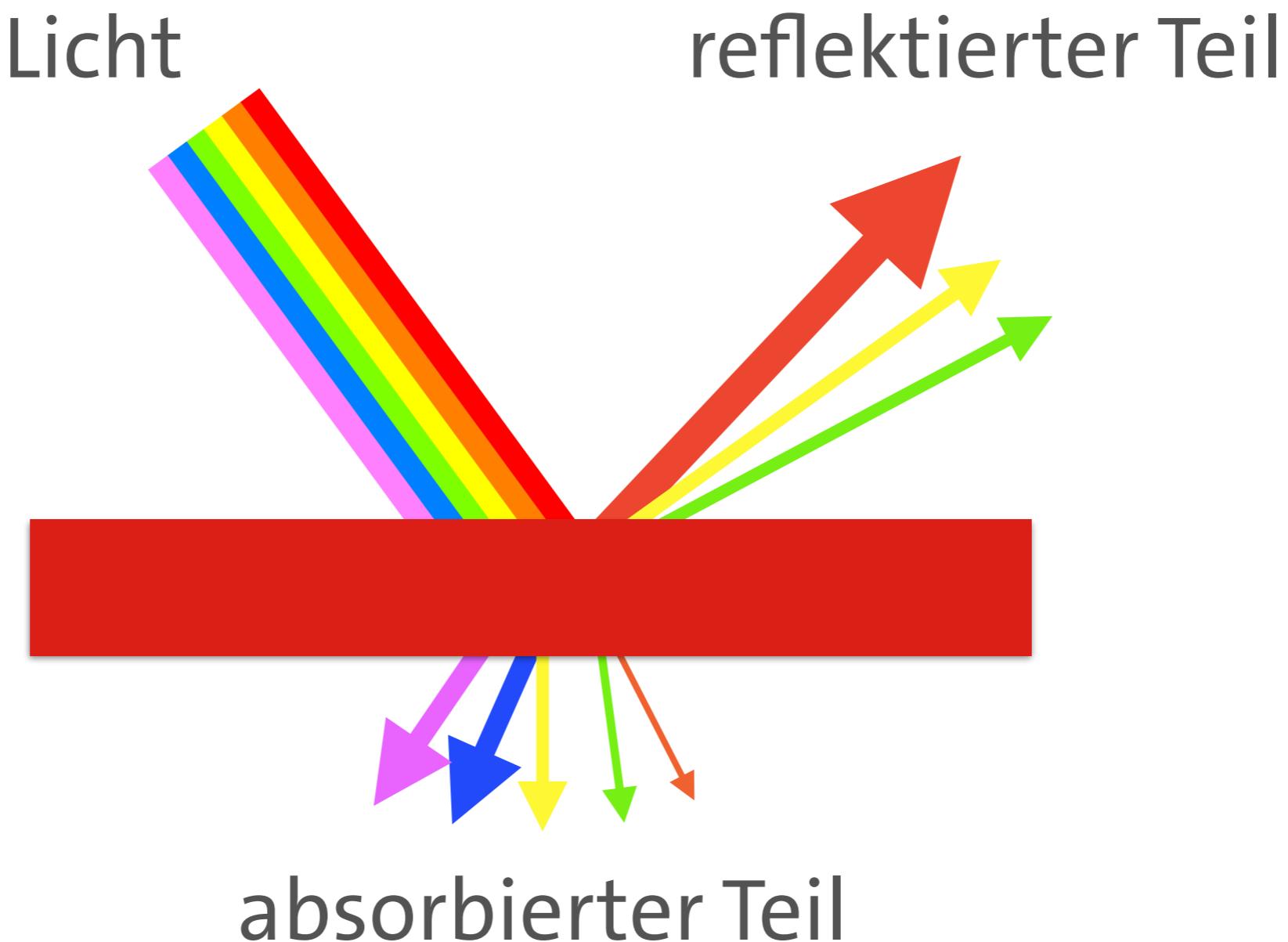


# Lichausbreitung

- **Lichtwellen** breiten sich **transversal** von Quelle in Richtung mit konstanter Geschwindigkeit aus
- Je nach Wellenlänge und Beschaffenheit von Hindernissen, kann Welle **abgelenkt** bzw. **reflektiert**, **absorbiert** oder Hindernis **durchdringen**

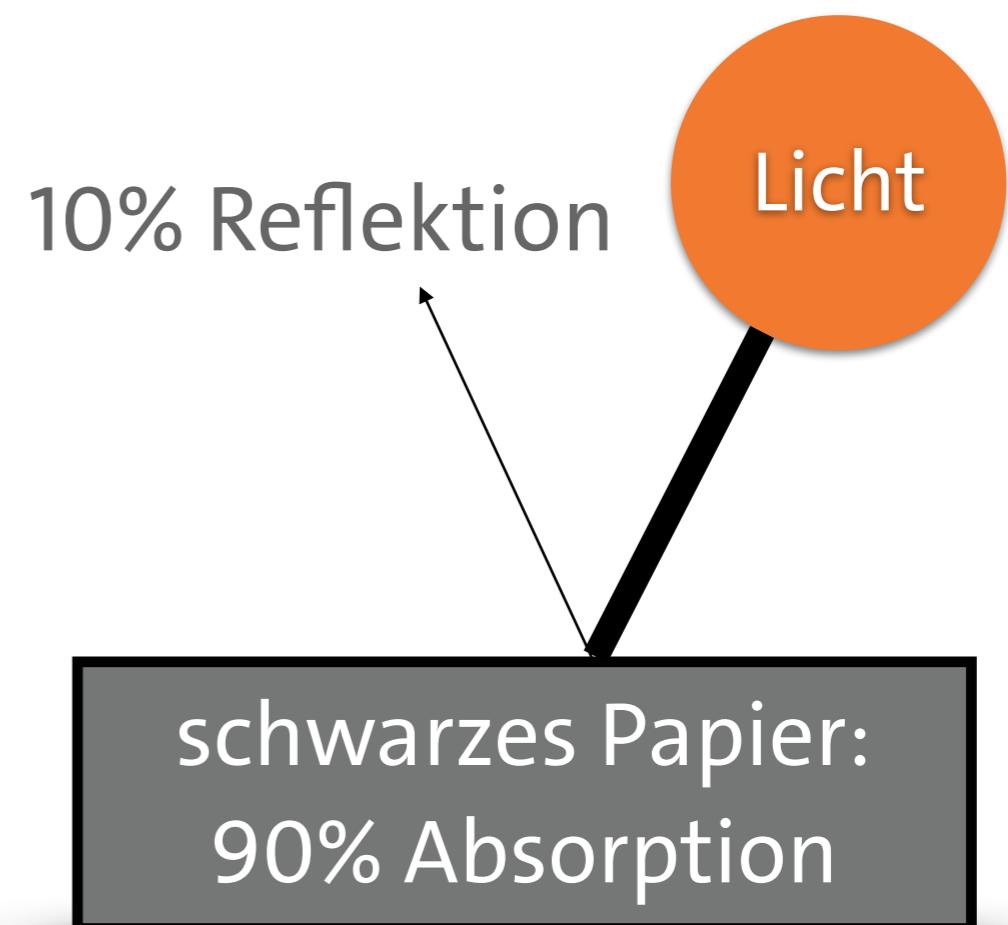
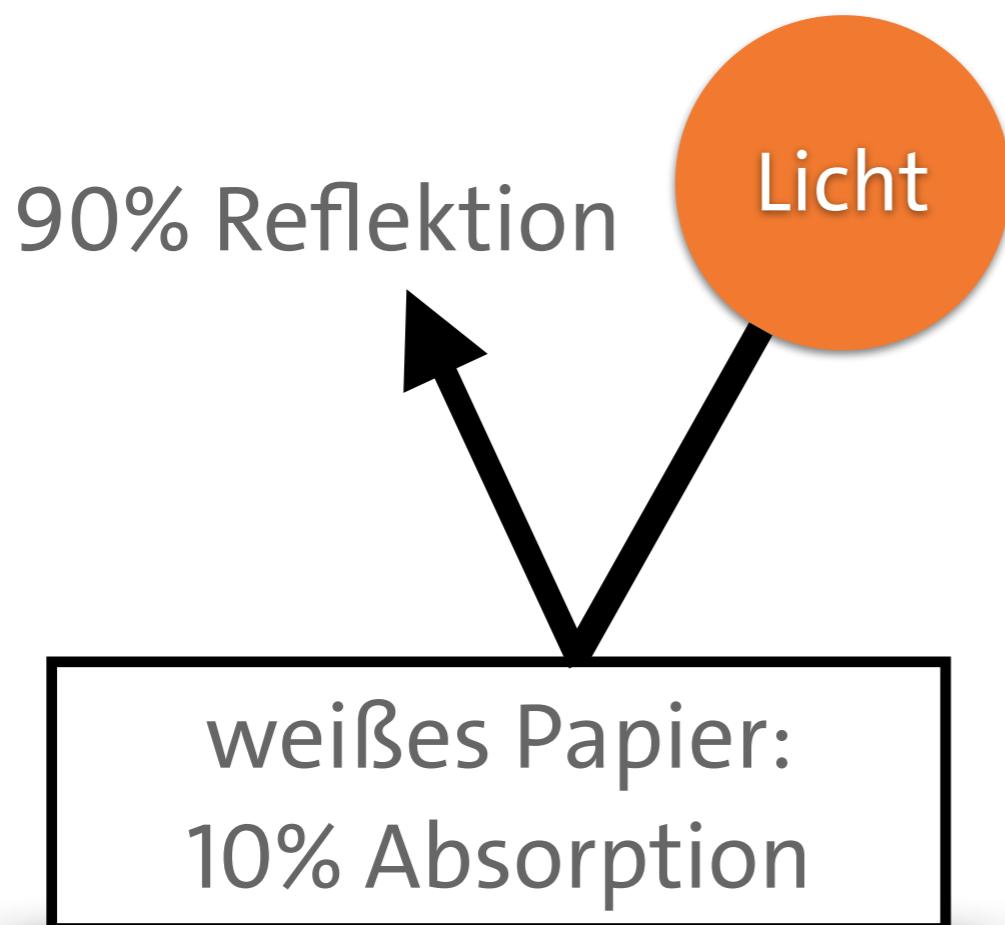
# Absorption & Reflektion

## Beispiel



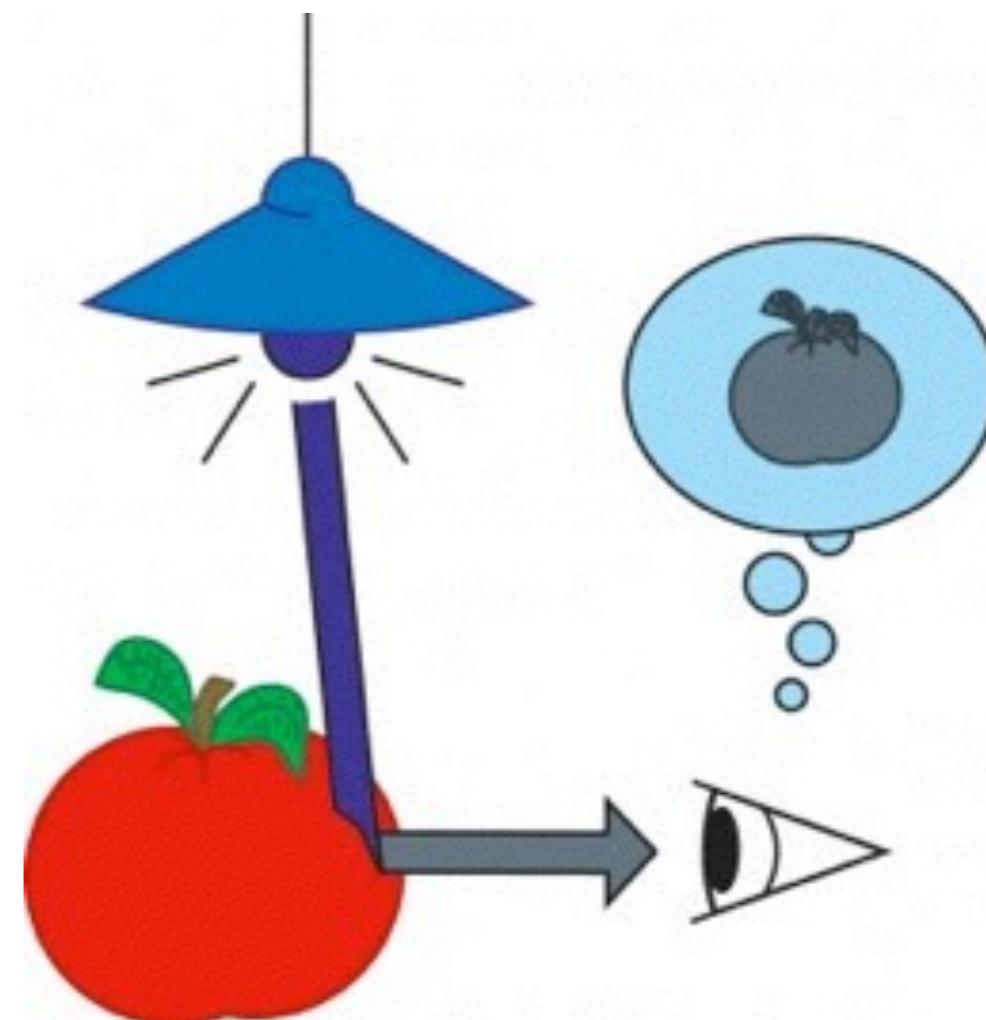
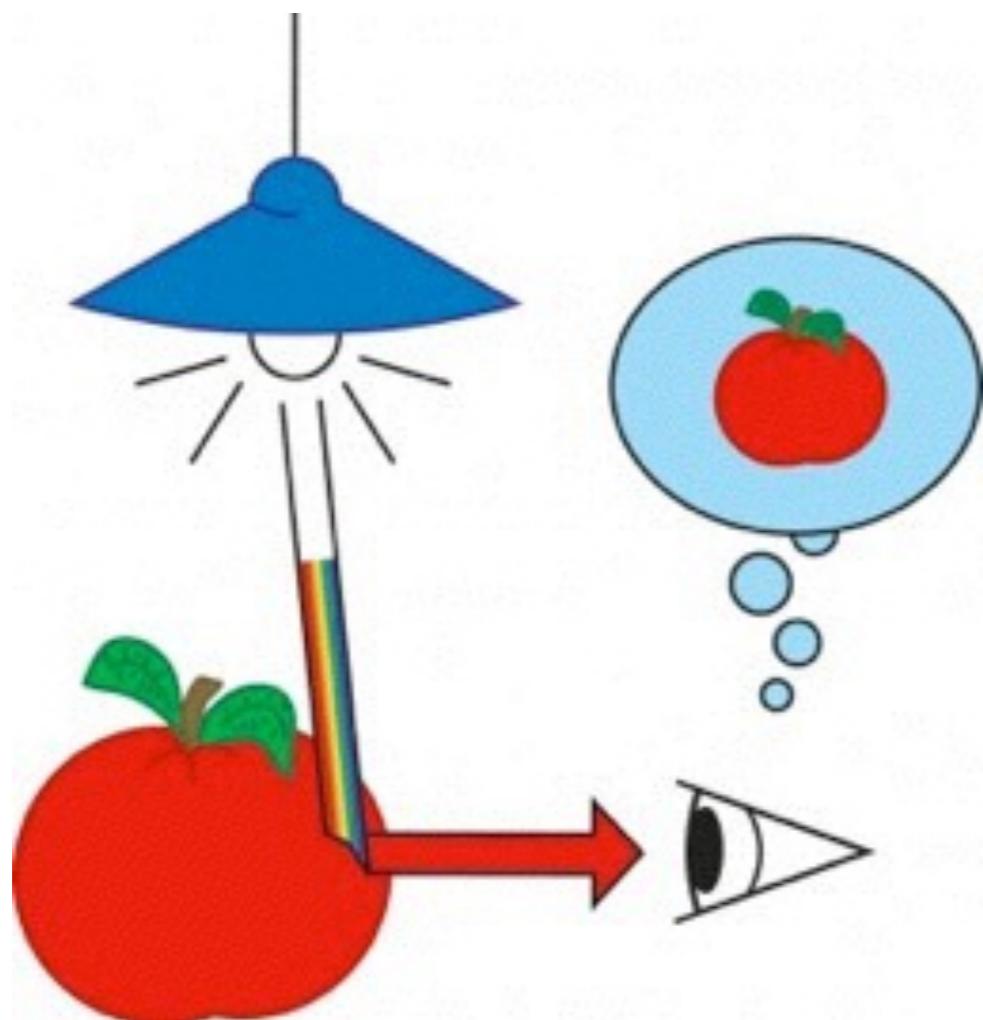
# Absorption & Reflektion

## Beispiel



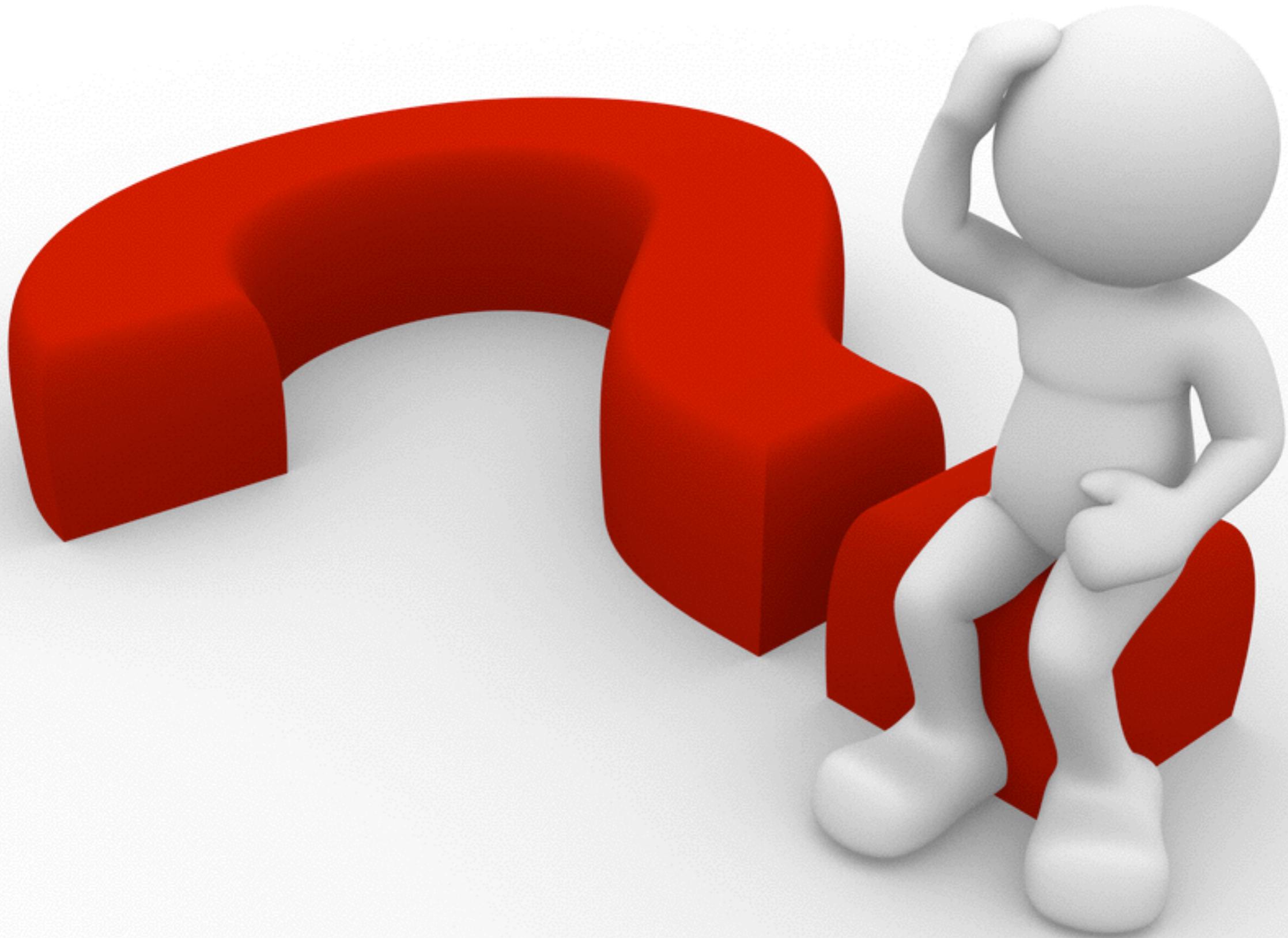
# Absorption & Reflektion

## Beispiel: Tomate



# Wahrnehmung von Licht

- **Licht wird gefiltert durch unser visuelles System wahrgenommen**
- **Licht, das nicht auf Objekt trifft und von dort zu uns (direkt oder indirekt) reflektiert wird, wird nicht gesehen**





# Interaktive Medien

## Kapitel Bilder

### Visuelles System

# Visuelles System

- **Visuelles System** ist Teil des **zentralen Nervensystems** und *interpretiert* Information der sichtbaren Welt
  - Rezeption von Licht
  - Konstruktion von binokularen Bildern
  - Identifikation und Kategorisierung
  - Führung von Benutzerbewegungen
  - ...

# Visuelles System

- **Visuelle System** arbeitet nach **zweistufigem Modell**
  1. **Physikalische Rezeption** von Stimuli der sichtbaren Welt
  2. **Verarbeitung und Interpretation** des Stimulus

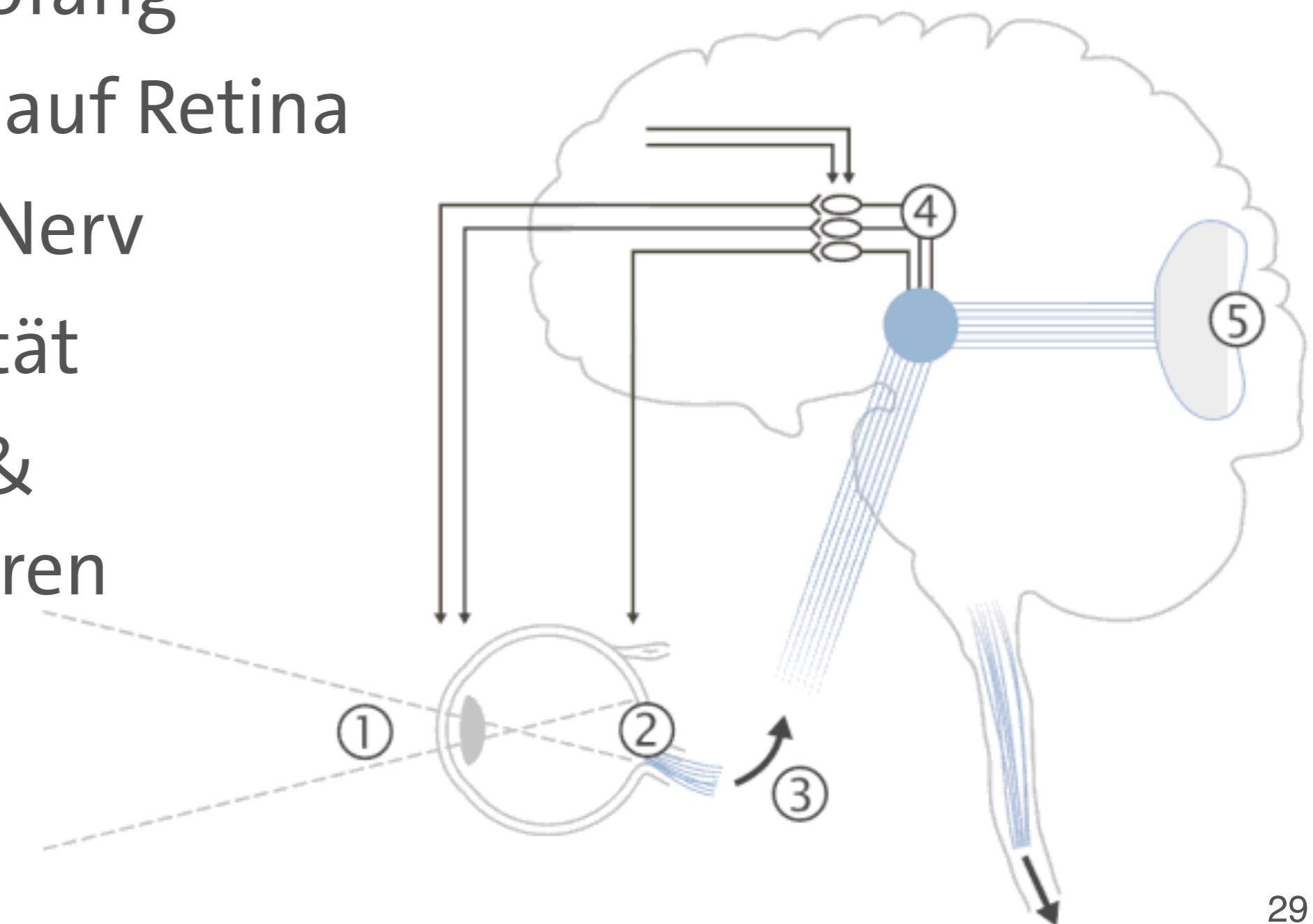
# Visuelles System

- Licht wird von Objekten in realer Welt **reflektiert**
- Bild wird **umgekehrt** auf **Retina** abgebildet
- **Rezeptoren** transformieren Licht-Signale in **elektrische Signale**, die an Gehirn weitergegeben werden

# **Animation des visuellen Systems**

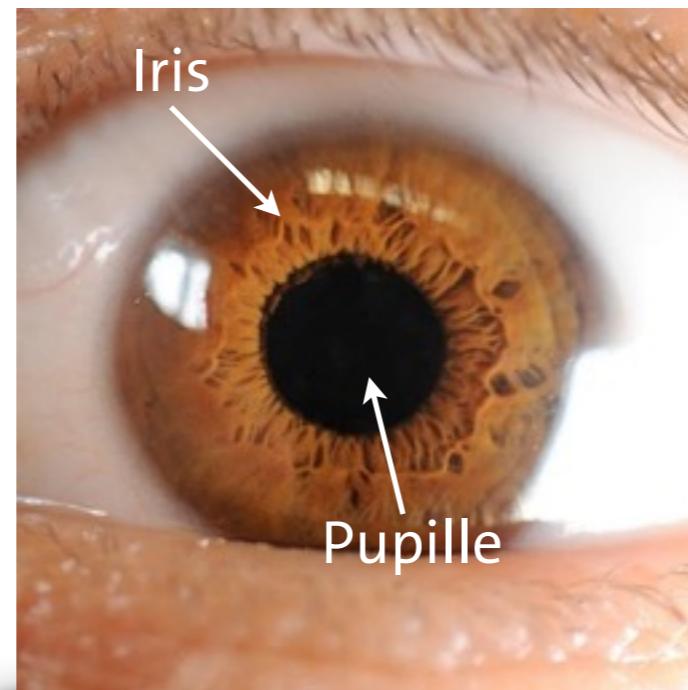
# Visuelles System

- (1) Signal-Empfang
- (2) Projektion auf Retina
- (3) Optischer Nerv
- (4) Binokularität
- (5) Erkennen & Interpretieren

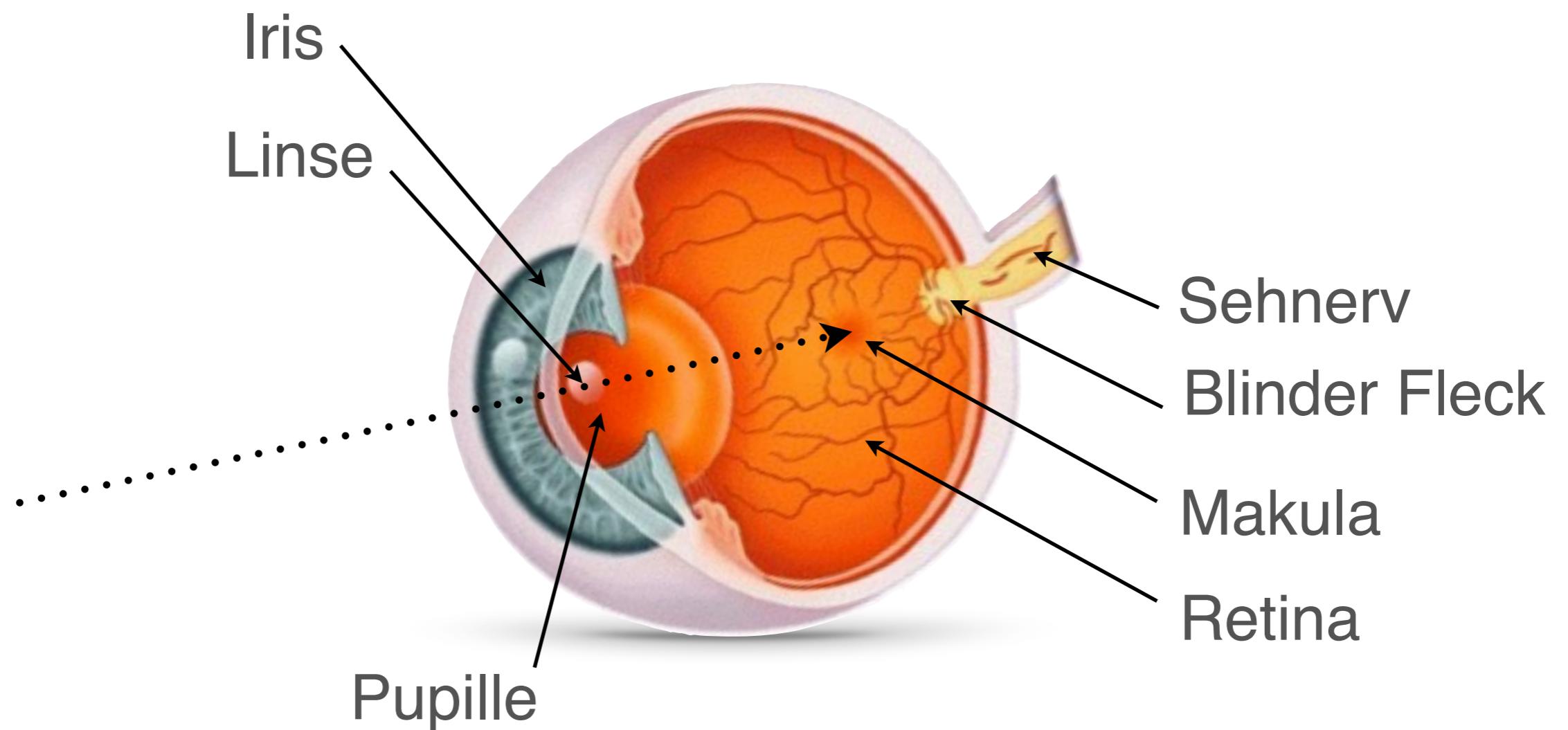


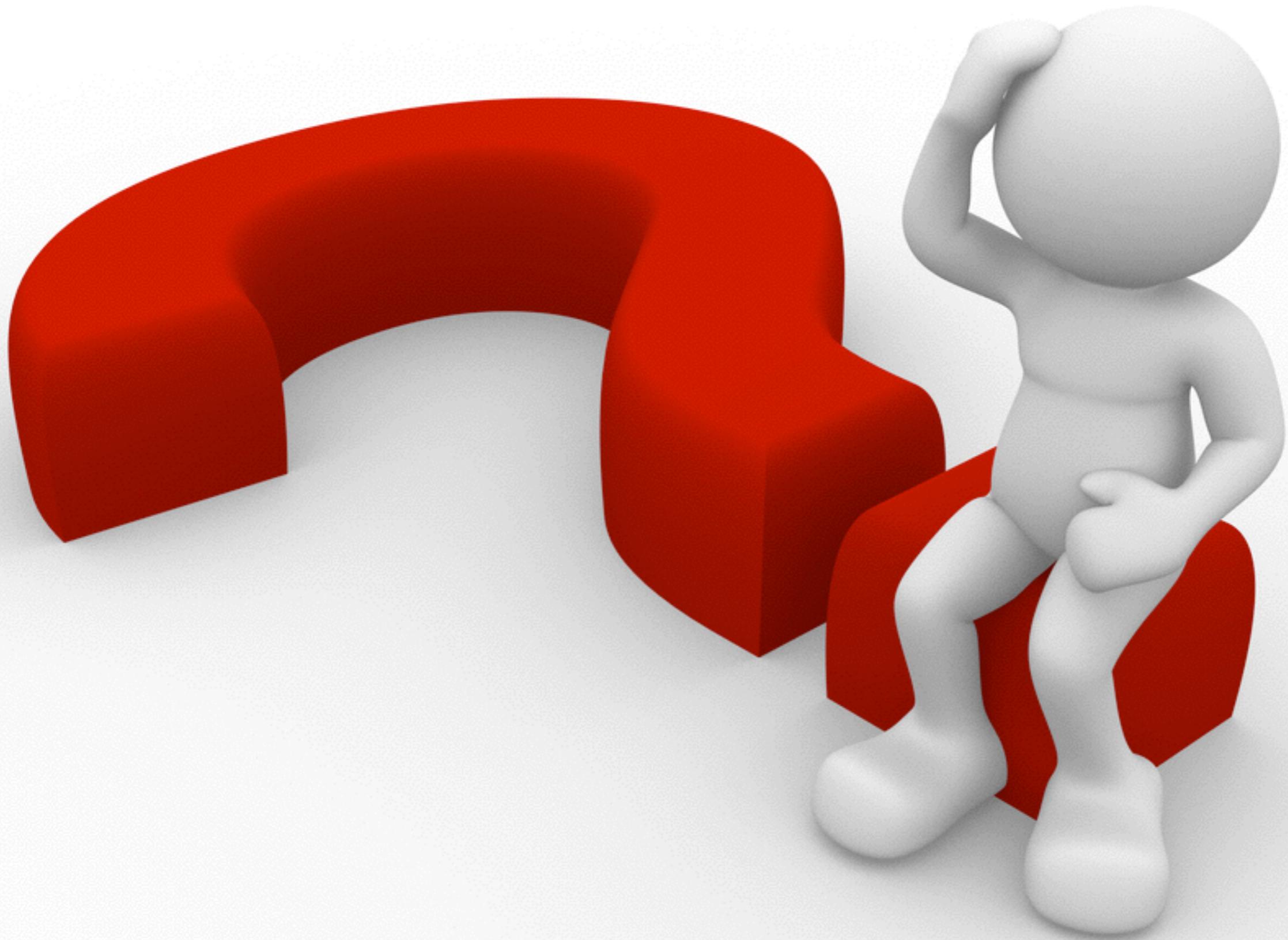
# Menschliches Auge

- **Menschliche Auge** ist Teil des visuellen Systems
- **Menschliche Auge** ist Sinnesorgan zur Wahrnehmung von Lichtreizen



# Menschliches Auge







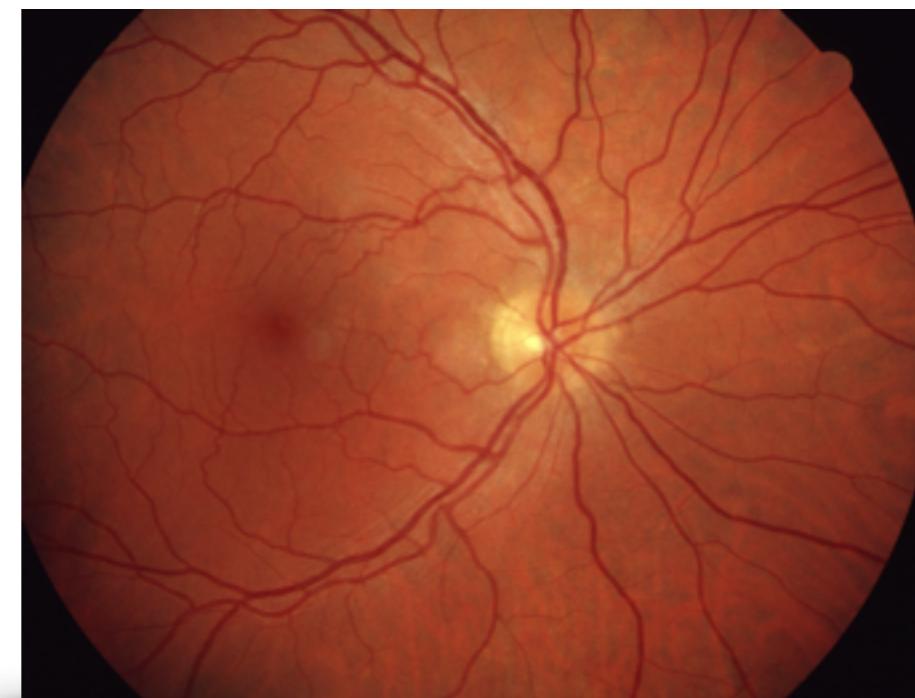
# Interaktive Medien

## Kapitel Bilder

### Farbwahrnehmung

# Retina (Netzhaut)

- Retina ist lichtempfindliche Rückwand des Auges
- Nervenzellen im Auge wandeln Lichtwellen in neuronale Aktivitäten

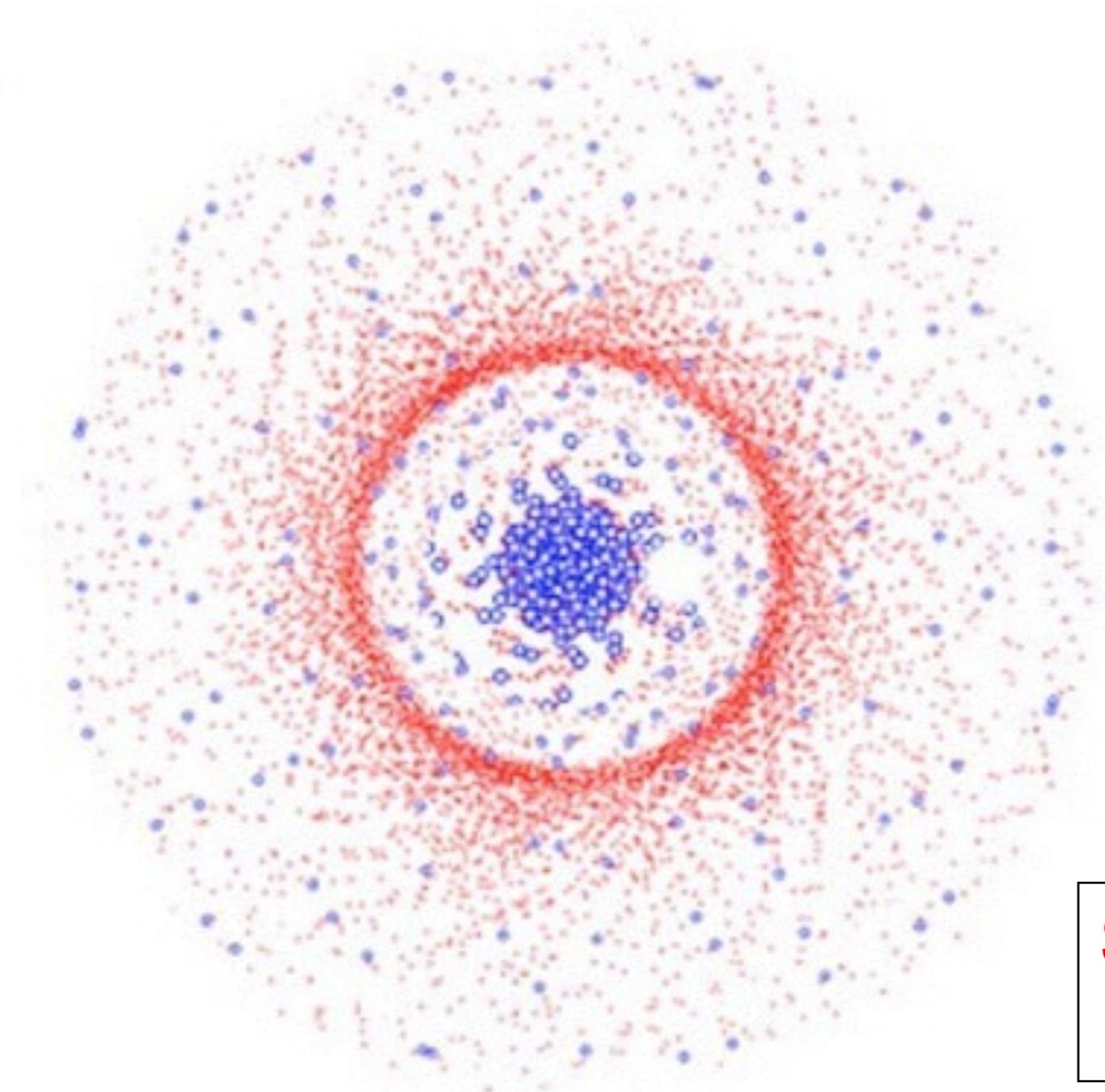


# Dreifarbtentheorie

- nach **Dreifarbtentheorie** enthält Retina zwei Arten von Fotorezeptoren
  1. **Zapfen** (7-8 Mio) (engl. *Cones*): Farb- und Tagessehen (**Photopische Sehen**), 3 Typen (S, M, L) unterschiedlicher Wellenlängensensitivität
  2. **Stäbchen** (100-120 Mio.) (engl. *Rods*): Hell-Dunkel-Empfindung, Dämmerungssehen (**Skotopischen Sehen**)

# Stäbchen & Zapfen

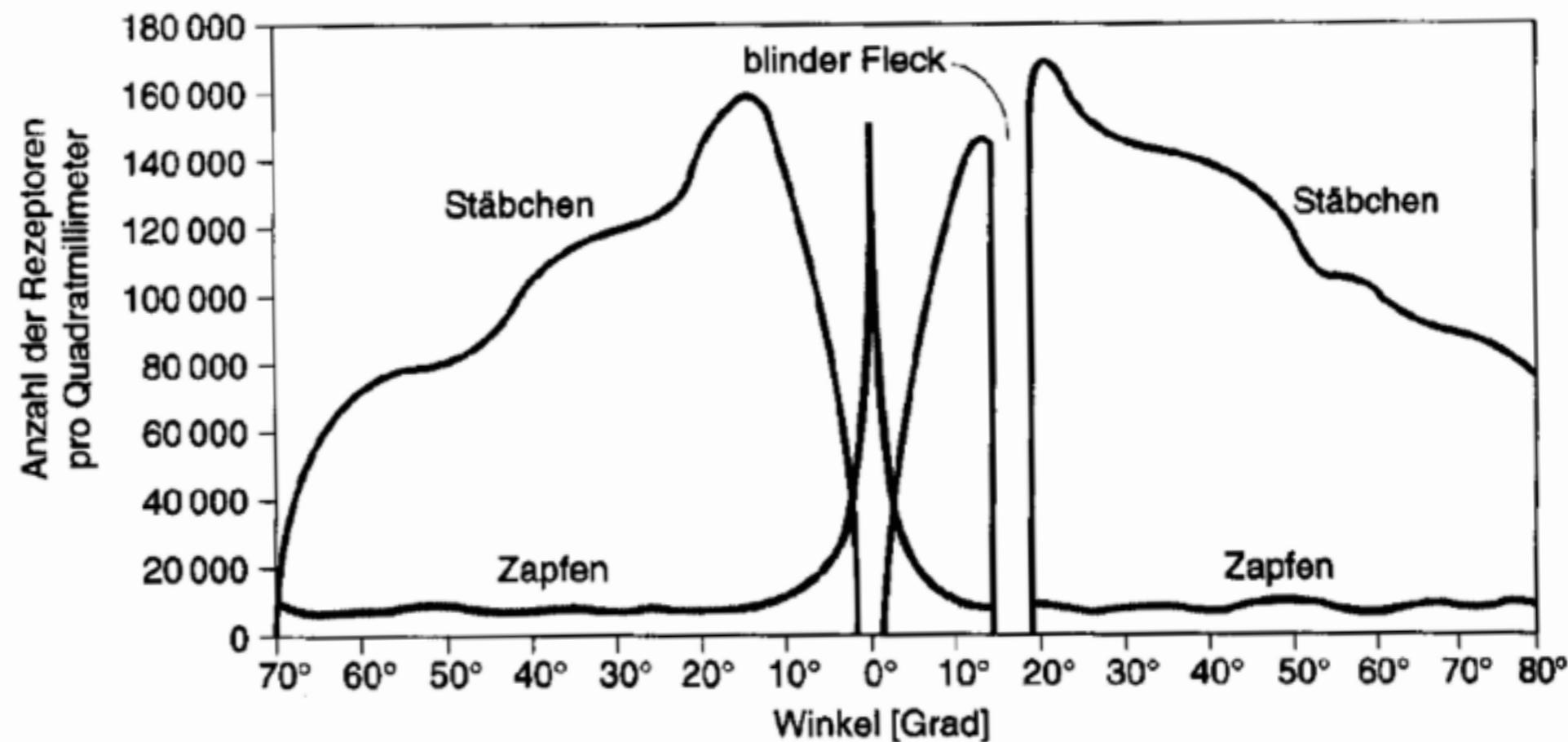
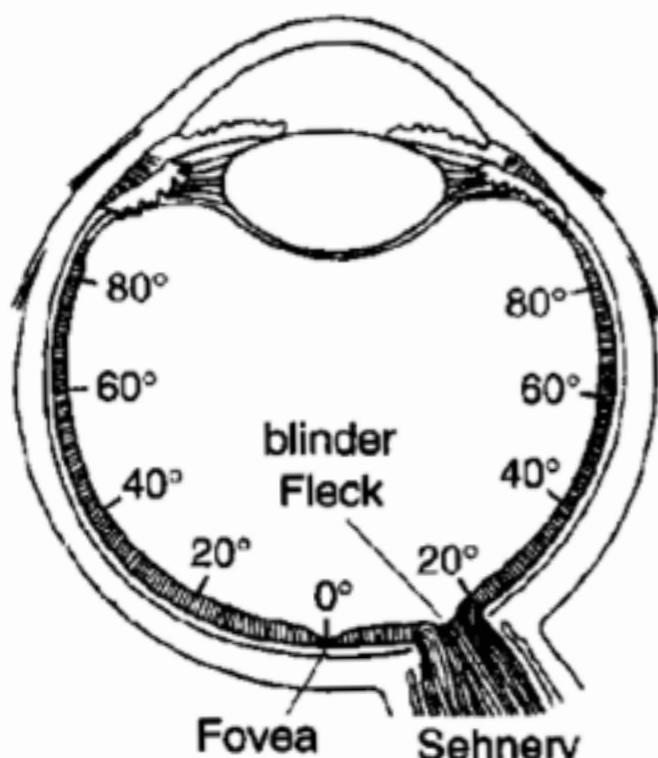
## Verteilung



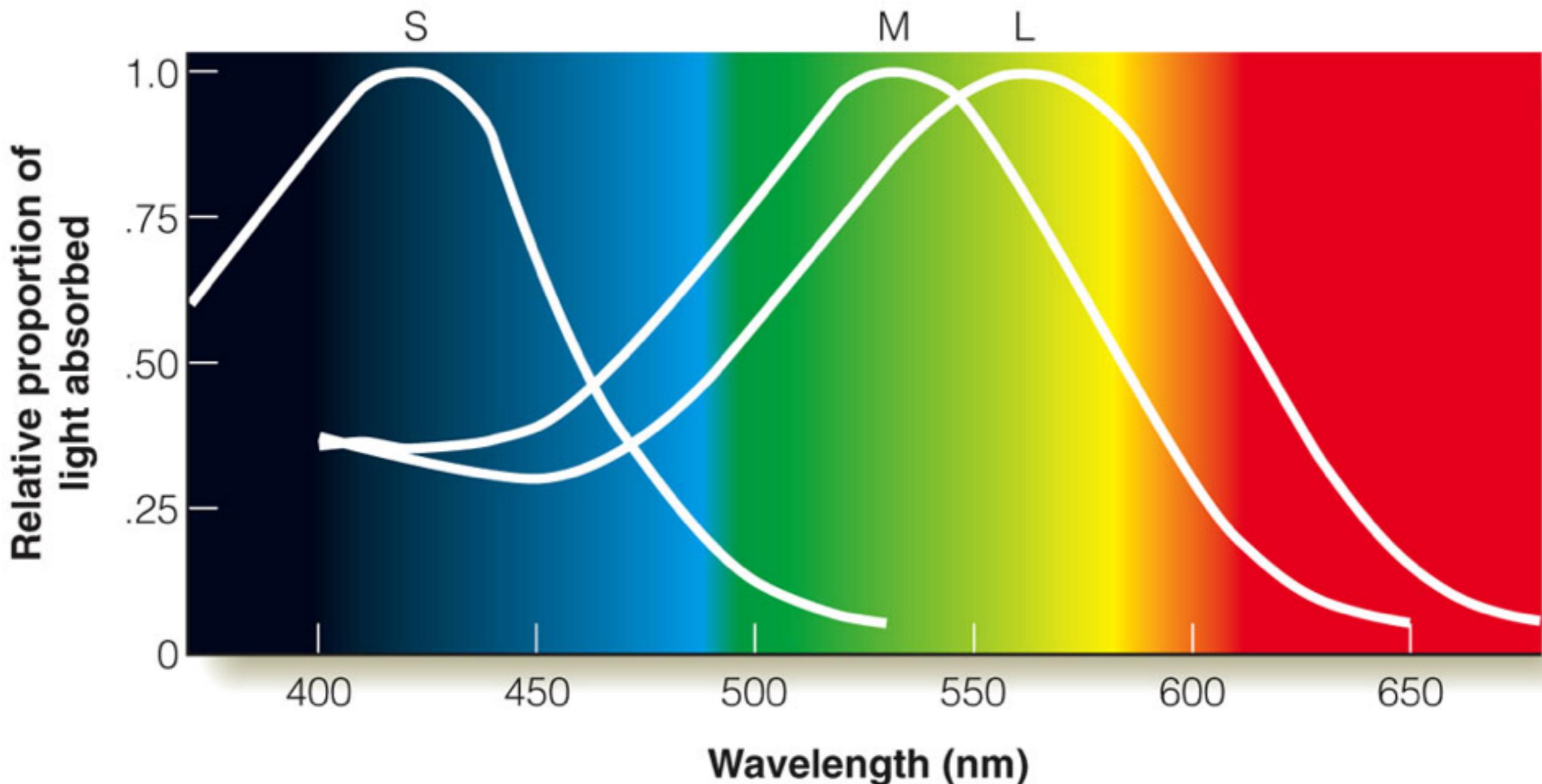
Stäbchen (rot)  
Zapfen (blau)

# Stäbchen & Zapfen

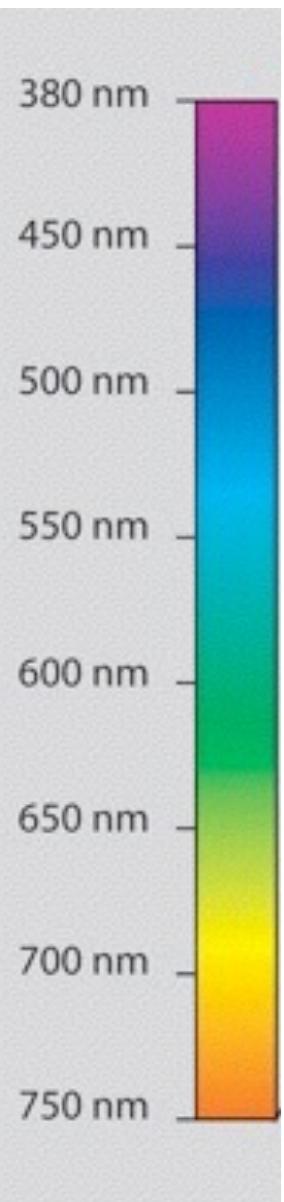
## Verteilung



# Zapfen Spektren



# Menschliches Auge



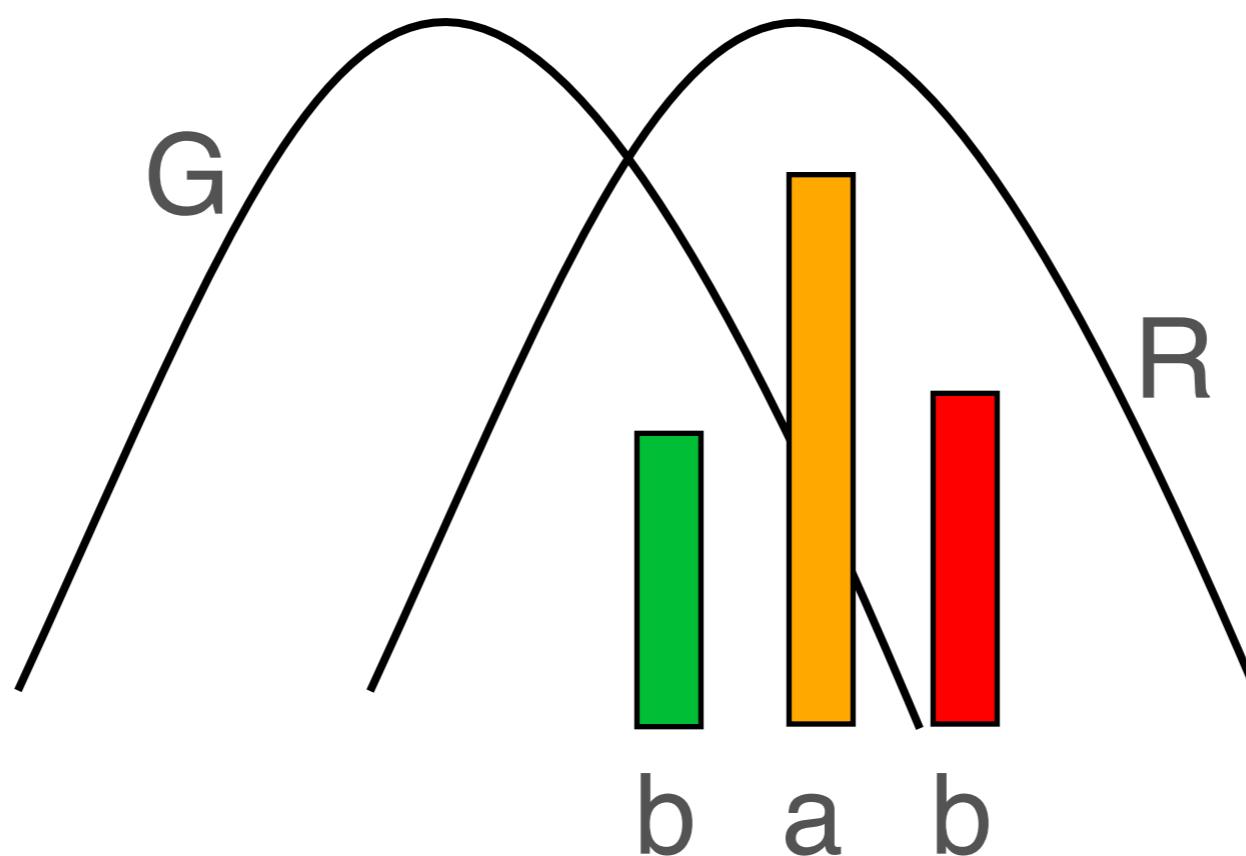
- Differenzierung der Farbtöne:
  - 10nm (Extrembereich im Spektrum)
  - 2nm bei 480nm – 580nm
  - 4nm sonst
- Unterscheidung von ca. 300 *Farbtönen*
- Unterscheidung von 100-150 *Helligkeitsstufen*
- Sensitivität nimmt mit fallender *Sättigung* ab

# Dreifarbtentheorie

- Dreifarbtentheorie von Thomas Young und Hermann von Helmholtz besagt, dass jede von uns wahrnehmbare Farbe sich durch Kombination von Licht von nur **drei verschiedenen Wellenlängen** herstellen lässt

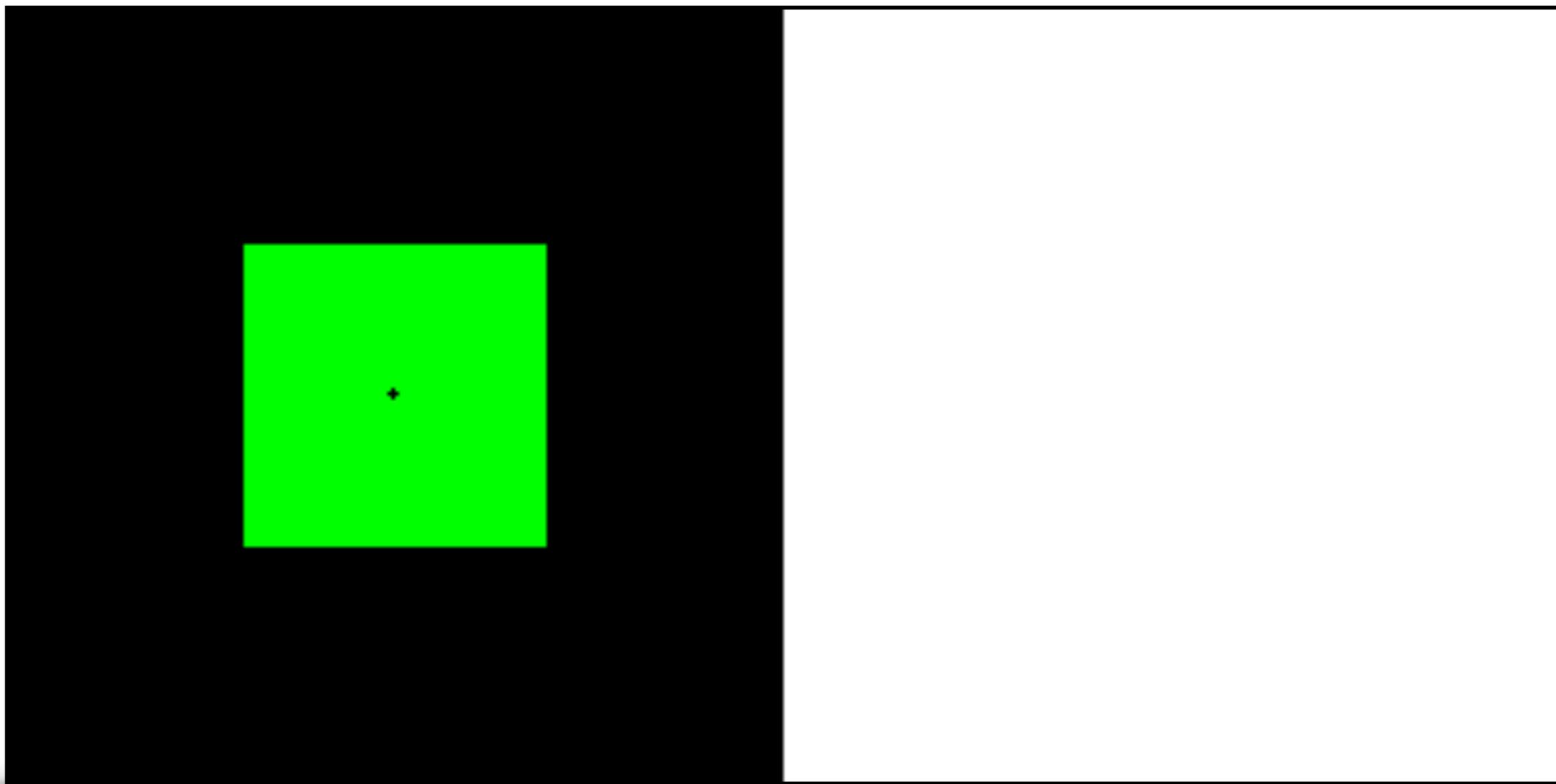
# Farbwahrnehmung

- Sei a Licht einer Frequenz und b Signal aus zwei sich überlagernden Frequenzen



→ a und b liefern gleiches Orange

# Gegenfarbtheorie

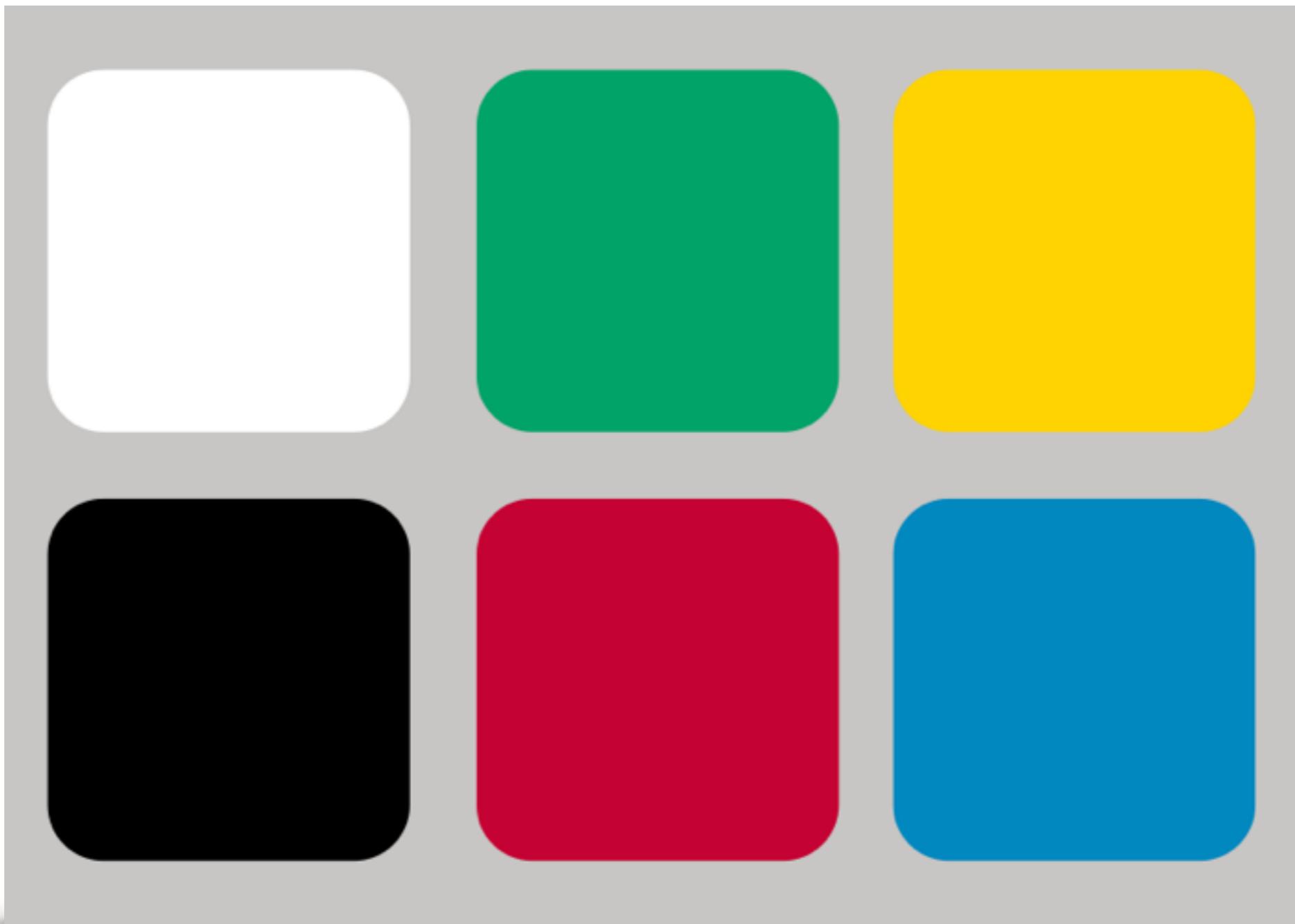


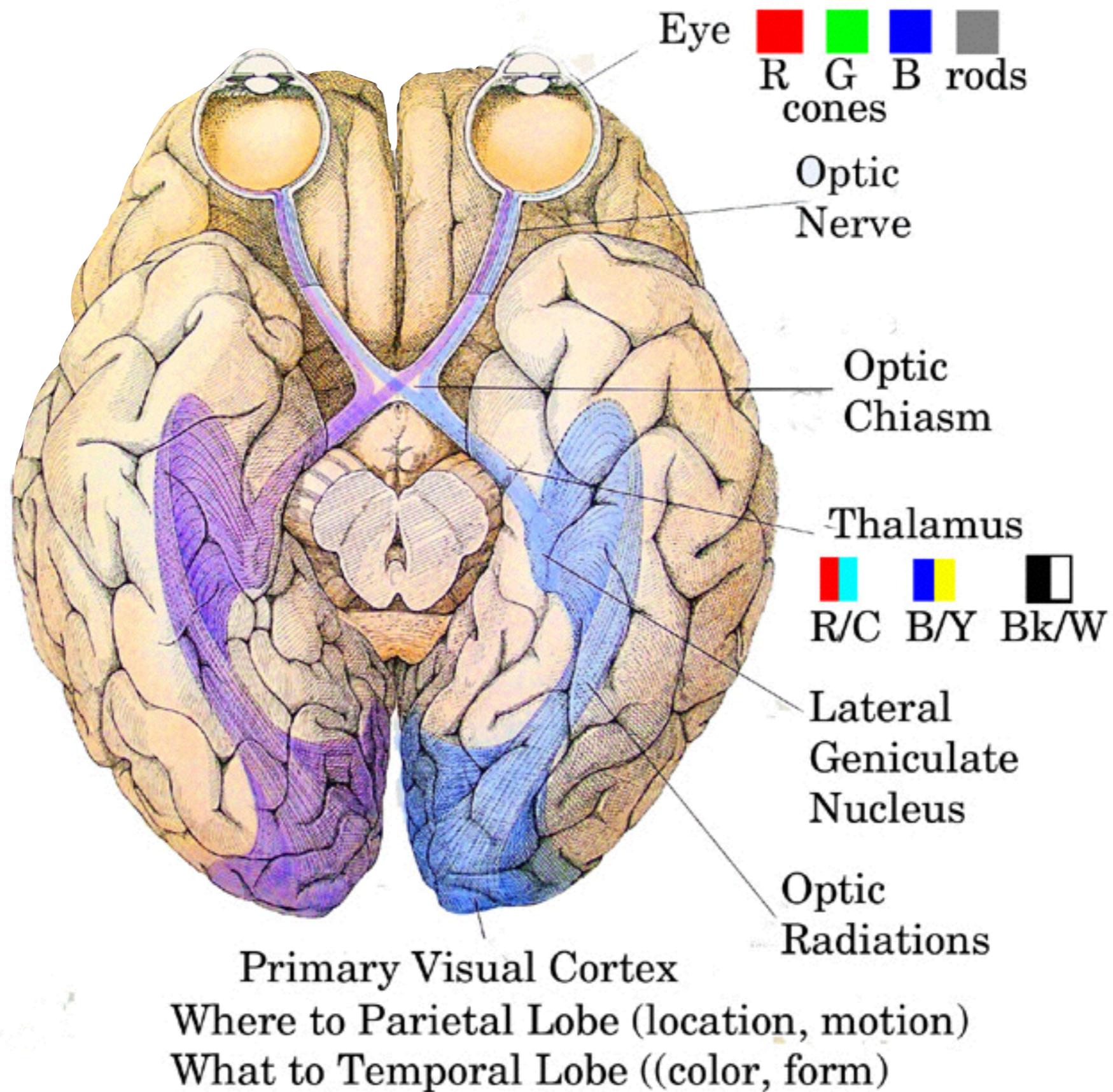
# Gegenfarbtheorie

- **Gegenfarbtheorie** nach **Ewald Hering** besagt dass drei getrennte Prozesse in Netzhaut mit je zwei Gegenfarben, mit je einem hemmenden und einem erregenden Anteil, die nach einem Gleichgewicht streben
- **Gegenfarbpaare** sind Blau–Gelb, Rot–Grün und Schwarz–Weiß

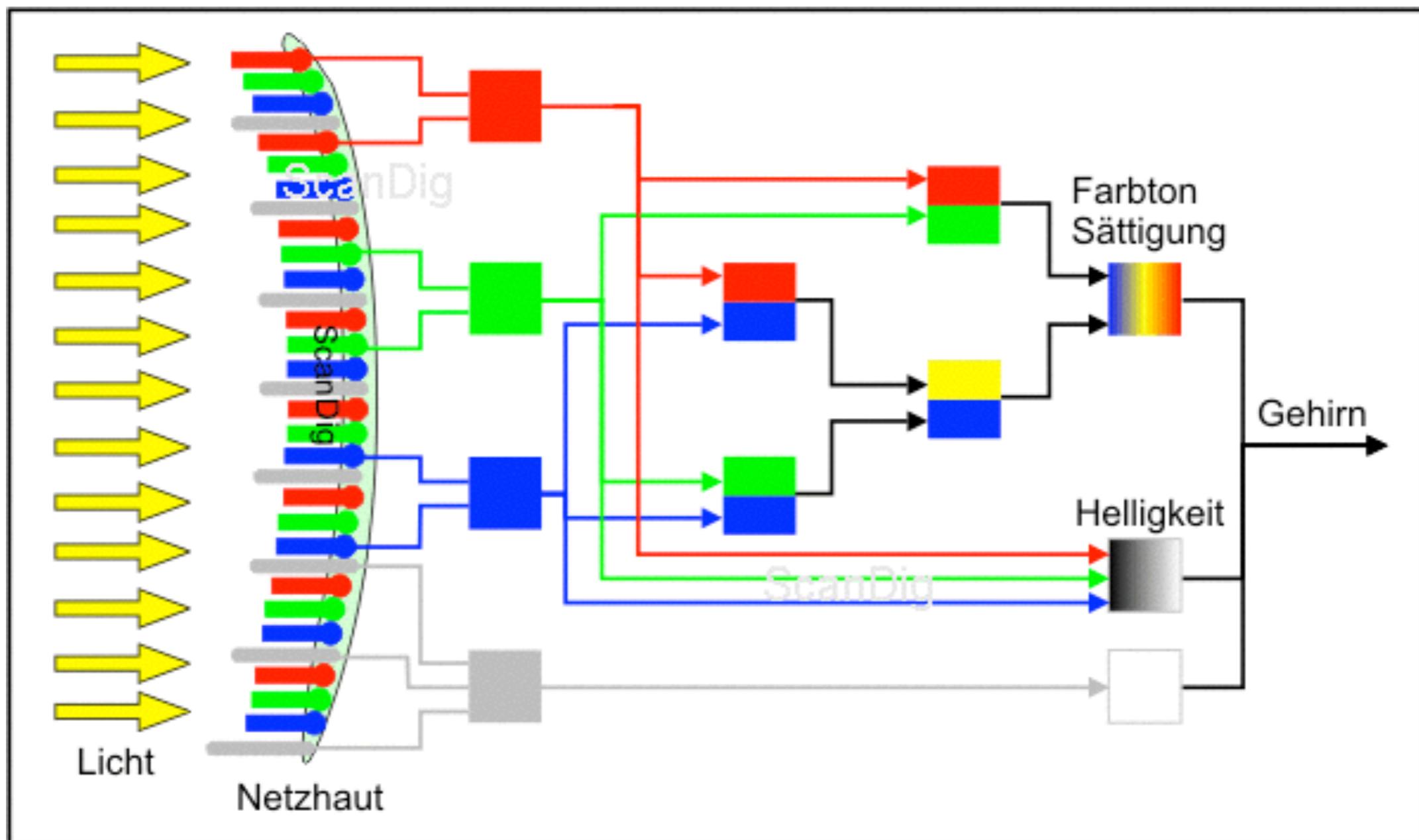
# Gegenfarbtheorie

## Gegenfarbpaare





# Farbwahrnehmung



# Gegenfarben

## Beispiele



Original



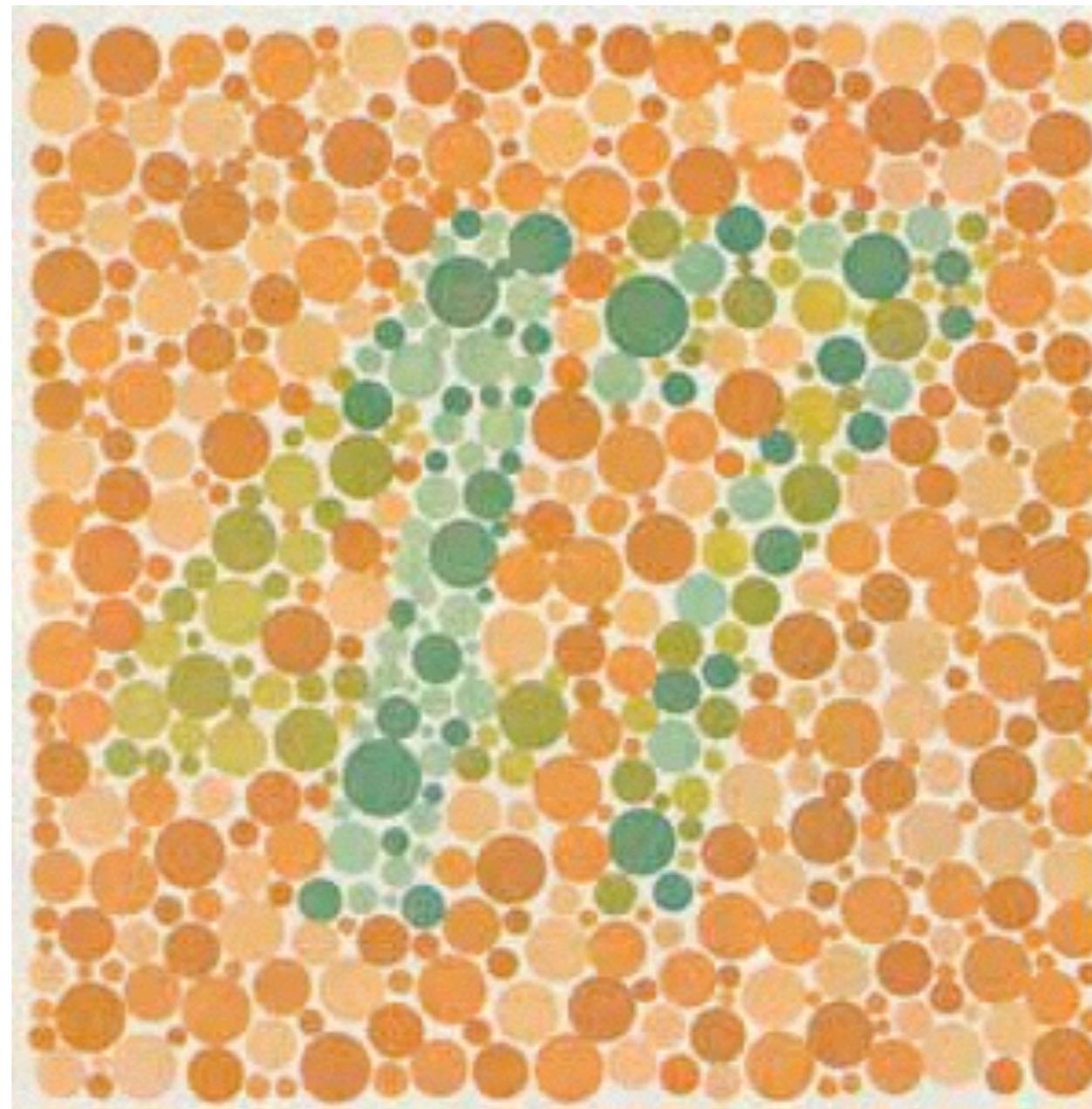
Rot-Grün-Kanal



Blau-Gelb-Kanal

# Rot/Grün Schwäche

## Beispiel

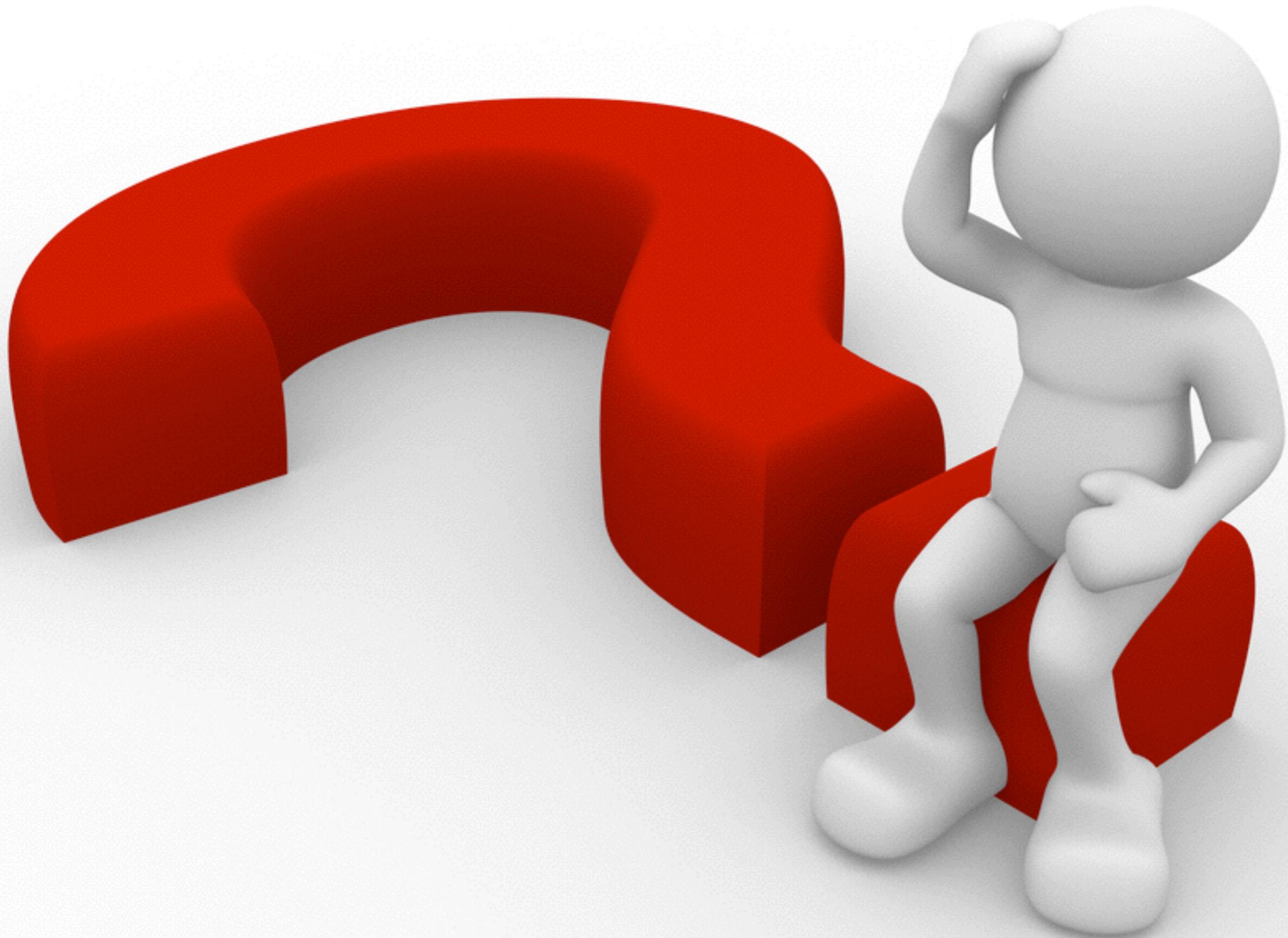


# Farbfehlsichtigkeiten

Beispiel: Rot-Grün-Sehschwäche

- **Rot-Grün-Sehschwäche/-Blindheit** steht für Farbfehlsichtigkeit, bei der Betroffene Farben Rot und Grün schlechter als Normalsichtige unterscheiden können







# Interaktive Medien

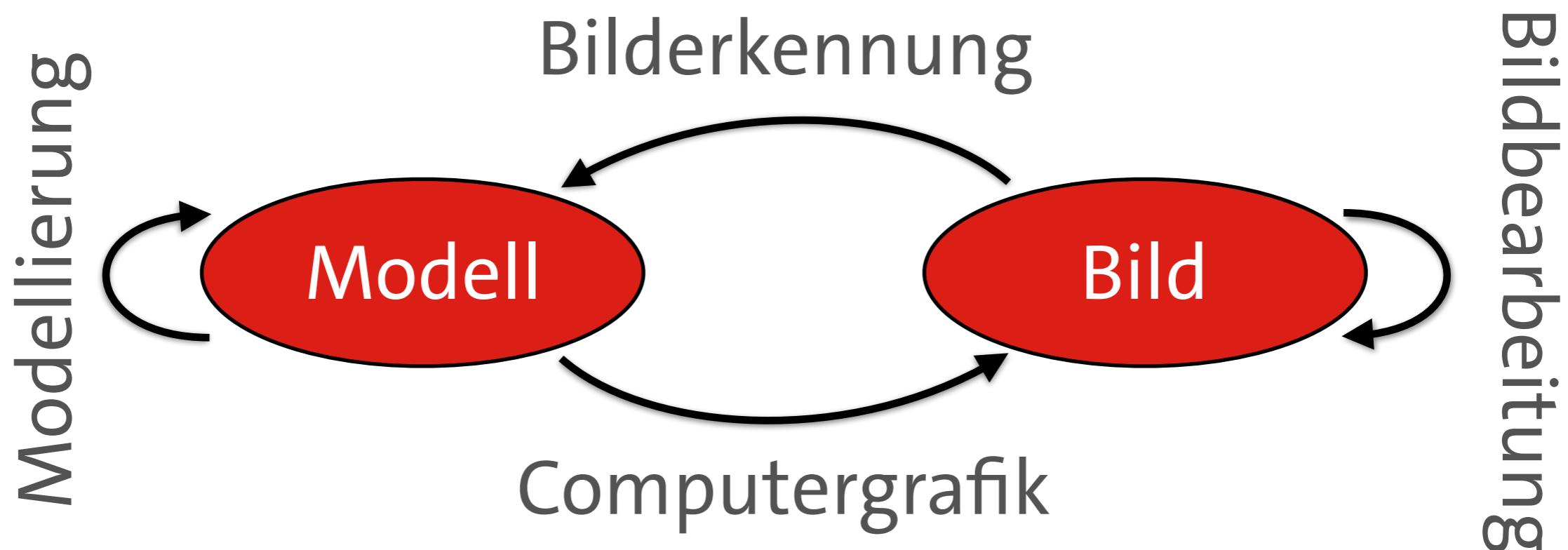
## Kapitel Bilder

### Digitale Bilder

# Visual Computing

- **Visual Computing** umfasst Teilgebiete der informationstechnischen Bildaufnahme (**Bildakquisition**), Verarbeitung und Analyse von Bilddaten (**Bildanalyse**) und Erzeugung von Bildern (**Bildsynthese**)

# Visual Computing



# Visual Computing

## Bereiche

- **Computergrafik** rendert Bilder aus Modellen oder Bildbeschreibungen
- **Bildverarbeitung** erzeugt aus Bildern Bilddaten oder Informationen
- **Bilderkennung** interpretiert Modelle aus Bildern
- **Bildbearbeitung** manipuliert Bilder
- **Modellierung** bearbeitet Modelle

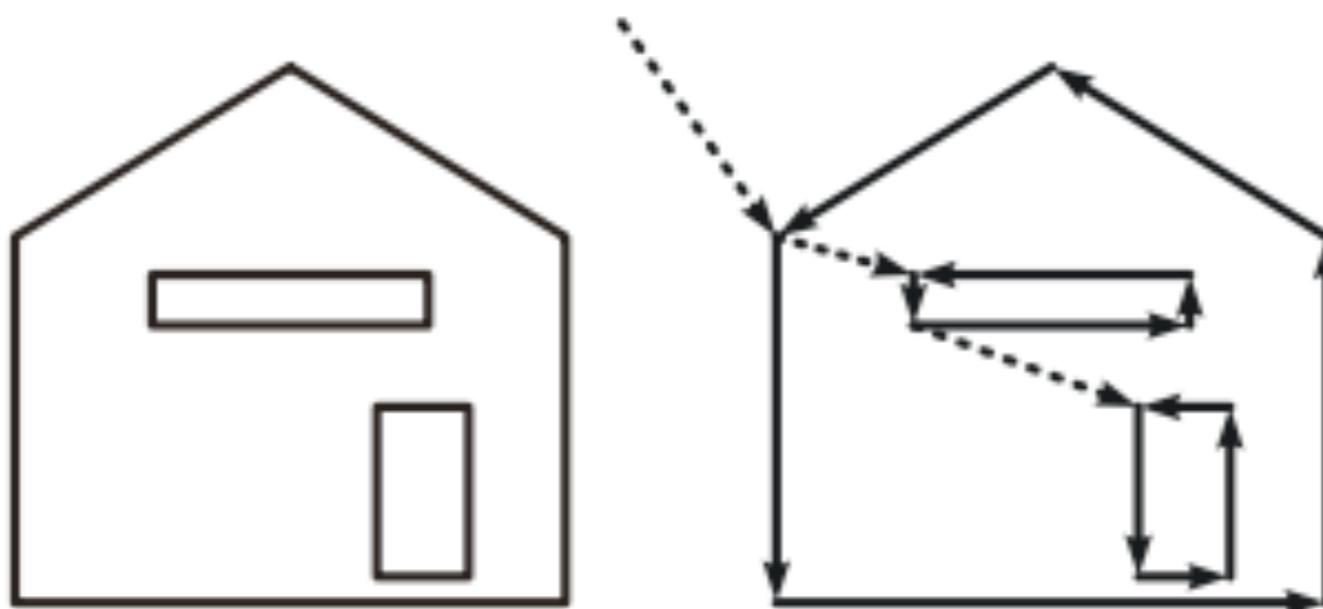
# Digitale Bilder

- Resultat der computergestützten Bildsynthese oder A/D-Wandlung ist **digitales Bild**
- Darstellung digitaler Bilder gemäß zweier Paradigmen
  1. **Vektorgrafik**
  2. **Rastergrafik**

# Digitale Bilder

## Paradigmen

- **Vektorgrafik** ist aus **grafischen Primitiven** wie Linien, Kreisen, Polygonen oder allgemeinen Kurven (Splines) zusammengesetzt



# Digitale Bilder

## Paradigmen

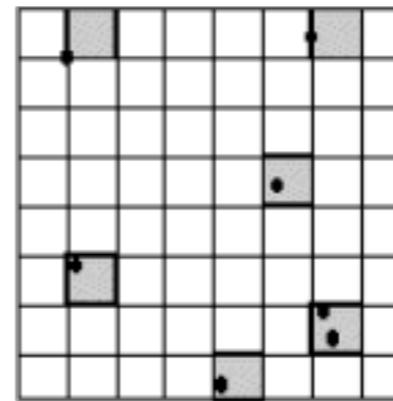
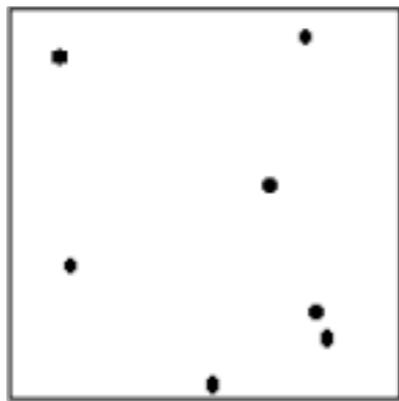
- **Rastergrafik** bestehen aus **rasterförmigen** Anordnung von **Bildpunkten (engl. Pixel)**, denen jeweils Farbe zugeordnet ist



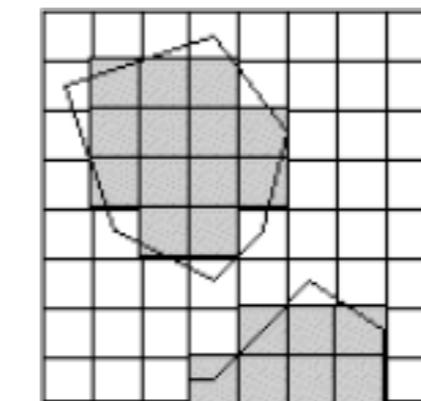
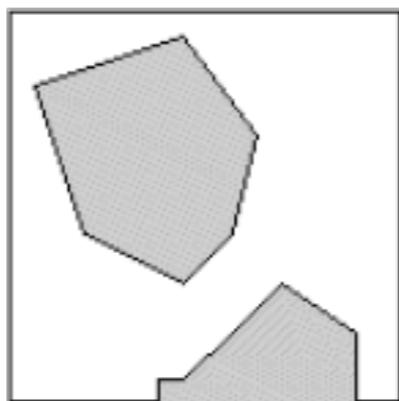
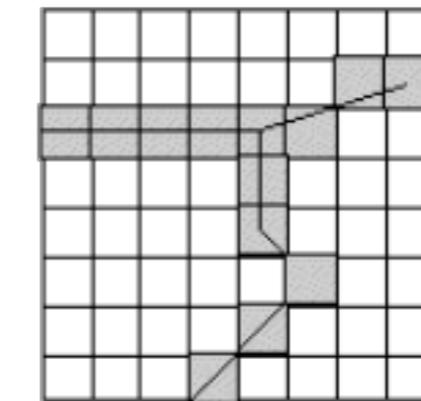
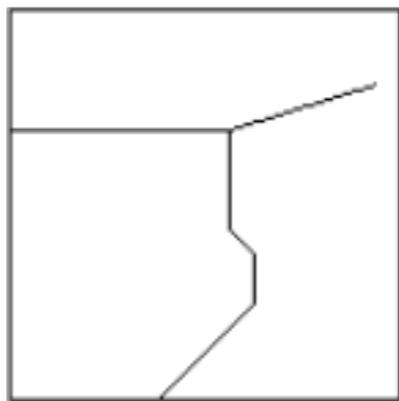
# Digitale Bilder

## Paradigmen

Vektorgrafiken

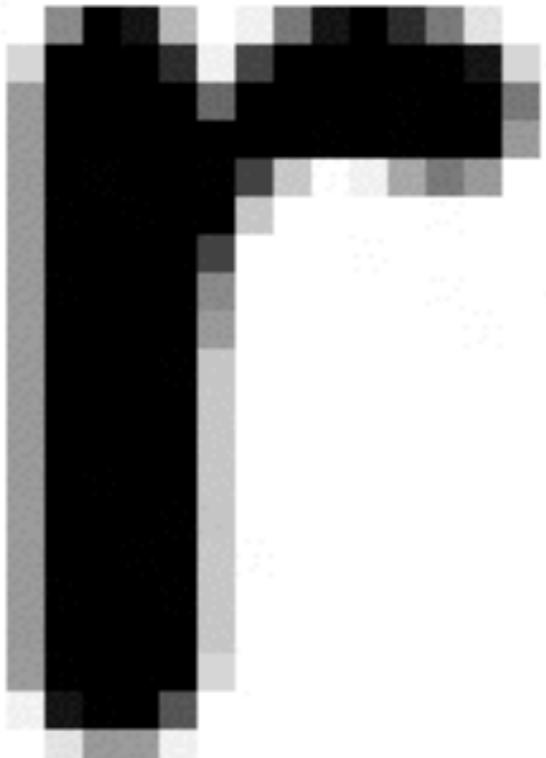


Rastergrafiken

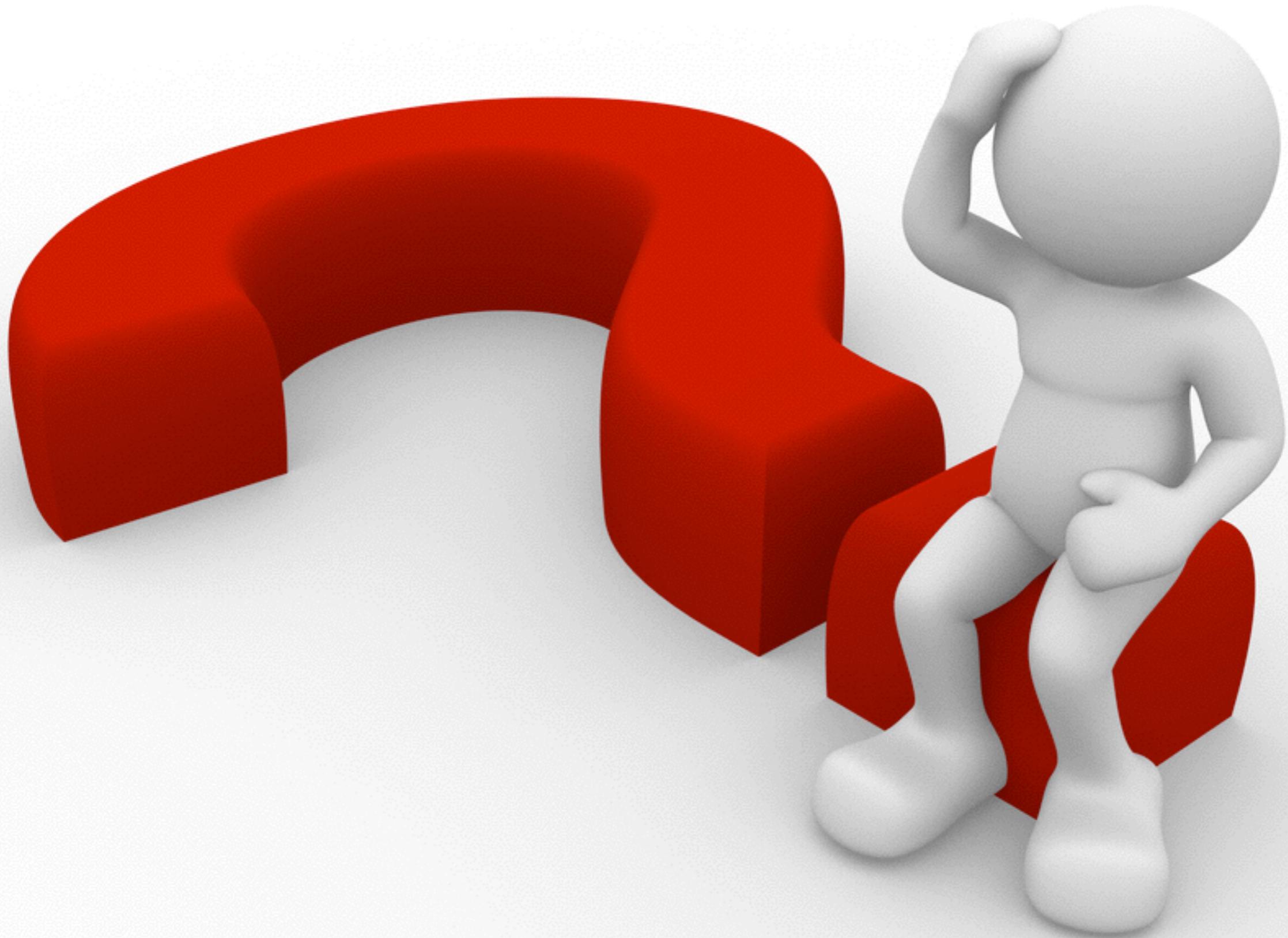


# Digitale Bilder Paradigmen

Rastergrafiken



Vektorgrafiken





# Interaktive Medien

## Kapitel Bilder

## Display-Technologien

# Vektorbildschirm

- **Vektorbildschirm** ist auf Kathodenstrahlröhre basierender Bildschirmtyp, der grafische Primitive, z.B. Punkte und Linien, direkt mit frei beweglichem **Kathodenstrahl** ohne festen Strahlweg darstellt

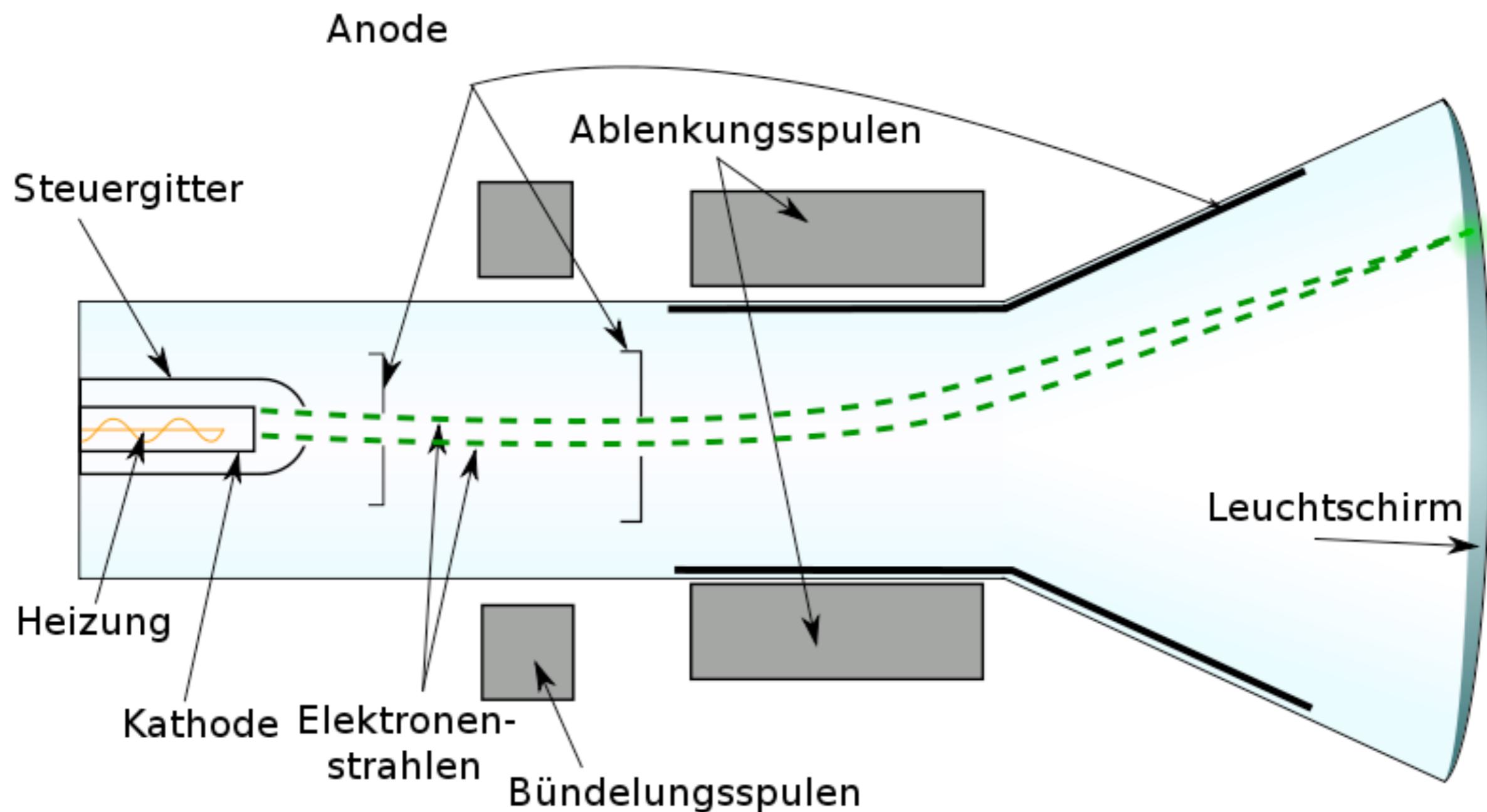
# Vektorbildschirm

## Kathodenstrahlröhre

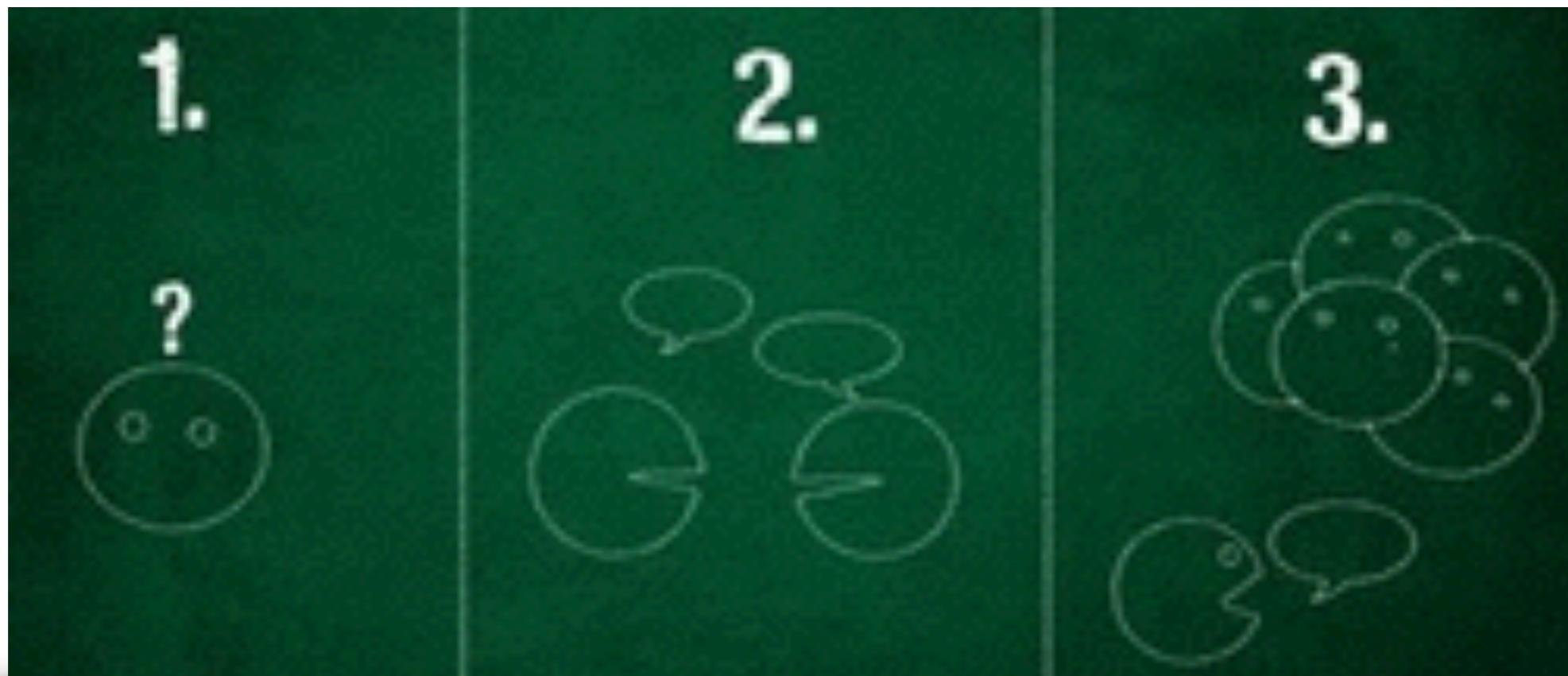
- Kathodenstrahlröhre (*engl. cathode ray tube, CRT*) ist Elektronenröhre, die gebündelten Elektronenstrahl erzeugt
- Elektronenstrahl wird mittels magnetischer oder elektrischer Felder abgelenkt bzw. moduliert
- Bild wird erzeugt, wenn Elektronenstrahl auf Leuchtstoff-Schicht trifft

# Vektorbildschirm

## Kathodenstrahlröhre



# Think! Pair! Share!



Was sind Vor- bzw. Nachteile von Vektorbildschirmen?

# Vektorbildschirm

## Eigenschaften

- Vorteile
  - geringer Speicheraufwand (Display-Liste)
  - keine Aliasingeffekte
- Nachteile
  - begrenzte Komplexität darstellbarer Information (Vektordarstellung)
  - Flimmern bei viel Bildinhalt

# Vektorbildschirm Eigenschaften



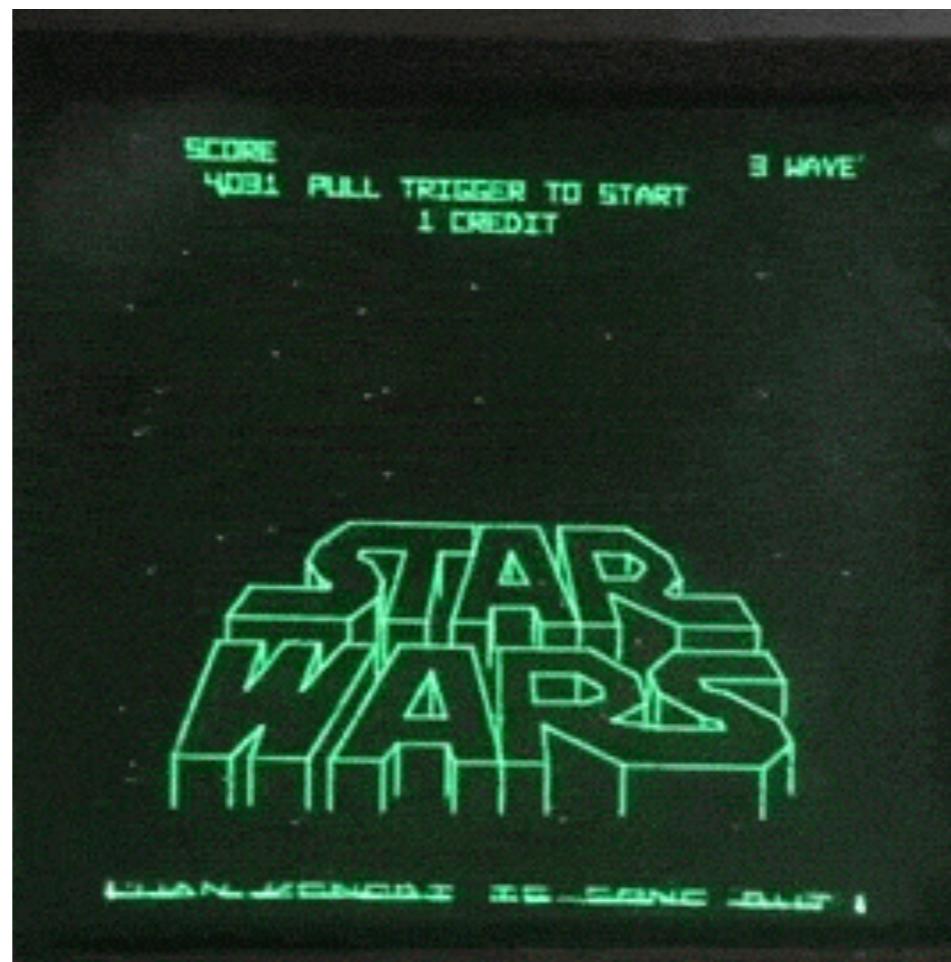
← Vektor

vs.

Raster



# Vektorbildschirm Eigenschaften



# Rastertechnik

- Dominanz der Rastertechnik in Vielzahl heutiger Technologien
- Beispiele:
  - Plotter vs. Drucker
  - Telex vs. Telefax
  - Rasterbildschirm vs. Vektorbildschirm
  - ...

# Rasterbildschirm

- **Rasterbildschirm** ist diskretes Display mit rasterförmiger Anordnung von Bildpunkten (engl. *Pixel*), denen jeweils Farbe zugeordnet ist
- **Rasterbildschirm** ist einfacher und generischer zu verwenden als Vektor-Display, z.B. im Bereich der Digitalfotografie, Satellitenbilder...

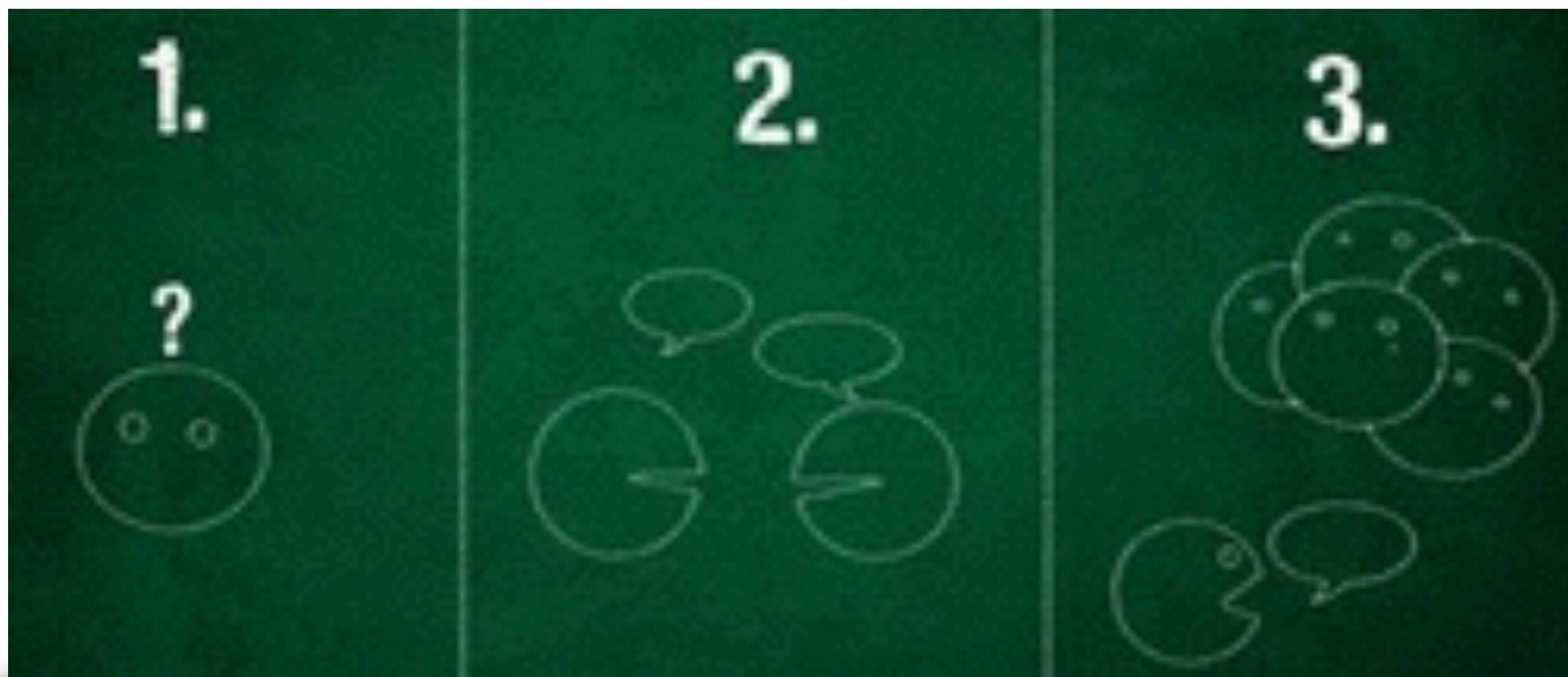
# Digitalisierung

- **Pixel** eines digitalen Rasterbildes besteht aus Helligkeit und Farbinformation
  - einfachster Fall: 1-Bit, d.h. schwarz oder weiß
- Farbbilder benötigen offensichtlich mehr Speicher (*siehe VL ICG - Farbmodelle*)

# Diskretisierung

- Überlagerung eines Rasters über kontinuierliche Bilder
- horizontale und vertikale Anzahl der Pixel bestimmt Auflösung
- **Dichte der Pixel** oder **Punktdichte** wird in *dots per inch (dpi)* ( $1 \text{ in} = 2.54 \text{ cm}$ ) gemessen
  - typische Computer-Displays haben mind. 100 dpi bis 200 dpi

# Think! Pair! Share!



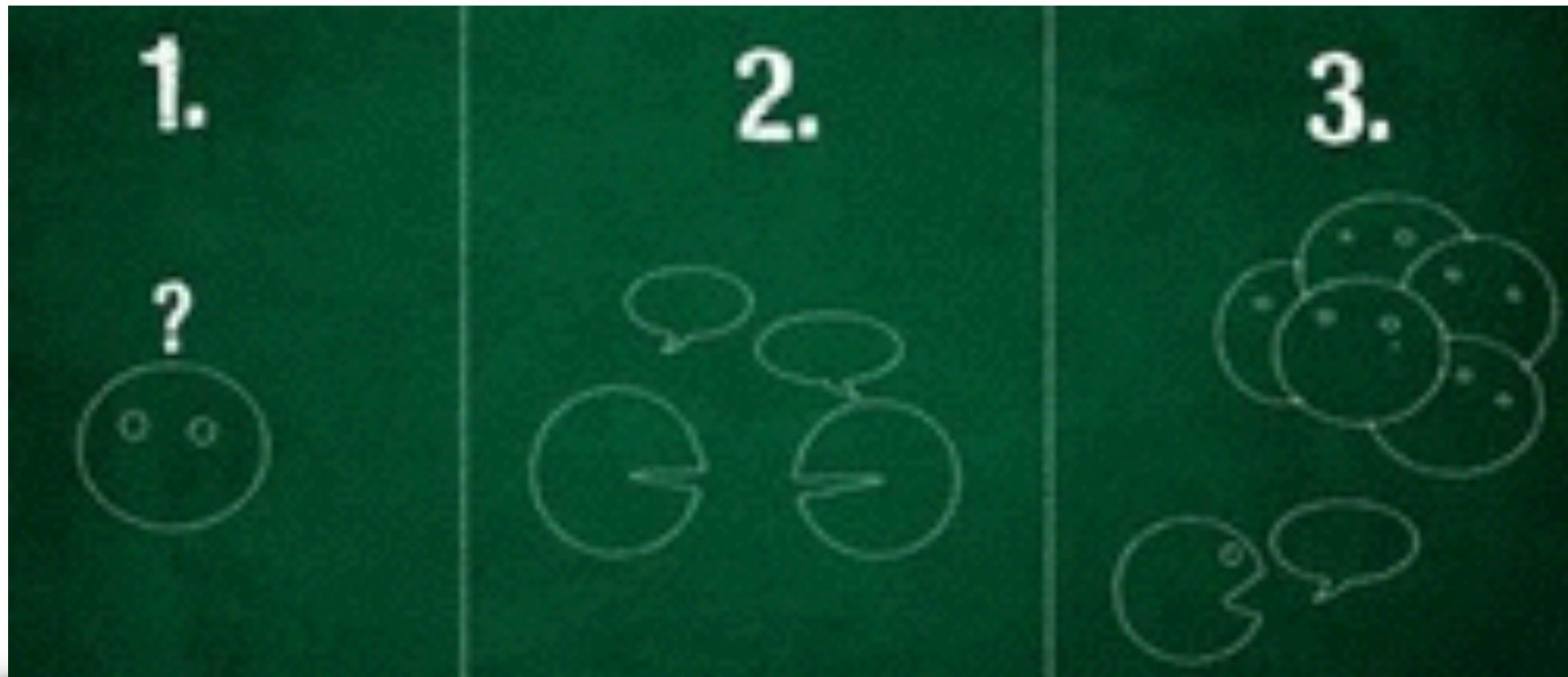
Welche Punktdichte hat folgendes Display?

# Pixelauflösung



2560 × 1600 MacBook Pro mit Retina Display (13")

# Think! Pair! Share!



Welche Pixelauflösung muss quadratisches Bild haben, welches 300 dpi auf 10 cm erfüllen soll?

# Rasterisierung

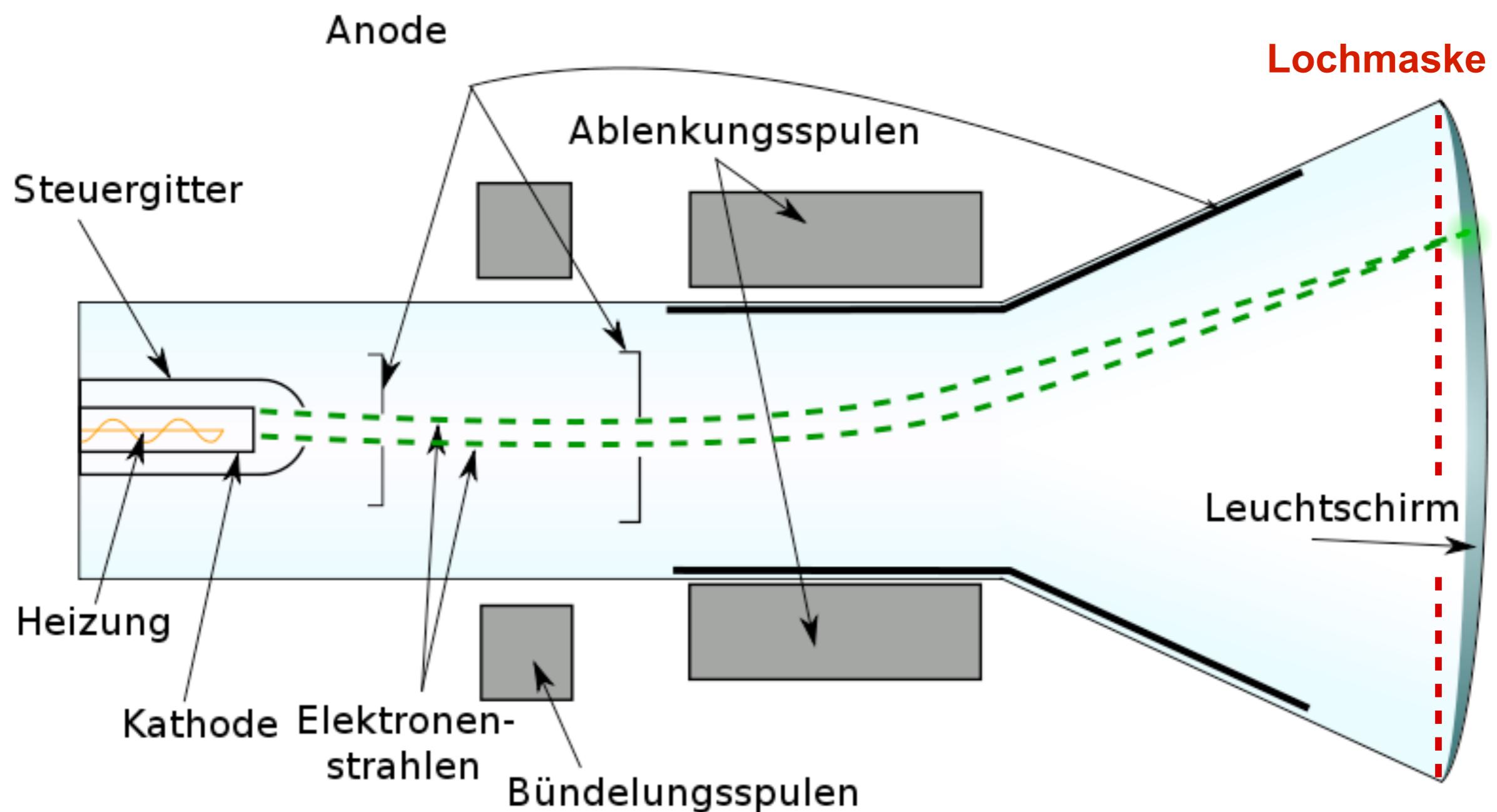
- Algorithmen zur **Rasterisierung**  
kontinuierlicher Daten notwendig
- Beispiele:
  - geometrische Objekte (Linien, Kurven)
  - Schriftdarstellung
  - allgemeine Flächen (Dreiecke)
  - ...

# Rastertechnik

## Herausforderungen

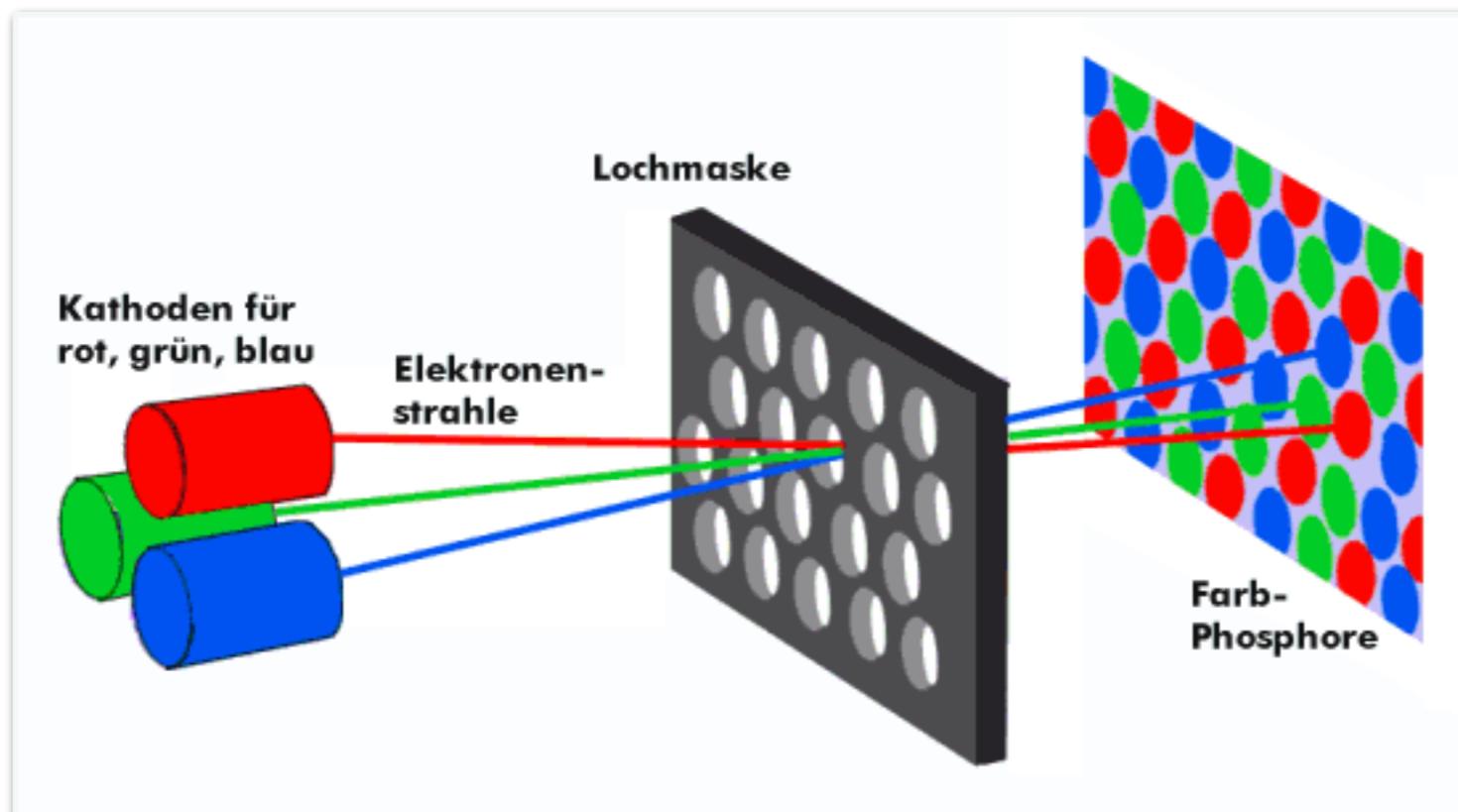
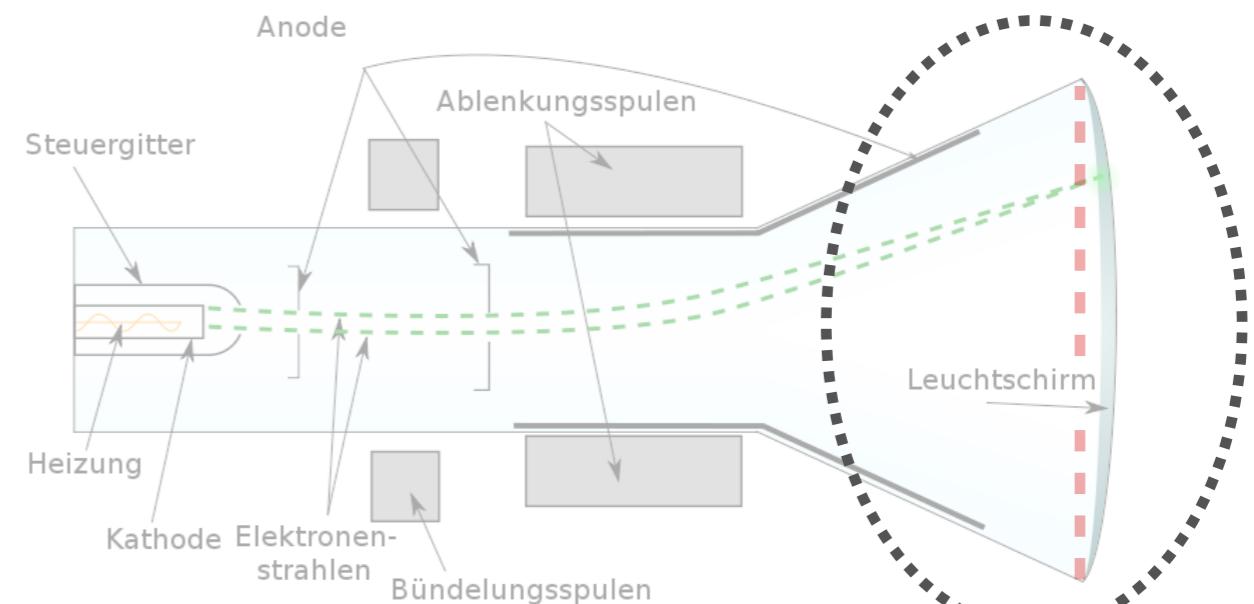
- Effizienz
- Aliasing
  - Treppenbildung
  - Moiré-Muster
- Rauschen
- ...

# CRT Display



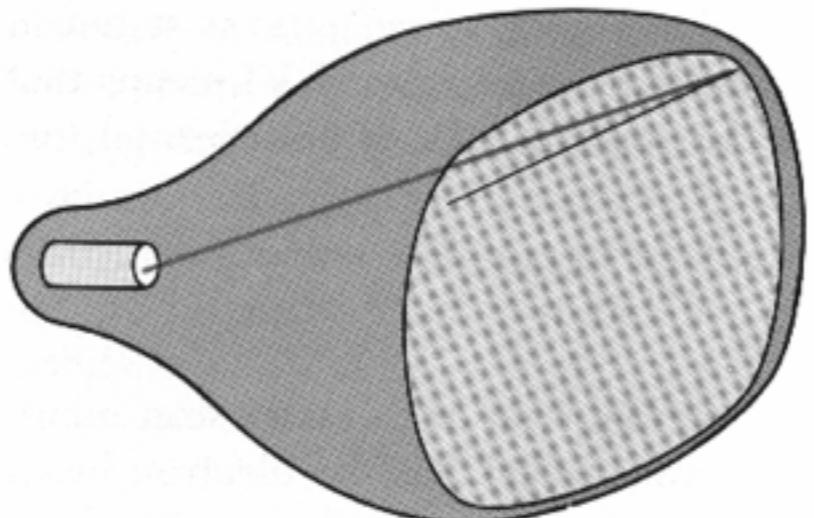
# CRT Display

## Lochmaske

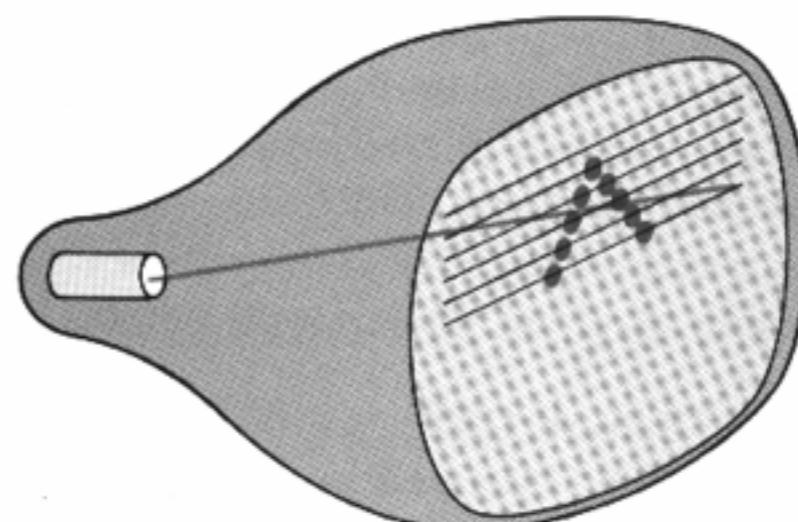


# CRT Display

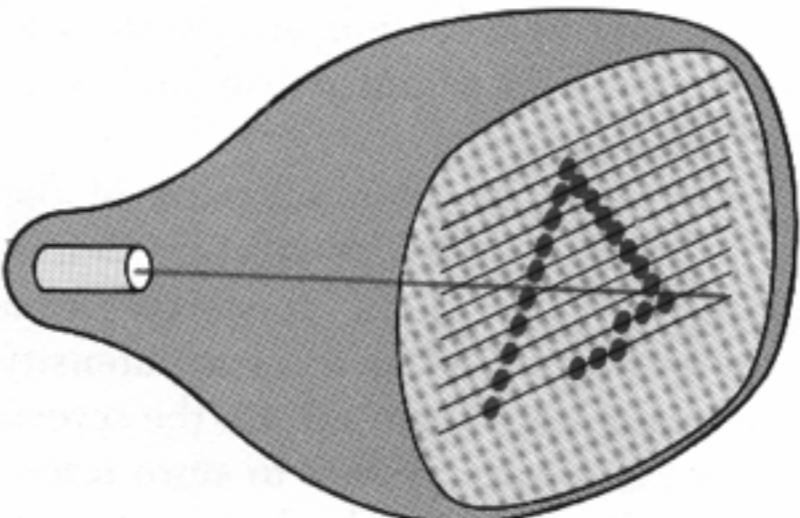
## Pixel-Traversierung



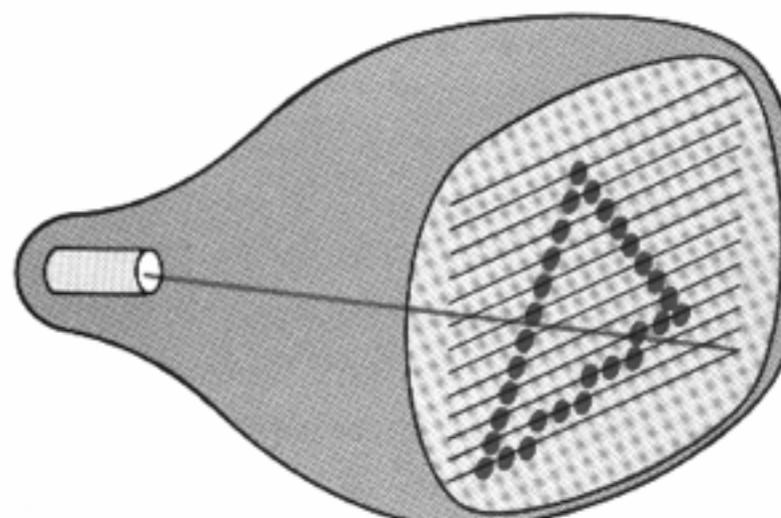
(a)



(b)



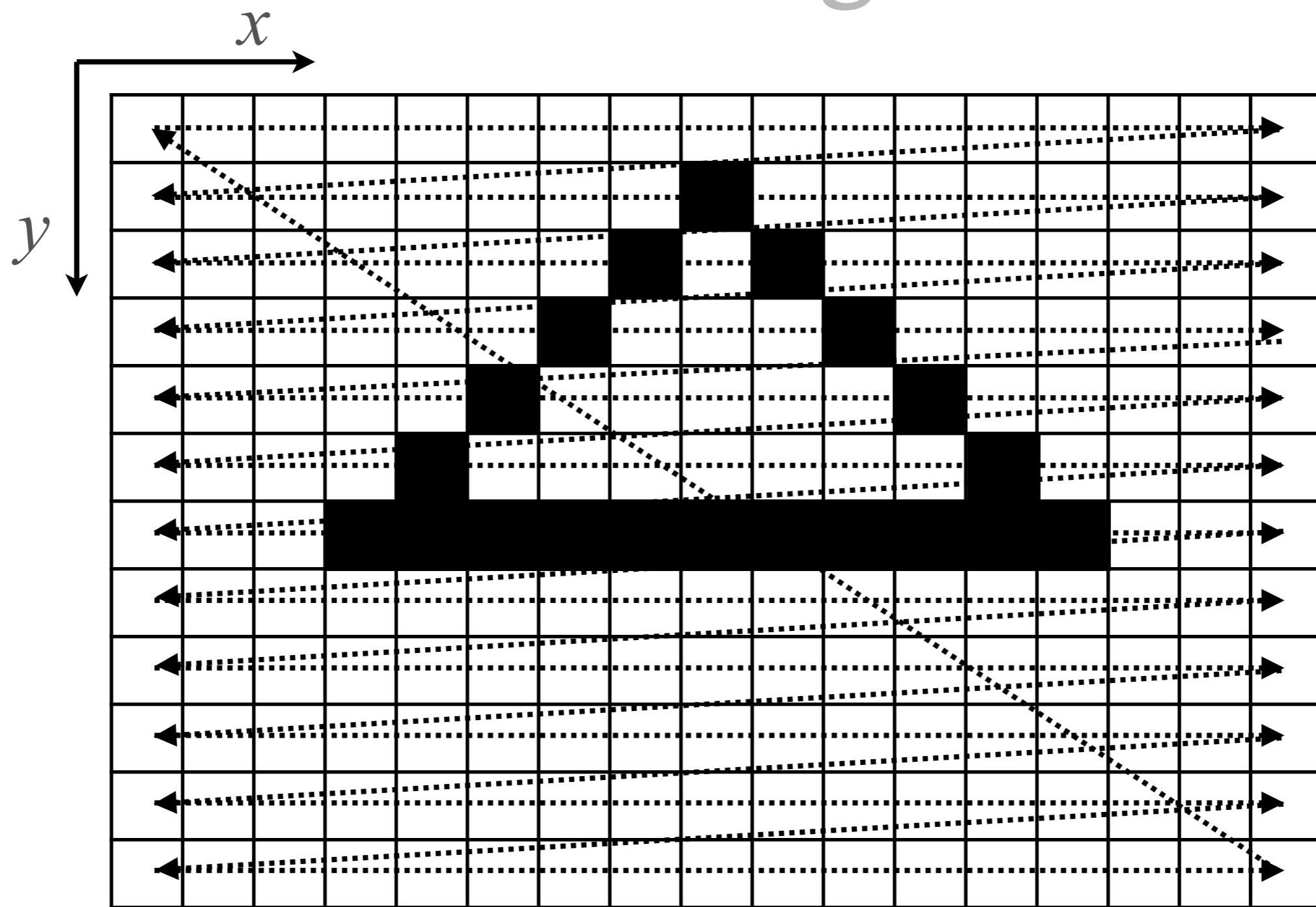
(c)



(d)

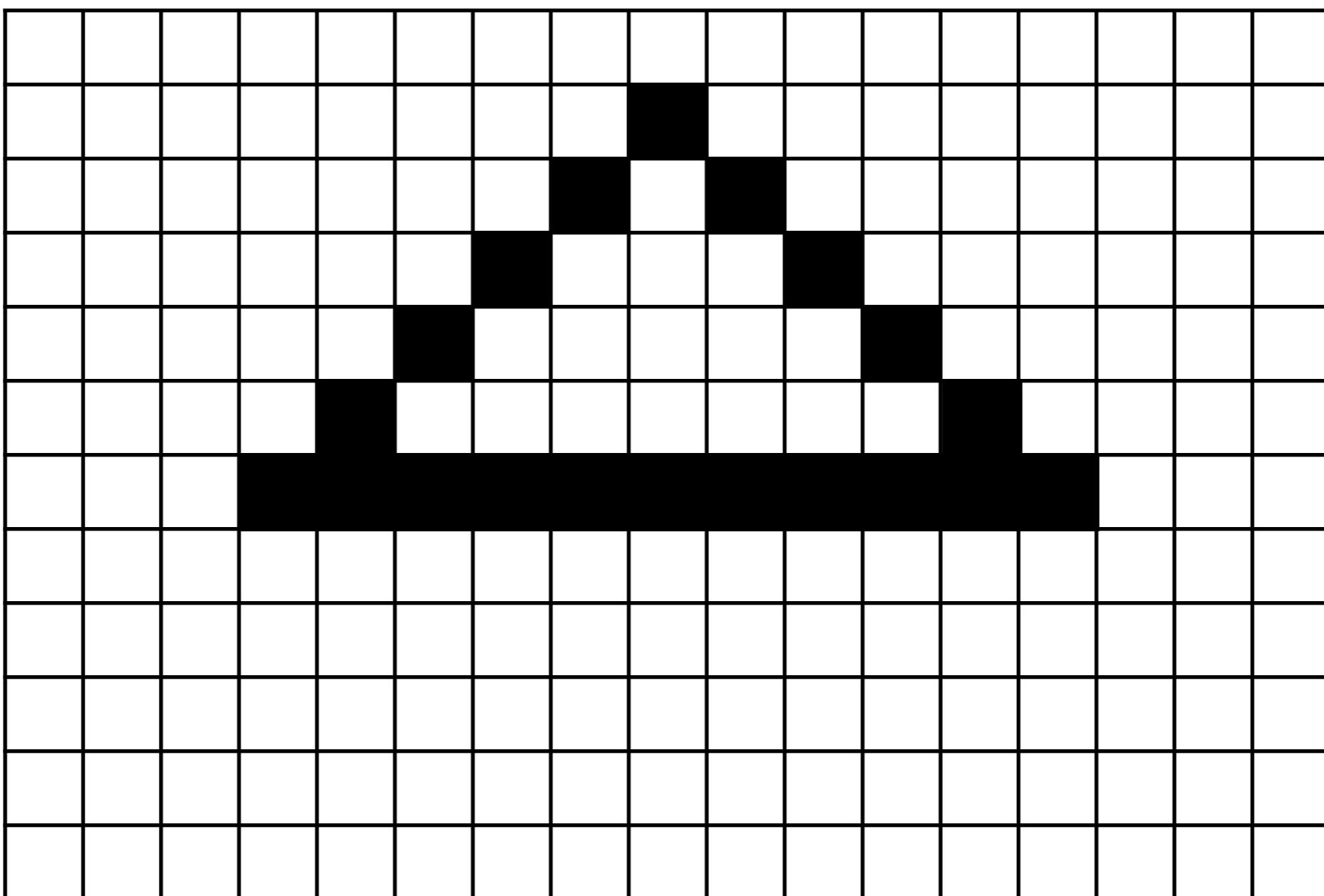
# CRT Display

## Pixel-Traversierung



# CRT Display

## Pixel-Traversierung

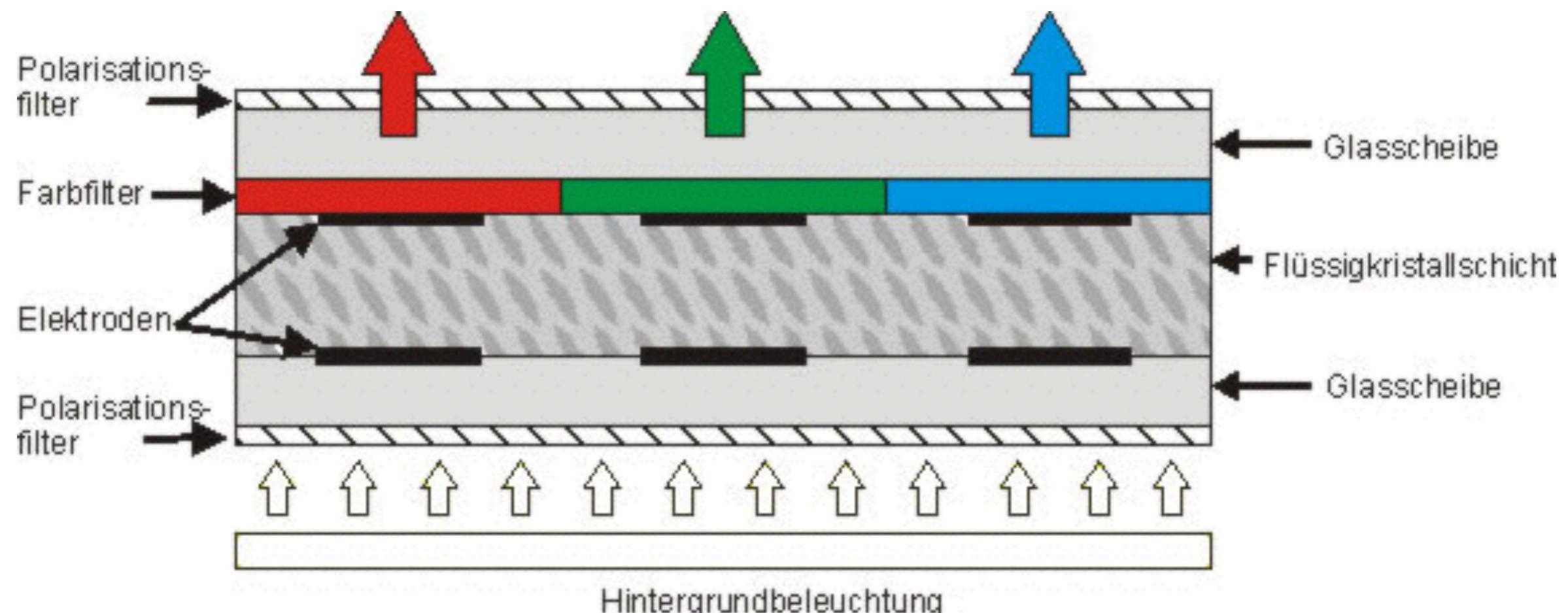


# Flüssigkristallanzeige

- Flüssigkristallanzeige (engl. *Liquid Crystal Display, LCD*) ist Bildschirm, dessen Funktion darauf beruht, dass Flüssigkristalle Polarisationsrichtung von Licht beeinflussen, wenn bestimmtes Maß an elektrischer Spannung angelegt wird

# Flüssigkristallanzeige

## Funktionsweise

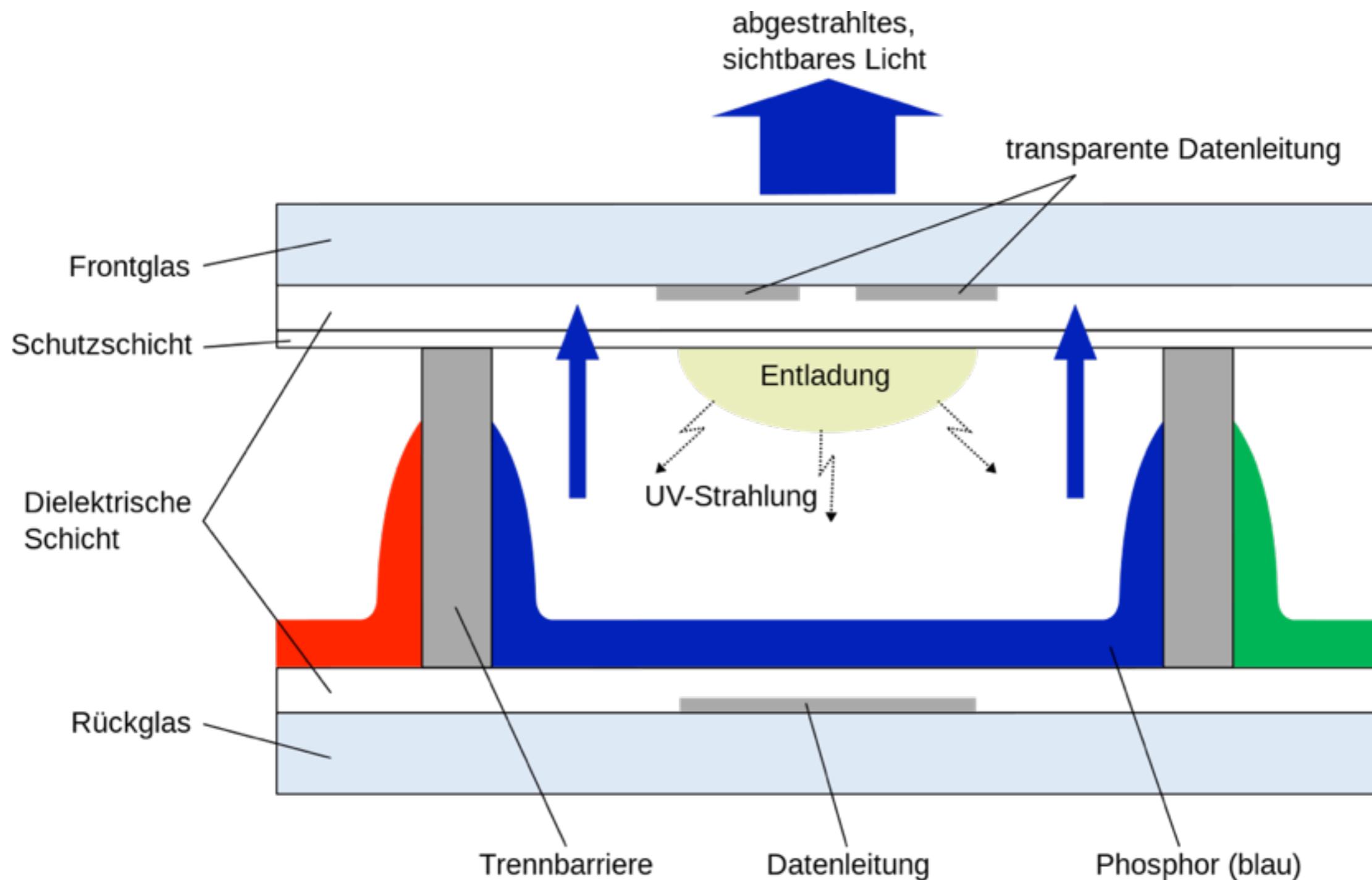


# Plasmabildschirm

- Plasmabildschirm (engl. *Plasma Display Panel*) ist Farbflachbildschirm, der verschiedenfarbiges Licht mit Hilfe von Leuchtstoffen erzeugt, die durch von Gasentladungen erzeugtes Plasma angeregt werden

# Plasmabildschirm

## Funktionsweise

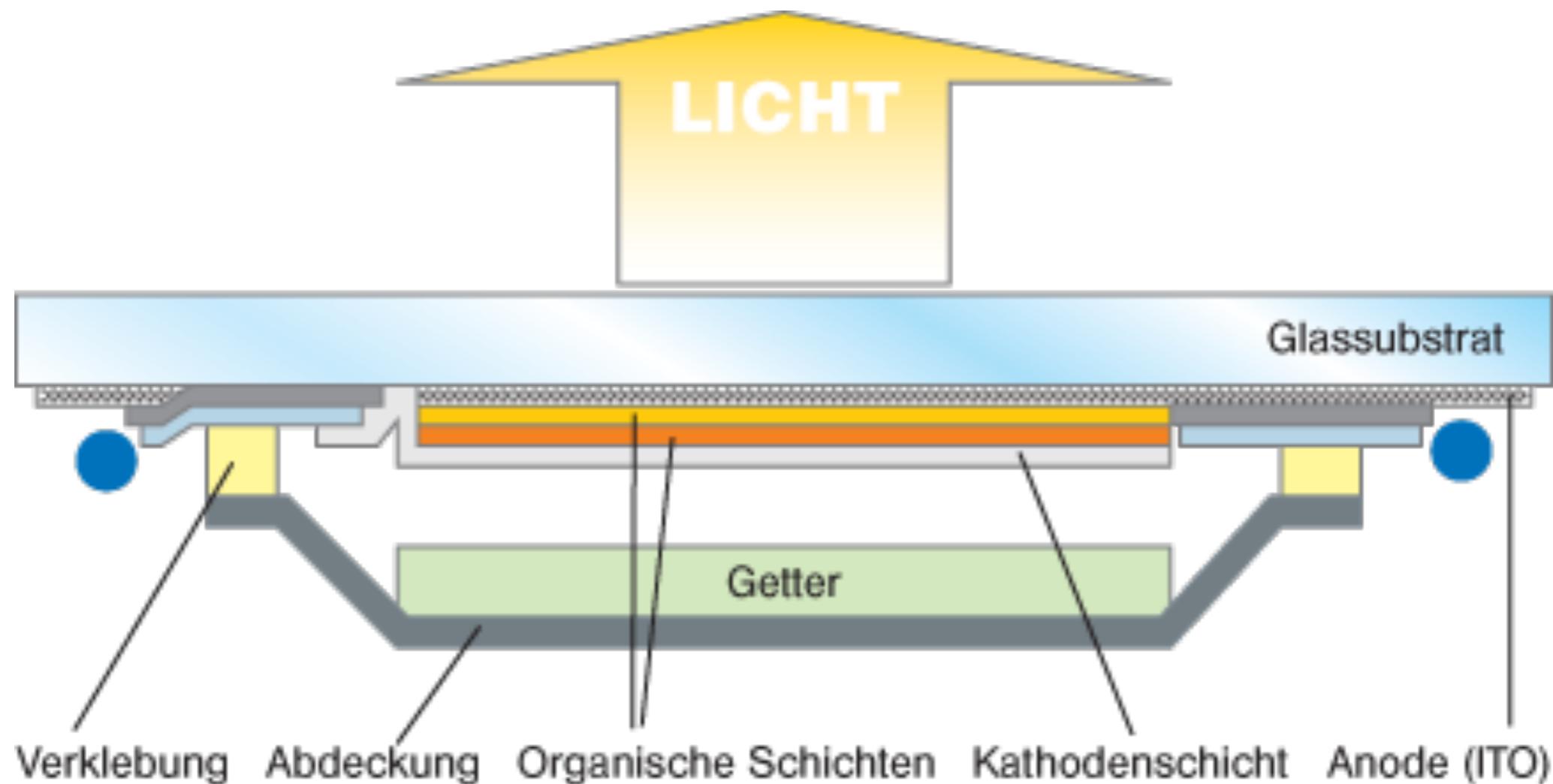


# OLED

- Organische Leuchtdiode (engl. *Organic Light Emitting Diode, OLED*) ist leuchtendes Dünnschichtbauelement aus organischen halbleitenden Materialien

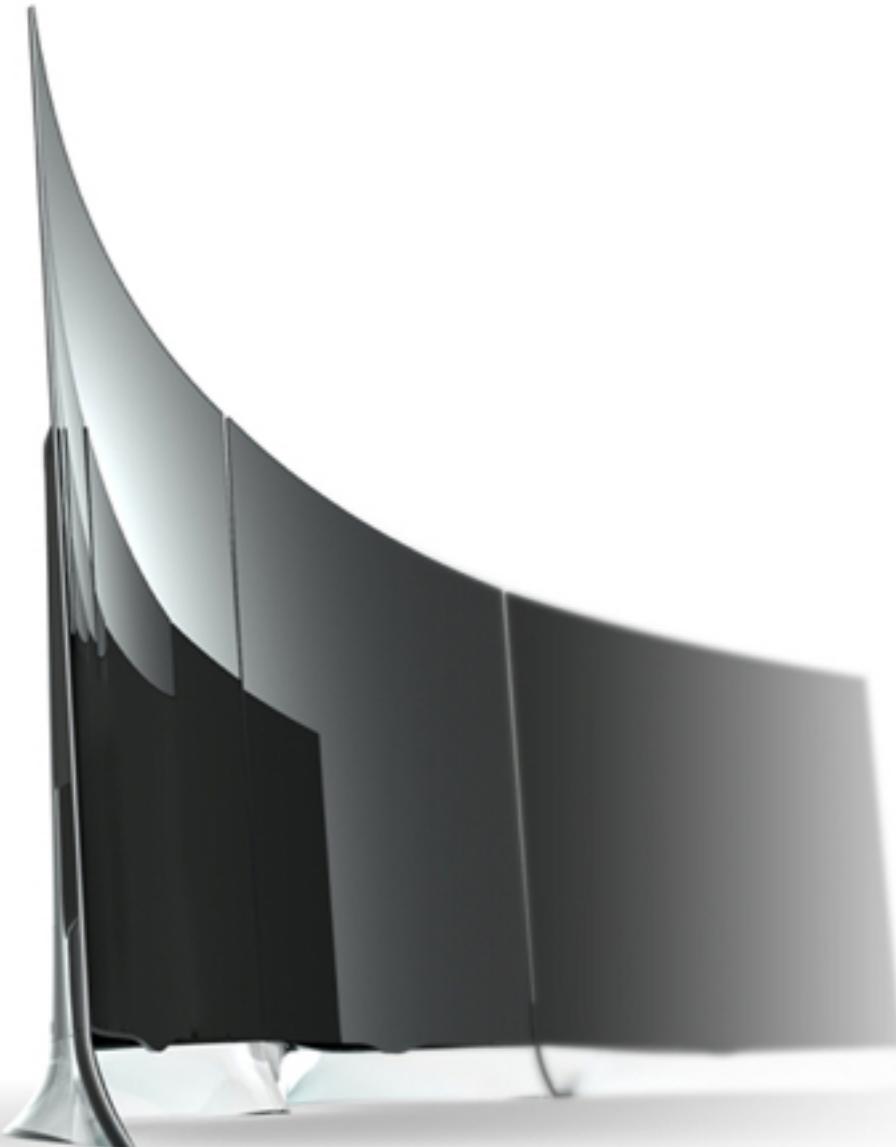
# OLED

## Funktionsweise



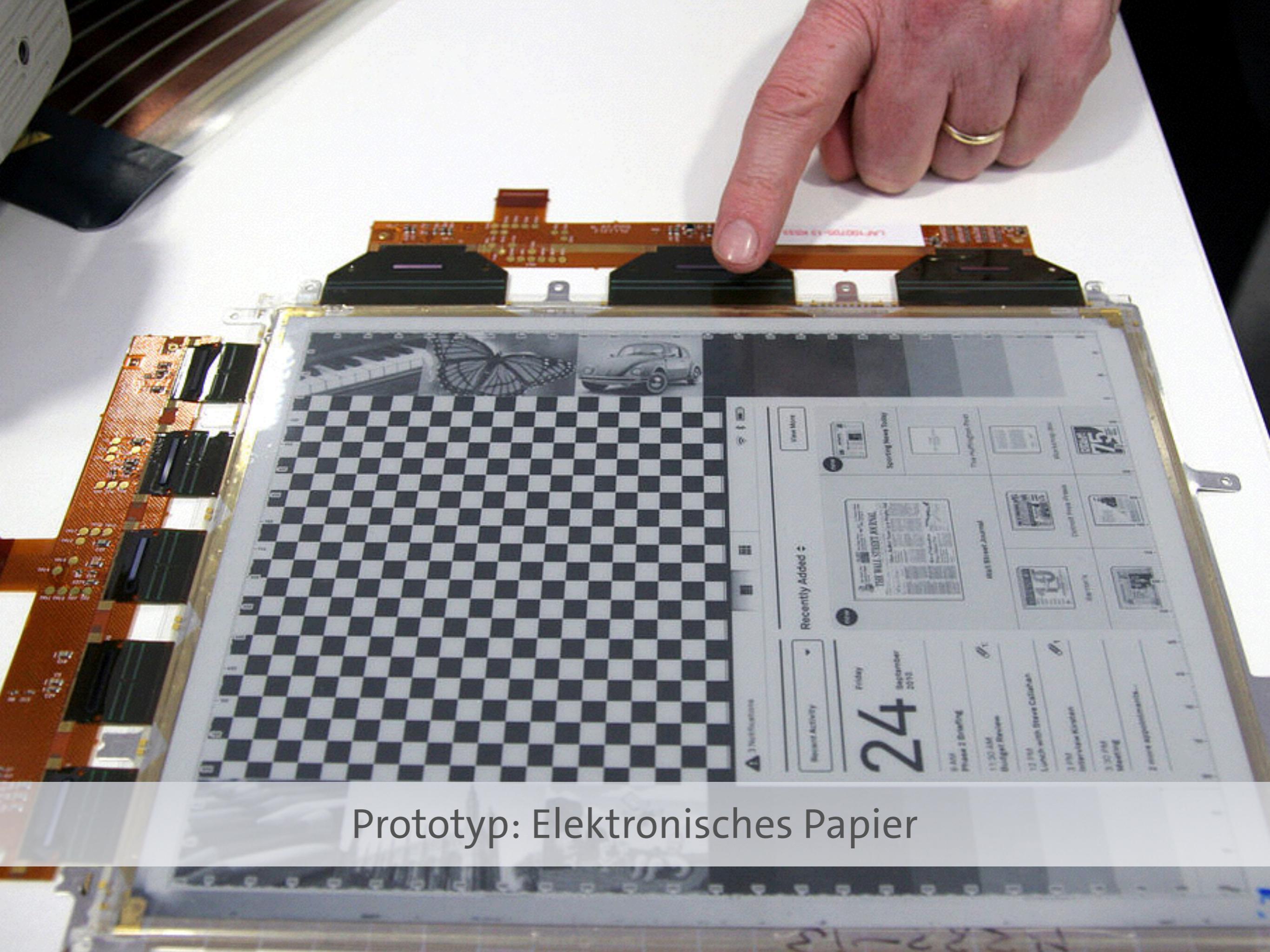
# OLED

## Beispiele



# Elektronisches Papier

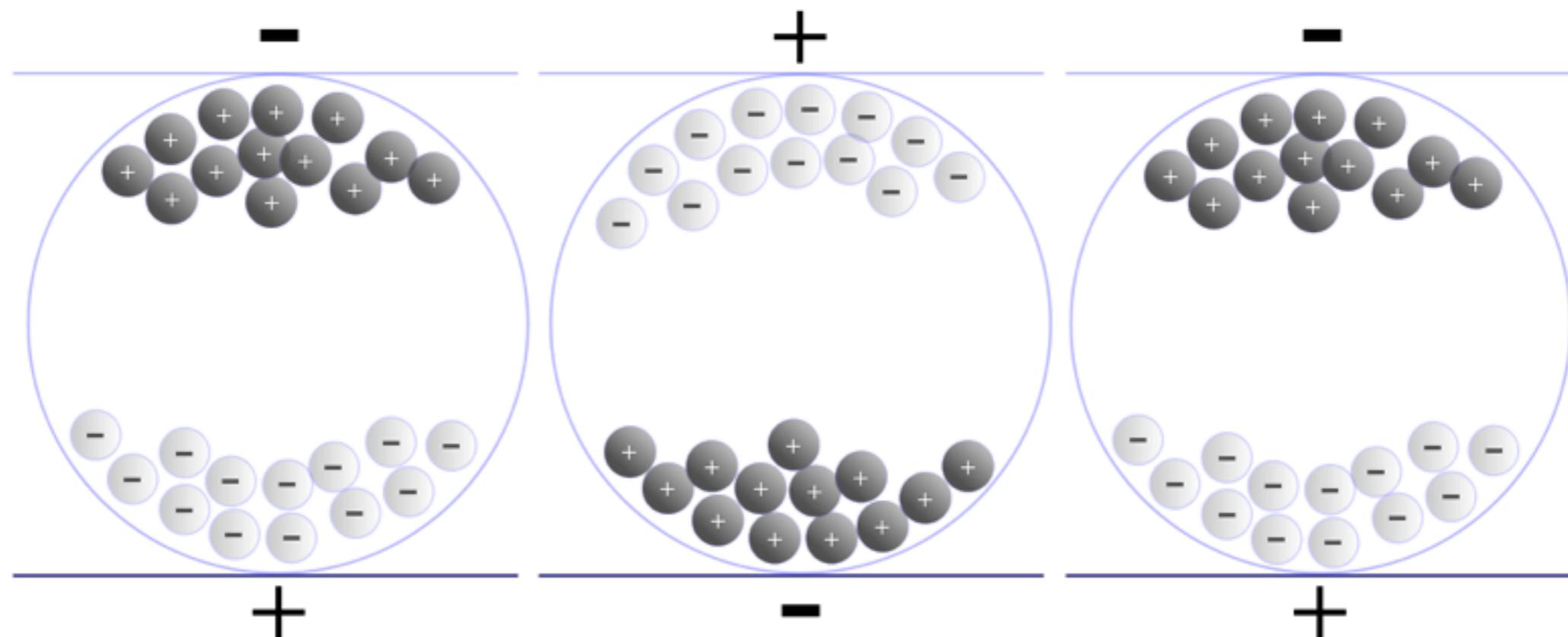
- **Elektronisches Papier** (engl. (kurz) *E-Paper*) sind Displays, die Aussehen von Tinte bzw. Farbe auf Papier nachbilden
- **E-Paper-Displays** reflektieren wie normales Papier Licht

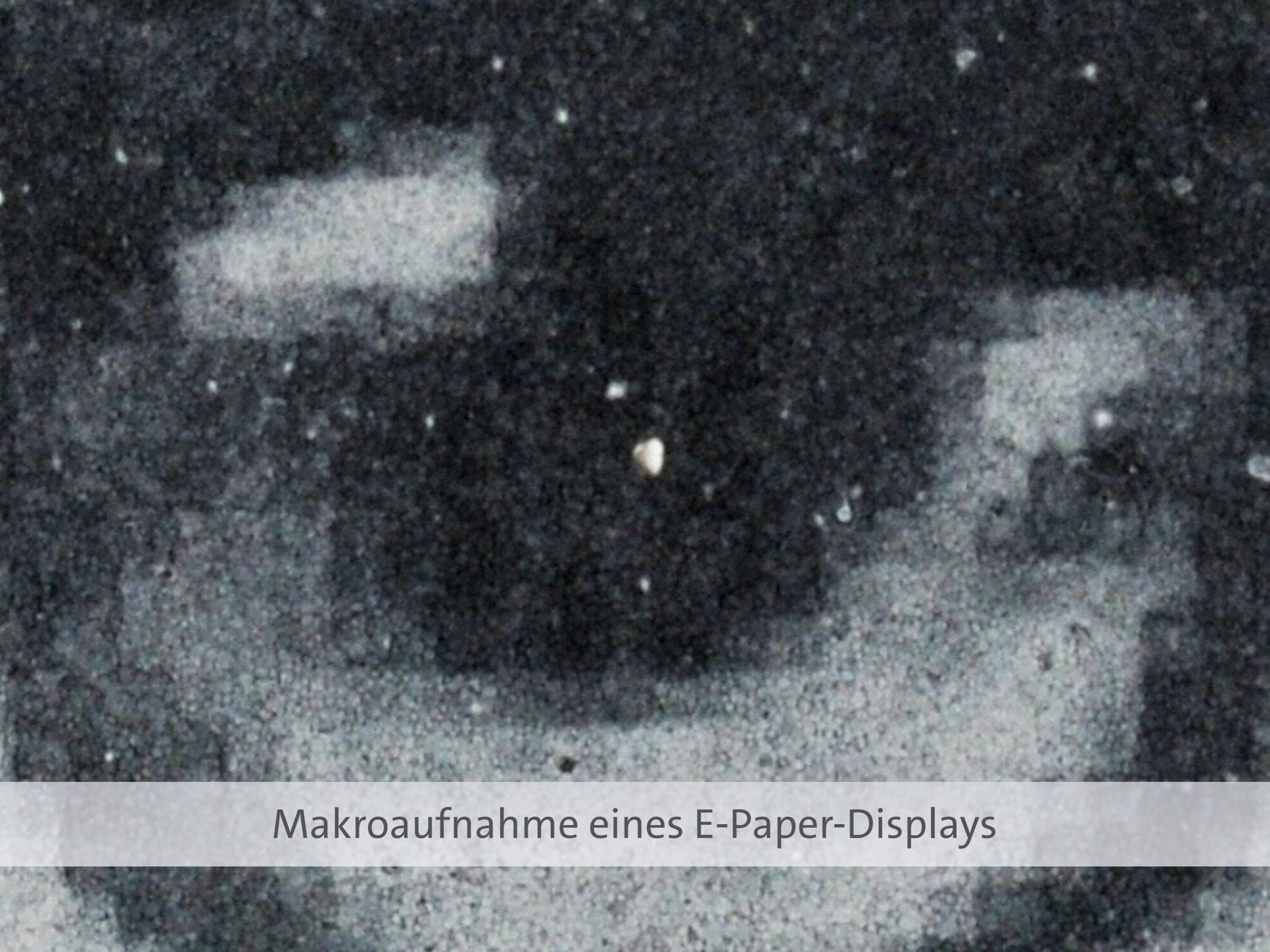


Prototyp: Elektronisches Papier

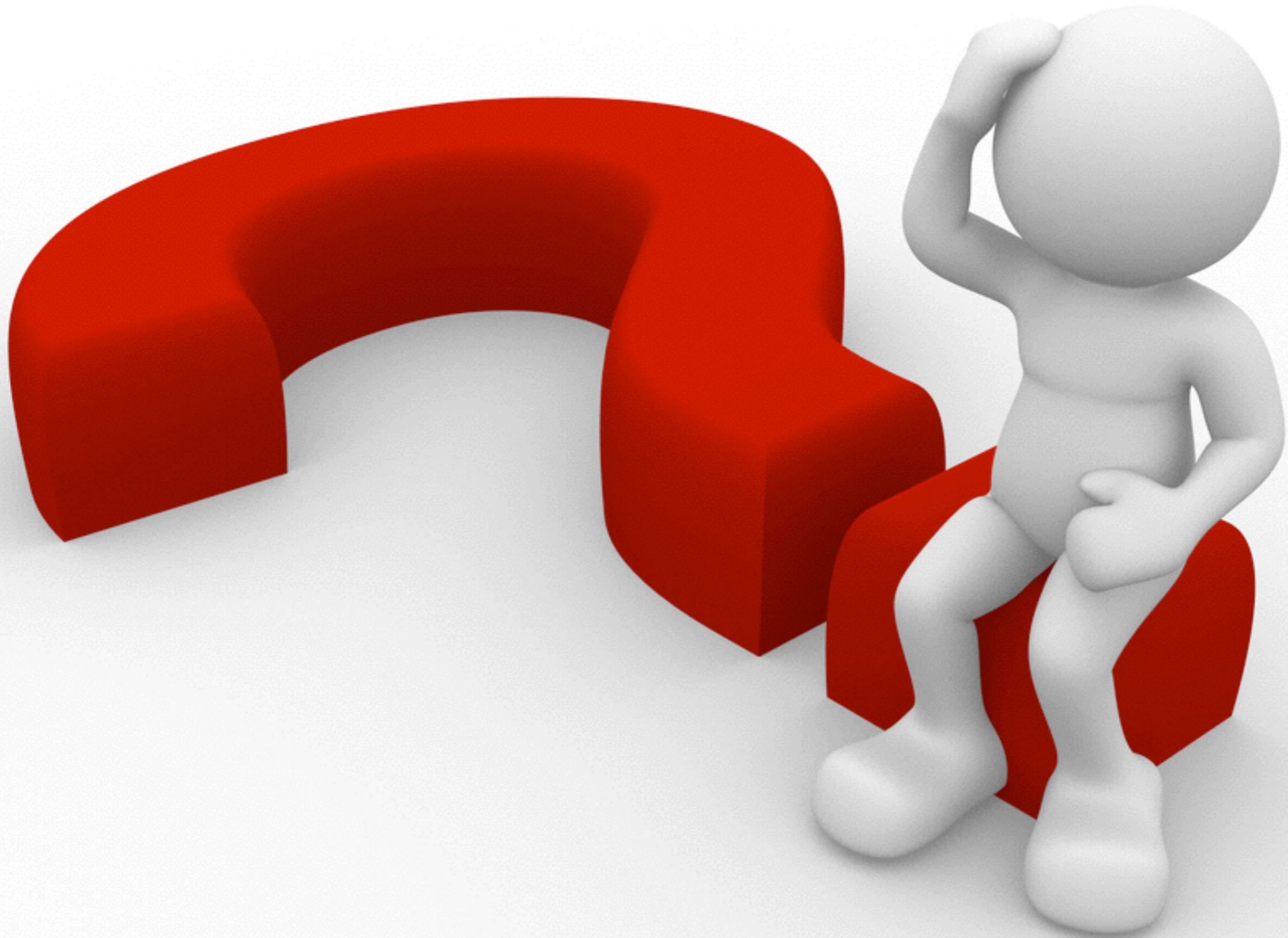
# Elektronisches Papier

## Funktionsweise





Makroaufnahme eines E-Paper-Displays





# Interaktive Medien

## Kapitel Bilder

### Farbmodelle

# Digitale Farben

## Begriffe

- **Farbmodelle** beschreiben mit Formeln und geometrischen Modellen, wie menschliches Auge Farben sieht und Farben zueinander in Beziehung stehen
- **Farbraum** ist Bereich der Farben, die mit Farbmodell erzeugt werden können (**Gamut**)
- **Gerätefarbraum** beschreibt Gamut, die physikalisches Gerät darstellen kann

# Farbmodell

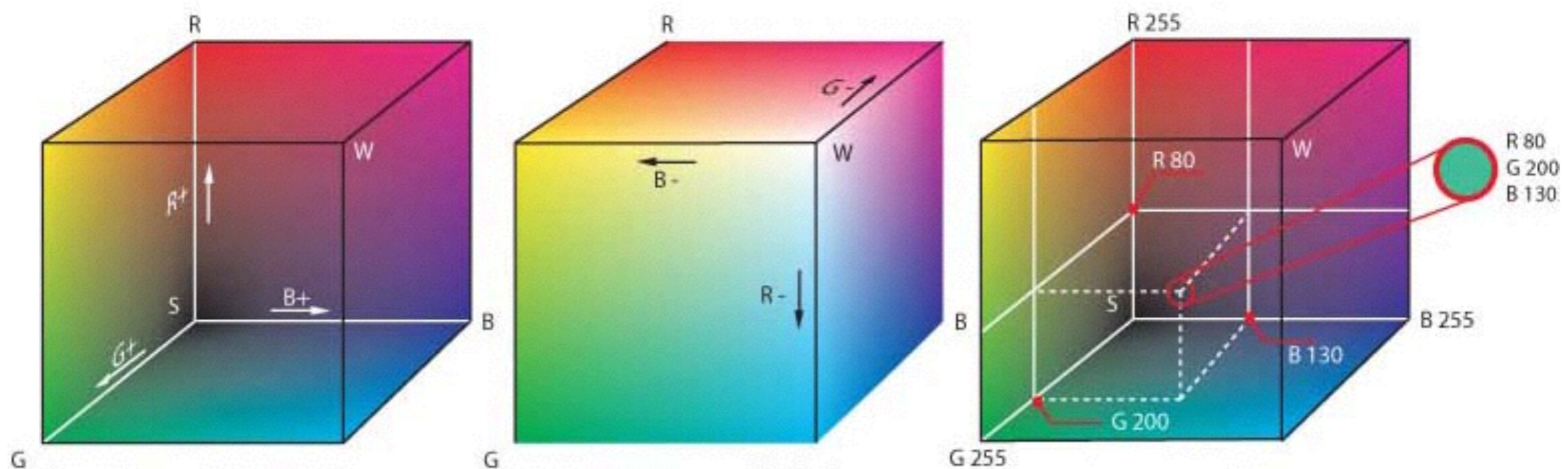
- Aufgabe:
  - Spezifikation von Farben
- Aufbau allgemein:
  - 3D-Farbkoordinatensystem
  - Farbe spezifiziert durch Koordinatentripel innerhalb des (endlich großen) Farbraums

# Farbaddition

- falls weder Stäbchen noch Zapfen durch Licht erregt werden, “sehen wir Schwarz”
  - werden alle Stäbchen und Zapfen erregt, “sehen wir Weiß”
- **additive Farbmischung:** Hinzufügen von Licht, um einzelne Farben (und schließlich weißes Licht) darzustellen

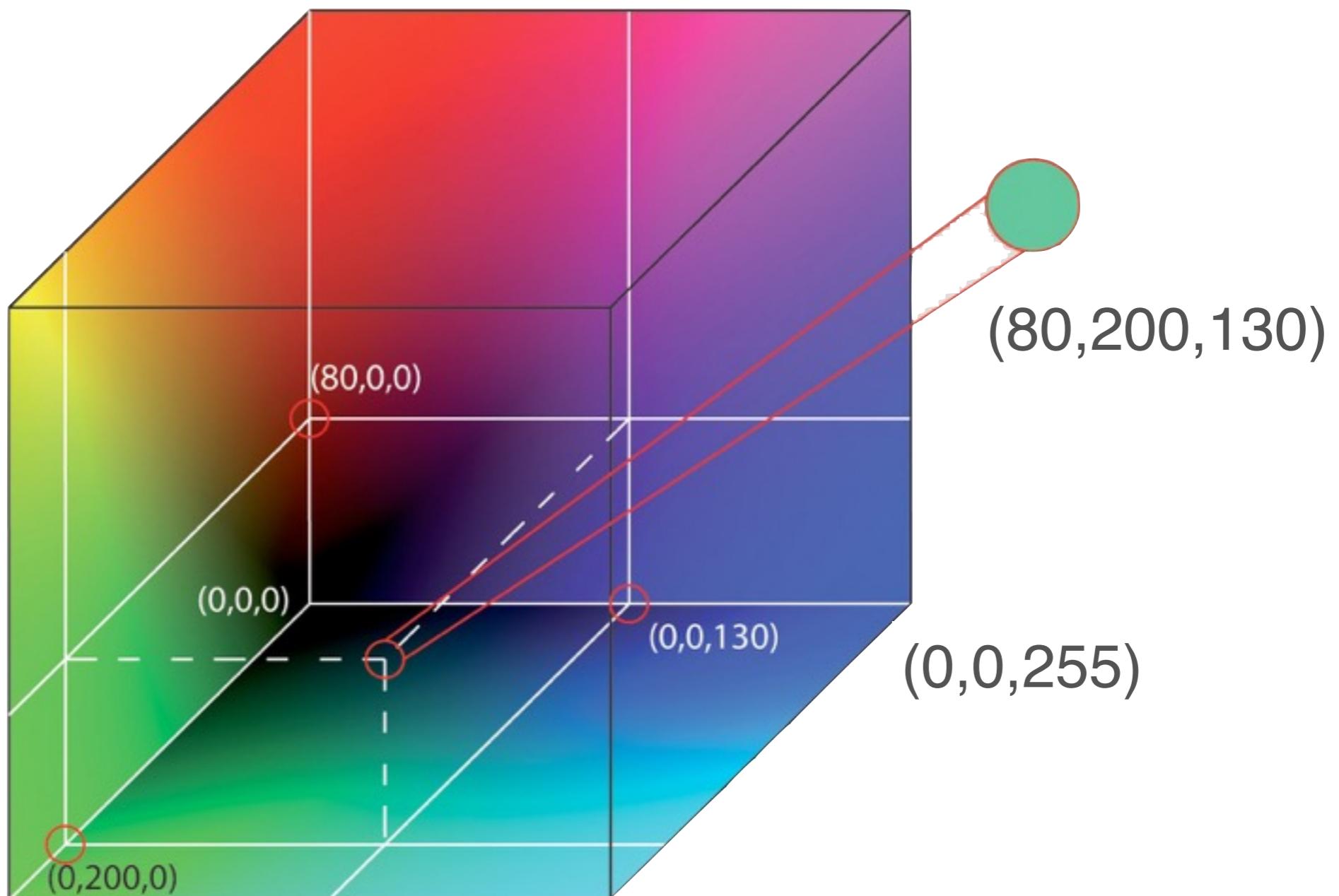
# RGB-Farbmodell

- RGB-Farbmodell verwendet drei Primärfarben “Rot”, “Grün” und “Blau”, um Farben additiv darzustellen



# RGB-Farbmödell

(255,0,0)



(0,255,0)

100

# RGB-Farbmodell

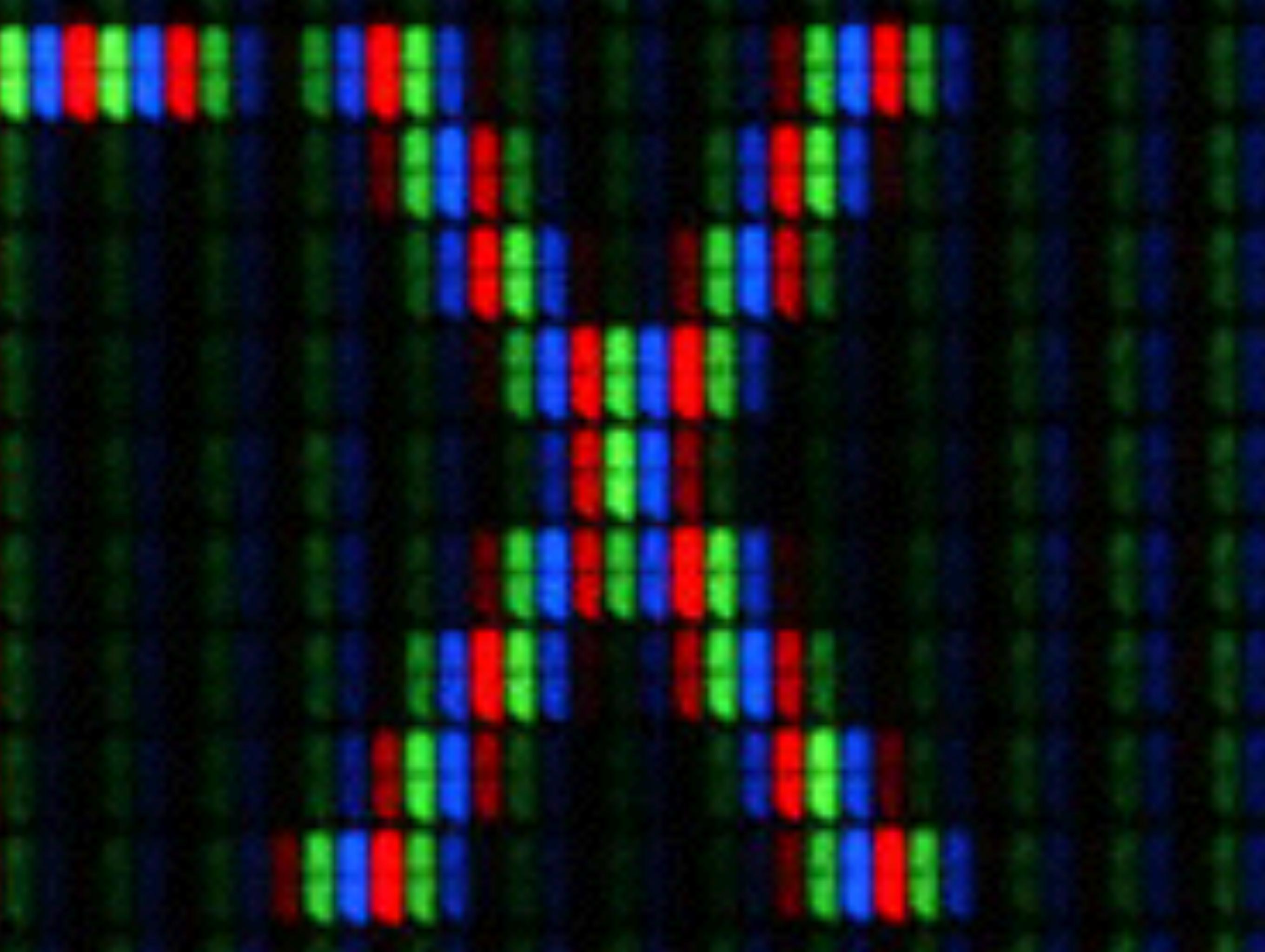
## Normalisierung

- Unabhängig von gewählter Quantisierung (mit maximalen Farbwert  $K_{max}$ ) lassen sich Farbkanäle mit Werten  $R$ ,  $G$  und  $B$  auf Bereich  $[0, 1]$  (oder in %) **normalisieren**

$$r = \frac{R}{K_{max}} \in [0, 1]$$

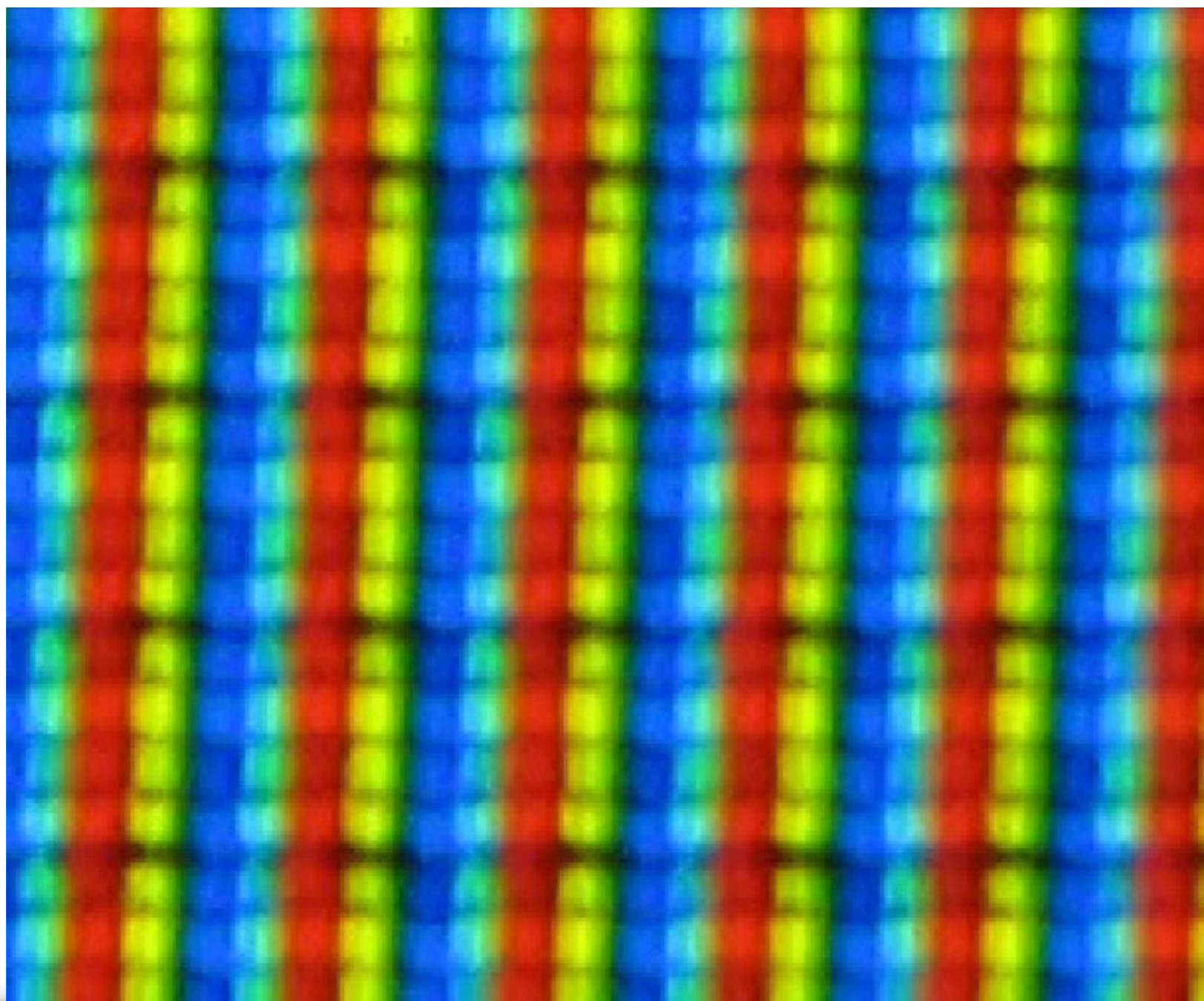
$$g = \frac{G}{K_{max}} \in [0, 1]$$

$$b = \frac{B}{K_{max}} \in [0, 1]$$



# Subpixel

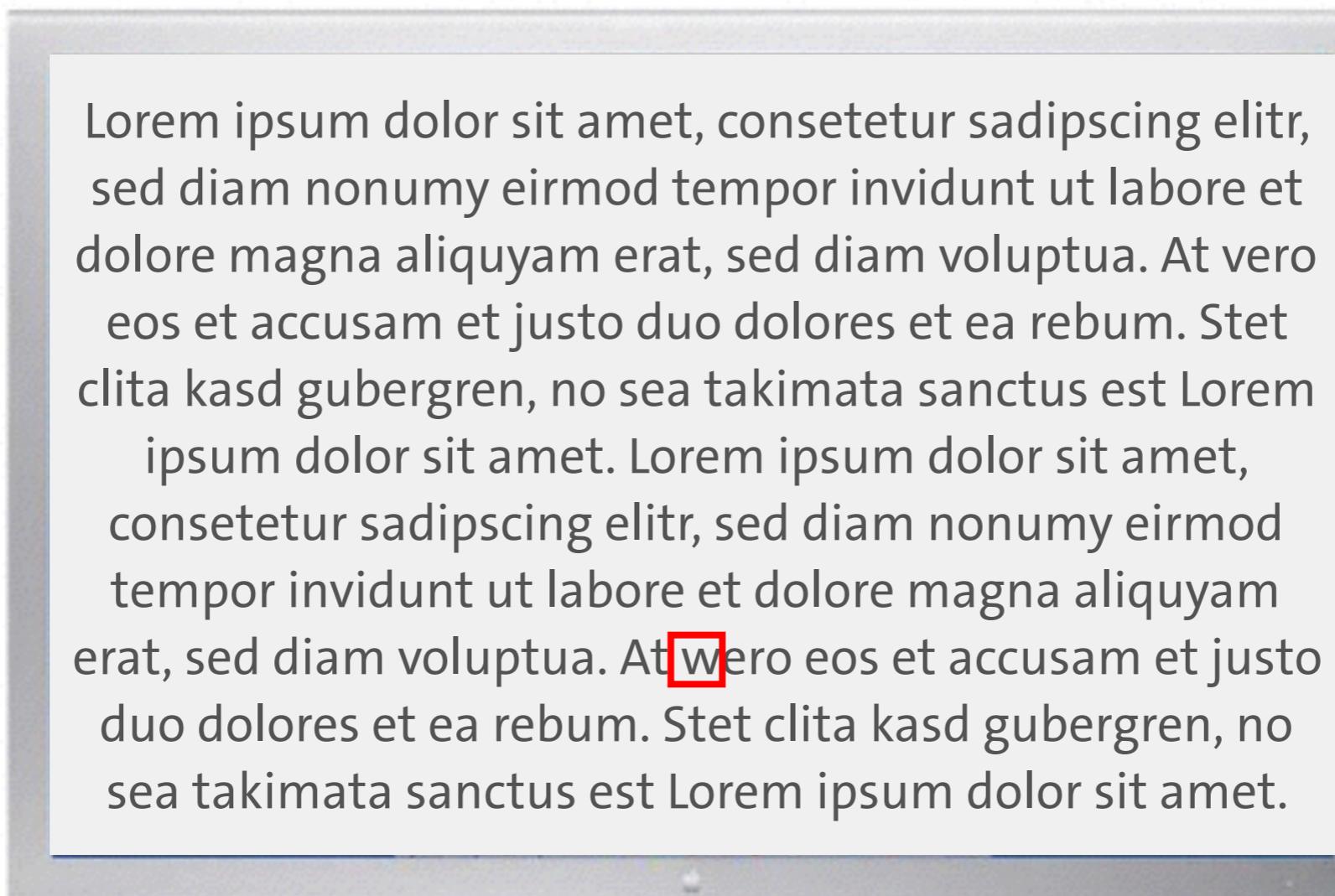
## Beispiel: Sharp LCD



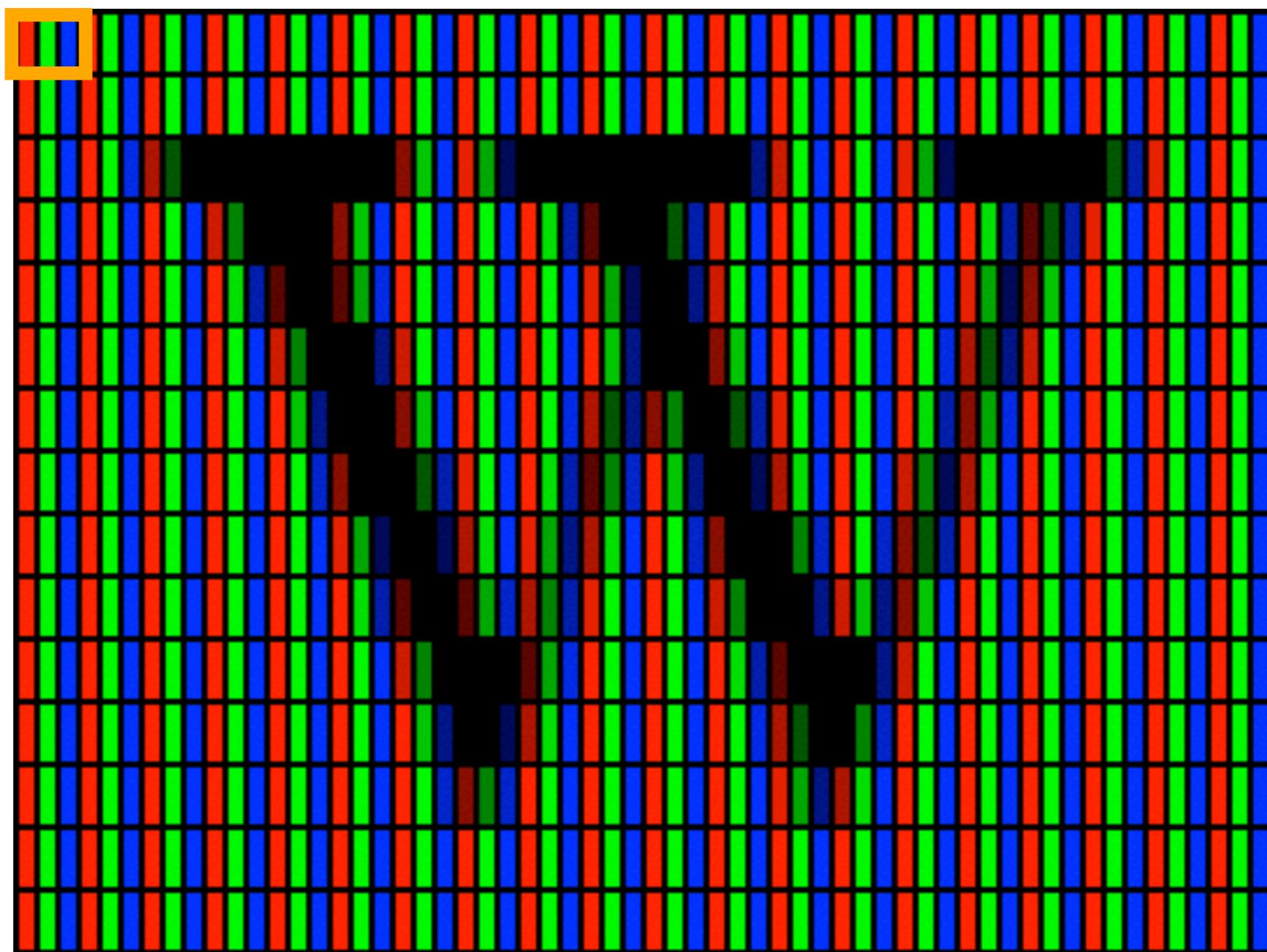
# Subpixel

- Subpixel bezeichnen (Teil-)Pixel aus denen sich bei Computermonitoren Grundfarben Rot, Grün und Blau zusammensetzen
- drei Subpixel Rot, Grün und Blau erzeugen Farbwahrnehmung durch additive Mischung

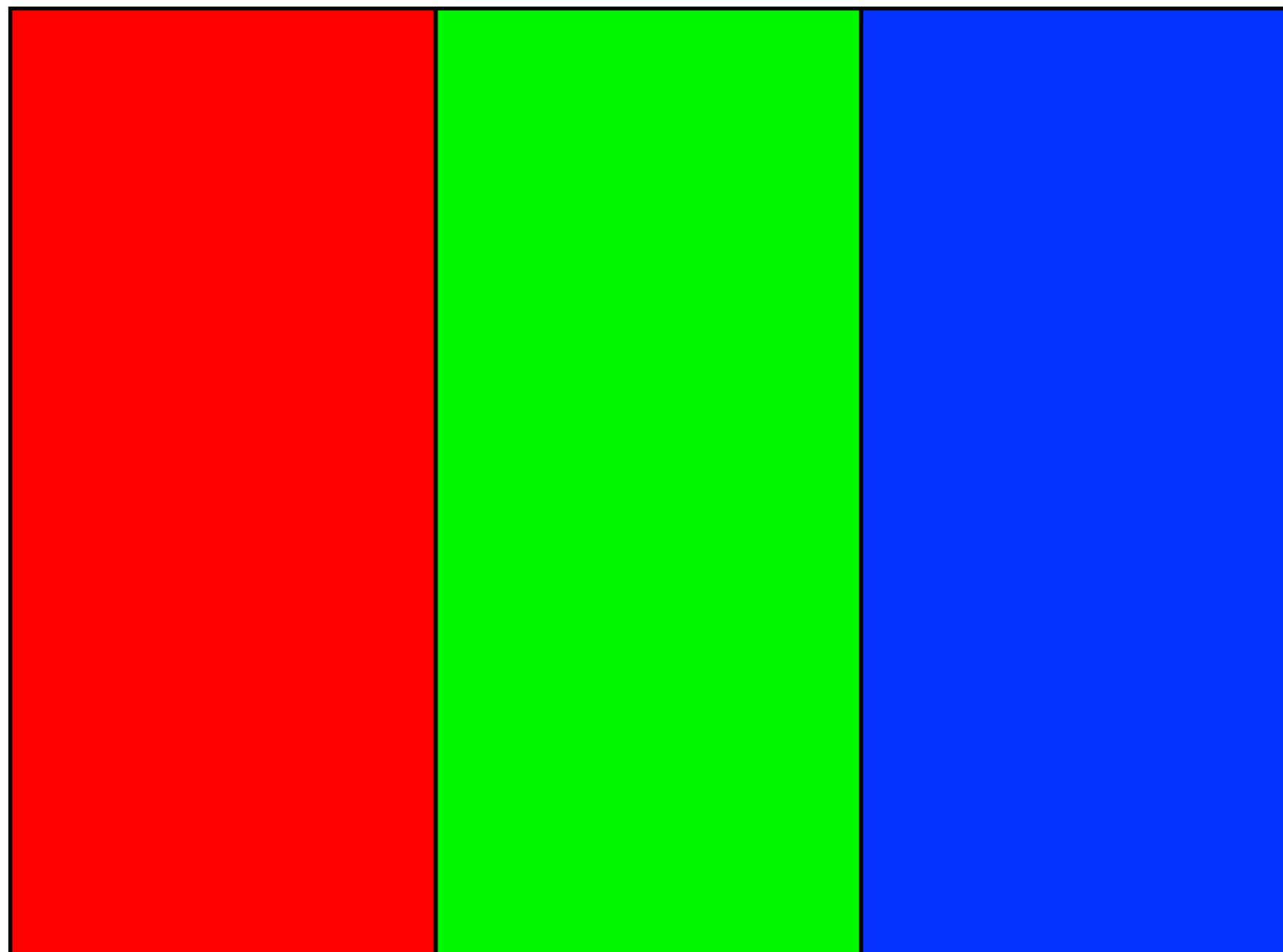
# Subpixel Beispiel



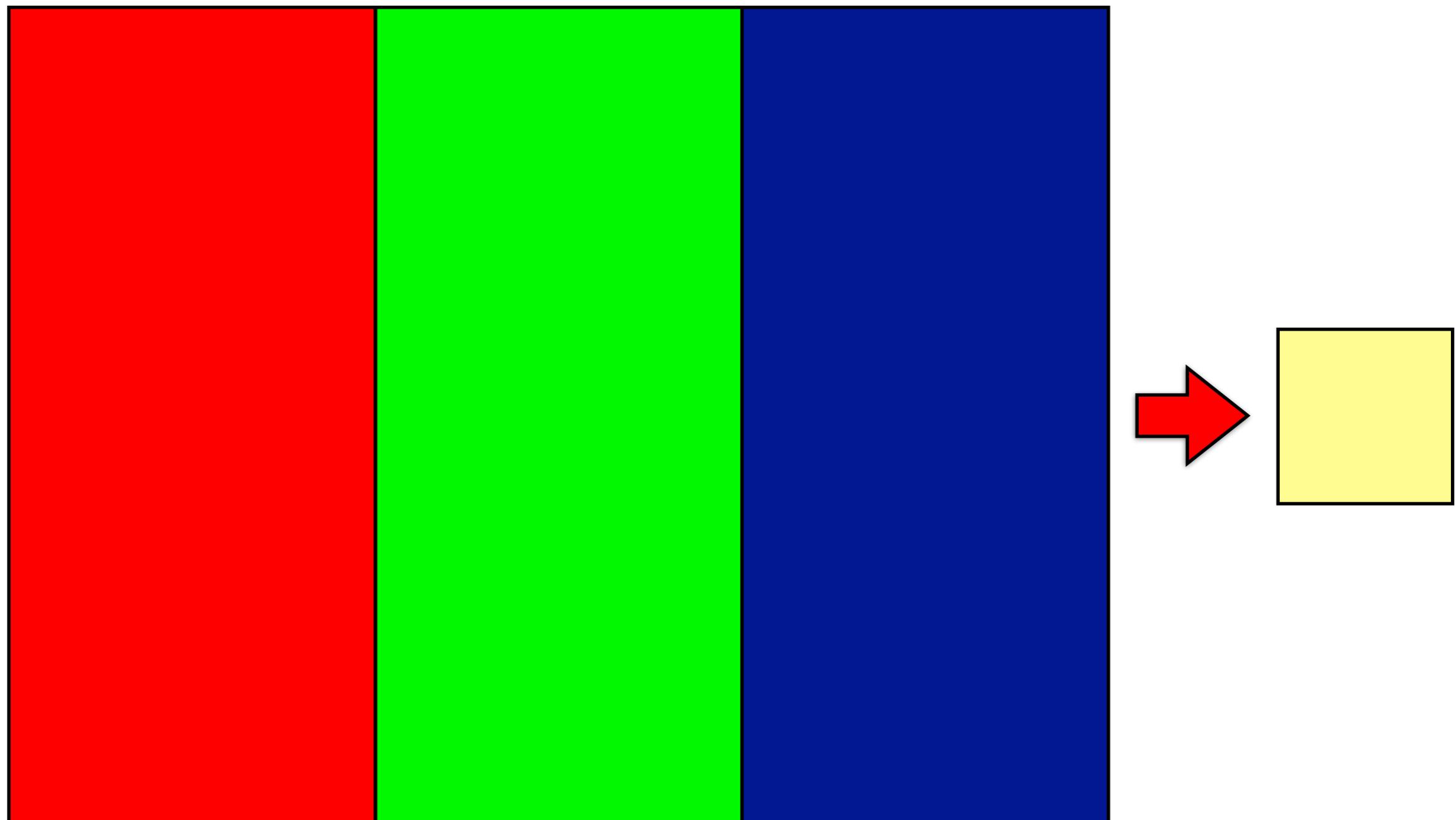
# Subpixel Beispiel



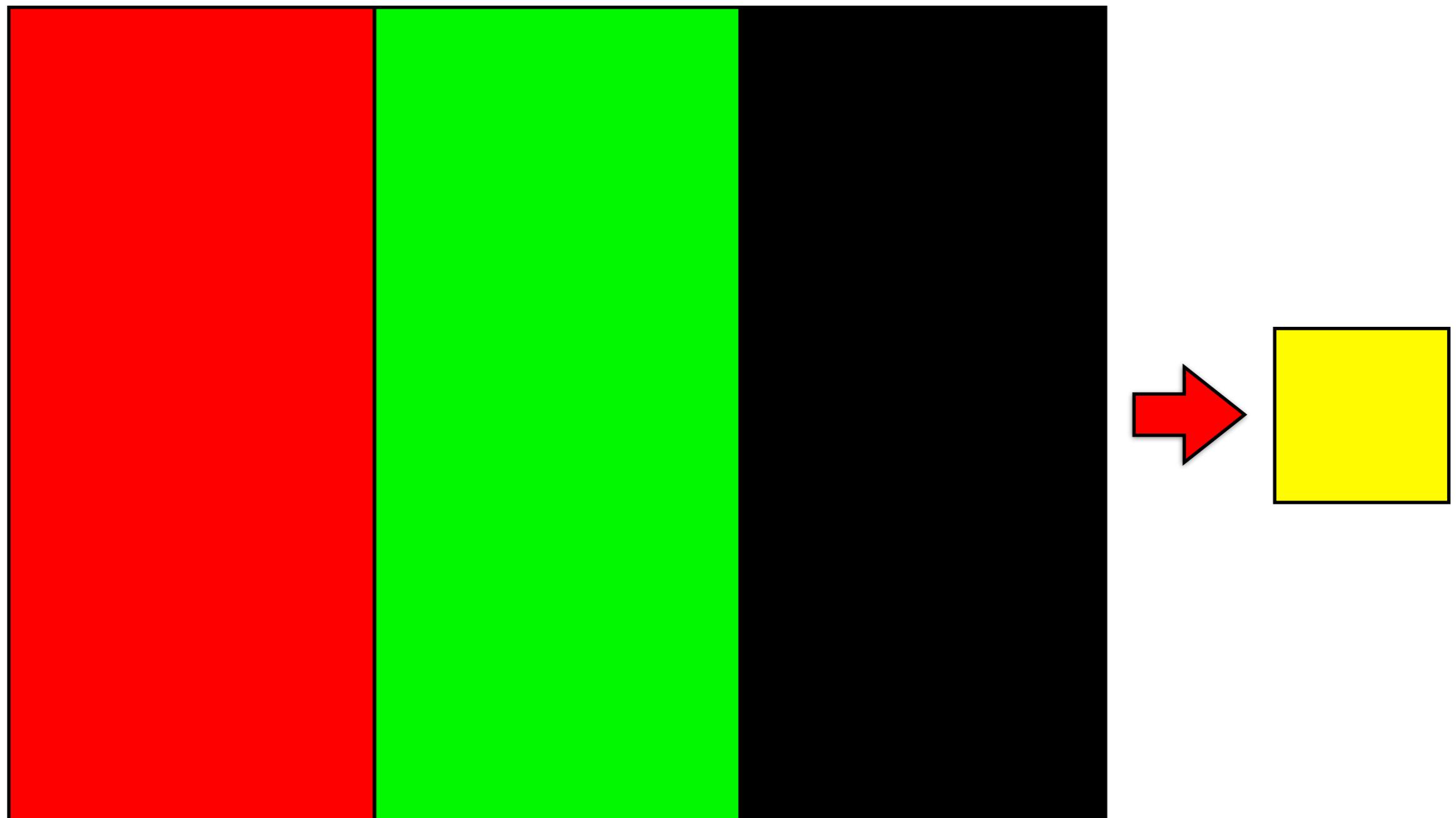
# Subpixel Beispiel



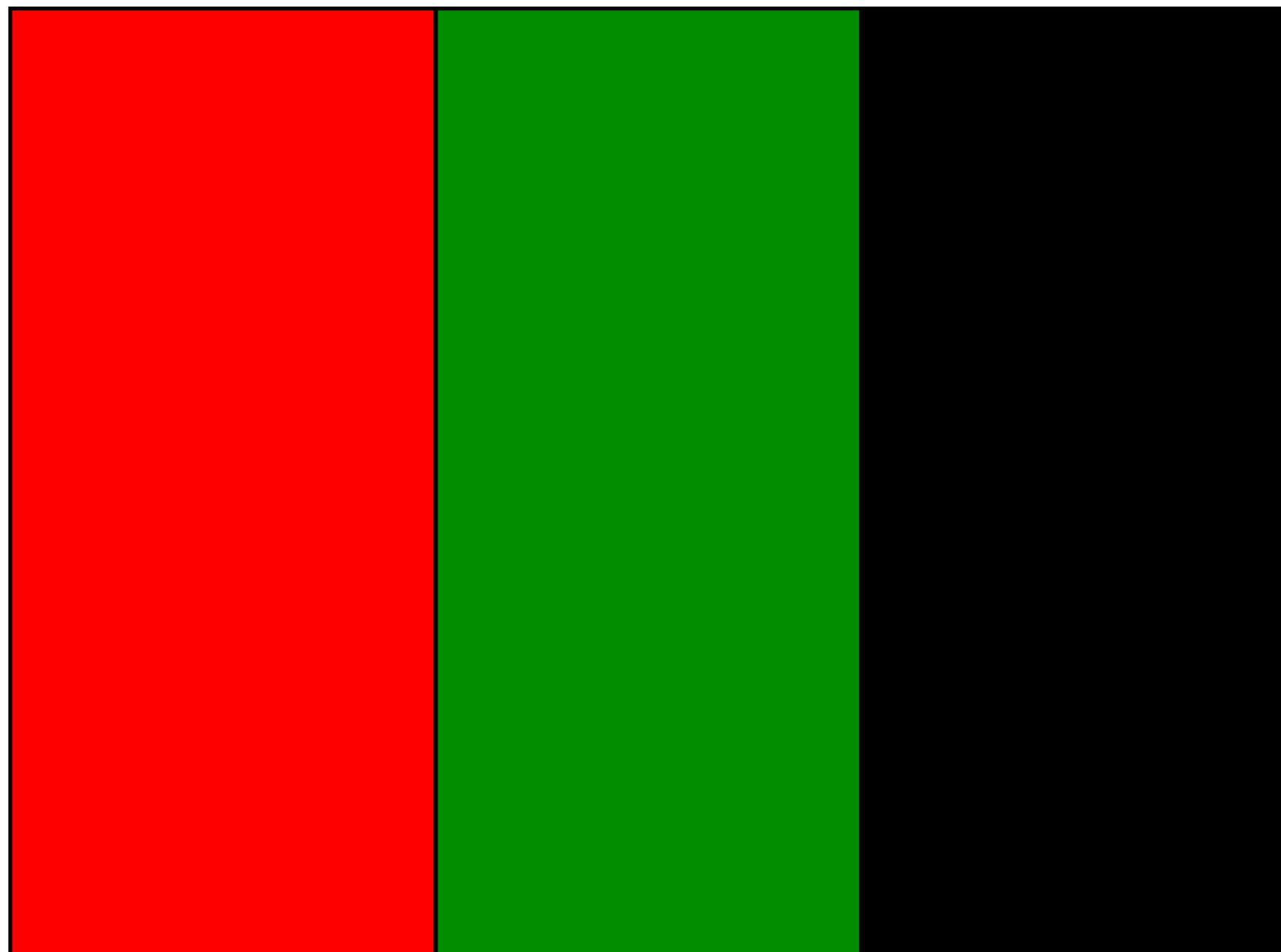
# Subpixel Beispiel



# Subpixel Beispiel



# Subpixel Beispiel



# Subpixel Beispiel



# Subpixel Beispiel



# RGB-Kanäle

## Beispiel



# RGB-Kanäle

Beispiel: Rot-Kanal



# RGB-Kanäle

## Beispiel: Grün-Kanal



G

# RGB-Kanäle

Beispiel: Blau-Kanal



# RGB-Kanäle

## Beispiel



# Gruppenarbeit

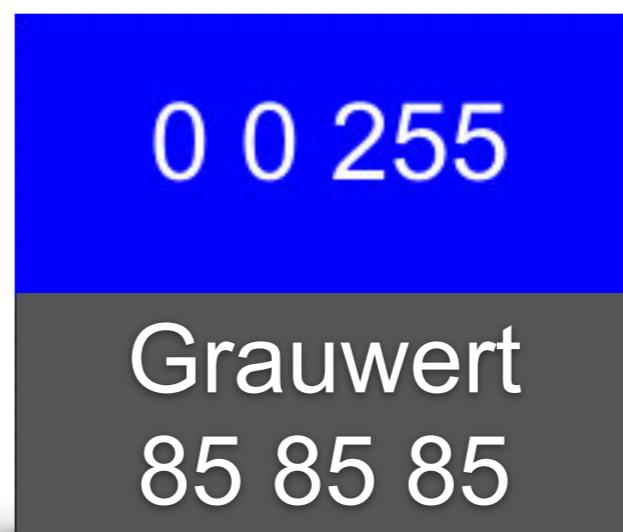
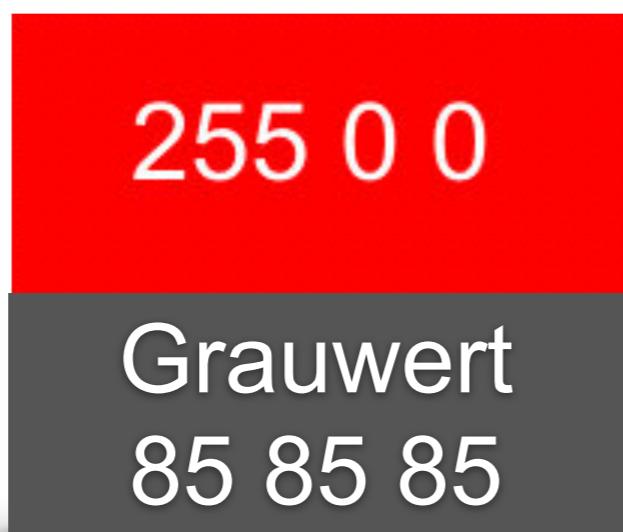


Wie lässt sich RGB- in Graustufenbild umwandeln?

# RGB-Farbmodell

## Naives Graustufenbild

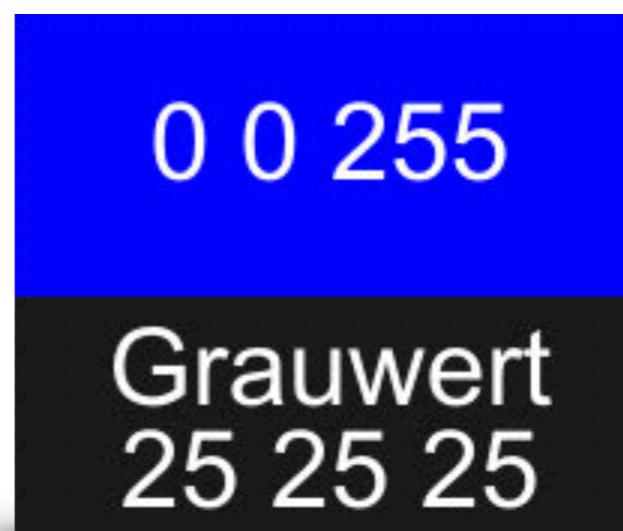
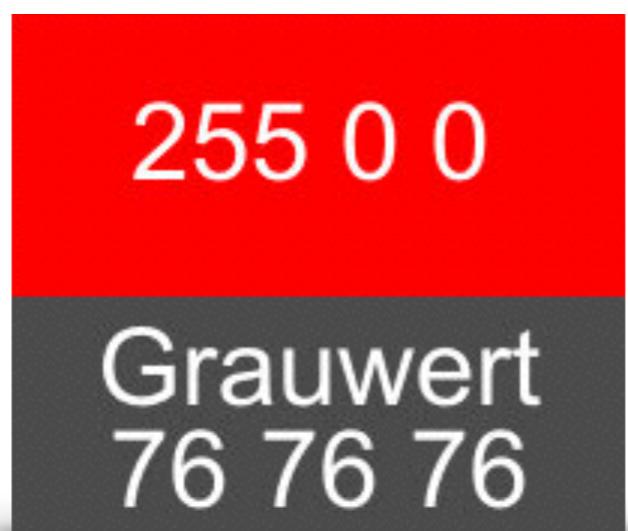
- Betrachtung des durchschnittlichen RGB-Werts, aber **Graustufenintensität**  $I = (R + B + G)/3$  liefert nur unzureichendes Grauweltbild
- Beispiel:



# RGB-Farbmodell

## Graustufenbild

- **Kanalgewichtung: Graustufenintensität**  
 $I = 0.299R + 0.587G + 0.114B$  (NTSC-Standard für Luminanz)
- Beispiel:





Original RGB-Bild



**Graustufenbild**  
**(ohne Kanalgewichtung)**



**Graustufenbild**  
**(mit Kanalgewichtung)**

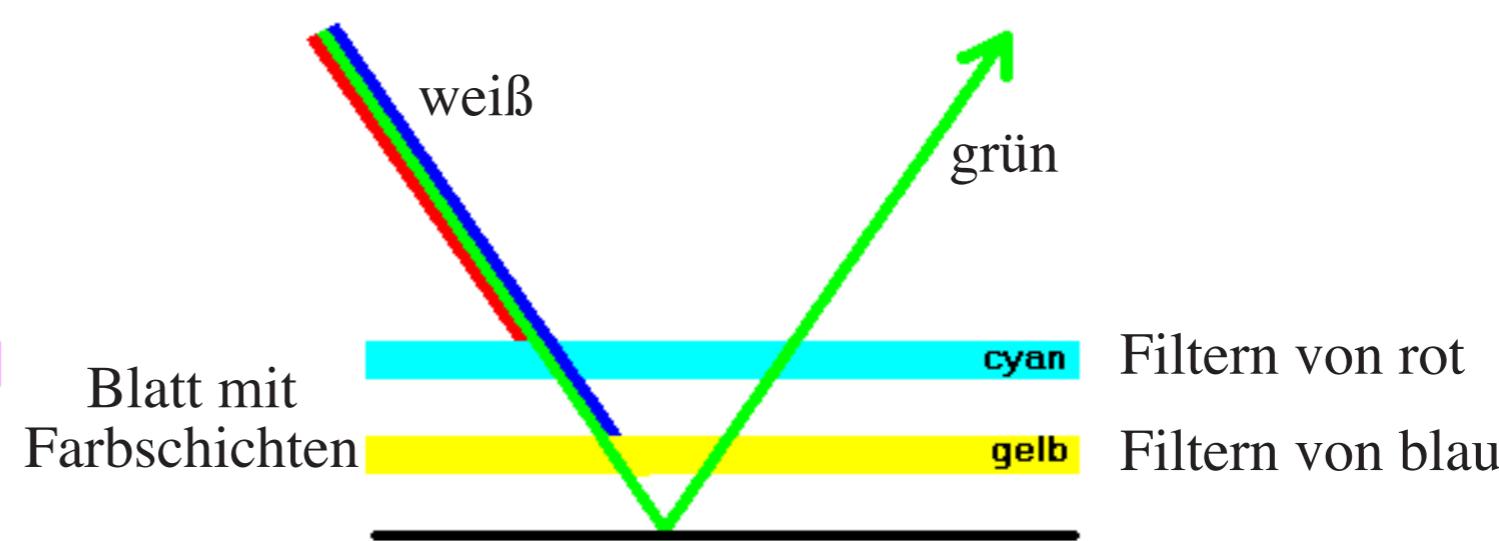
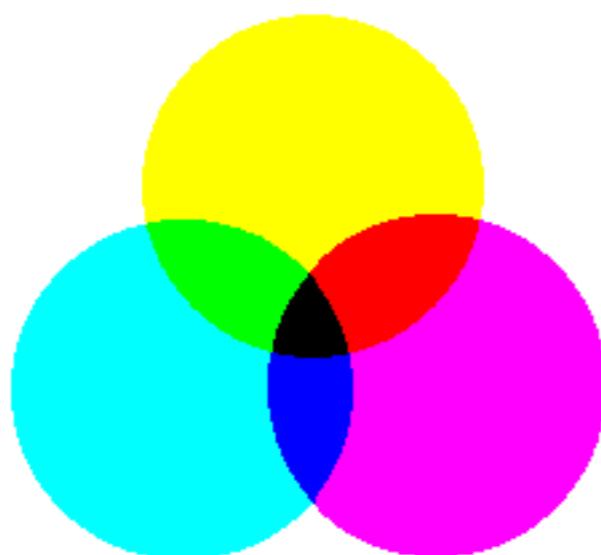
A black and white photograph showing several lily flowers in various stages of bloom. The flowers have long, pointed petals with small dark spots. The background is dark and out of focus.

# Farbsubtraktion

- wird Farbe auf weißes Papier hinzugefügt, ziehen wir Licht ab
  - z.B.: weißes Licht auf grünes Papier
    - blau und rot wird absorbiert und nur grün reflektiert
- subtraktive Farbmischung: Hinzufügen von Farbe führt zur Absorption der im RGB-Würfel gegenüberliegenden Farbe

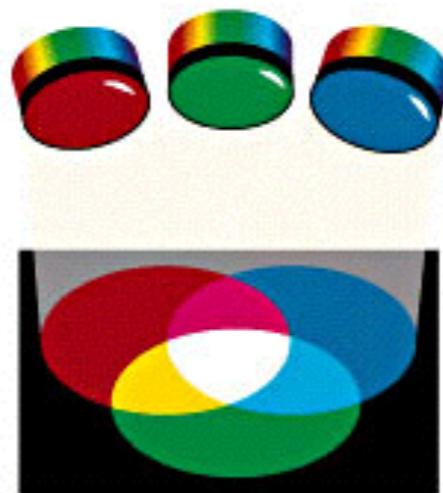
# CMY-Farbmodell

- **CMY-Farbmodell:** Hinzufügen von subtraktiven Primärfarben Cyan, Magenta und Gelb führt zur Absorption der im RGB-Würfel gegenüberliegenden Komplementärfarben Rot, Grün und Blau

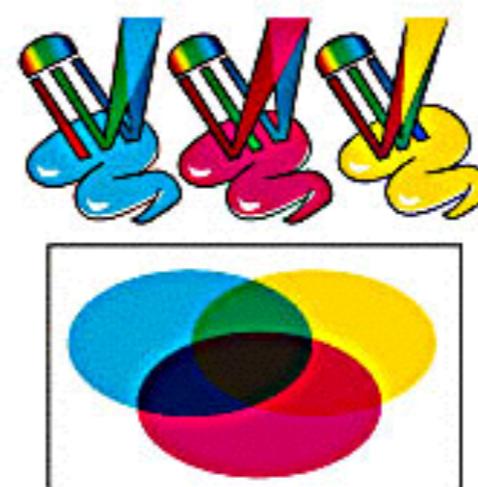
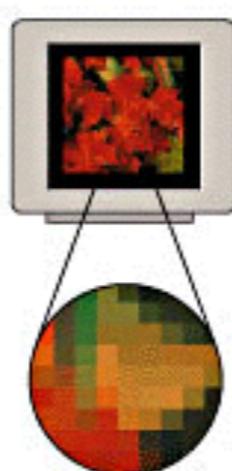


# Additiv vs. Subtraktiv

- leuchtende Geräte (Monitore etc.) nutzen i.d.R. additives Farbmodell
- Licht reflektierende Medien (Drucke etc.) nutzen i.d.R. subtraktives Farbmodell



Additive Farben (RGB)



Subtraktive Farben(CMYK)



# Farbmodelle

## Konvertierung RGB nach CMY

- Konvertierung vom additiven (RGB: Red, Green, Blue) ins subtraktive (CMY: Cyan, Magenta, Yellow) Farbmodell

$$f_{RGB} = \begin{pmatrix} f_R \\ f_G \\ f_B \end{pmatrix} = \begin{pmatrix} f_{max} - f_C \\ f_{max} - f_M \\ f_{max} - f_Y \end{pmatrix}$$
$$= \begin{pmatrix} f_{max} \\ f_{max} \\ f_{max} \end{pmatrix} - \begin{pmatrix} f_C \\ f_M \\ f_Y \end{pmatrix} = \text{weiß}_{RGB} - f_{CMY}$$

# Gruppenarbeit



Wie lässt sich Türkis (80,200,130)<sub>RGB</sub> von RGB in CMY in 8-Bit-Darstellung (pro Kanal) umrechnen?

# Farbmodelle

## Konvertierung RGB nach CMY

- Konvertierung zwischen additiven und subtraktiven Farbmodell bei normalisierter Darstellung

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} - \begin{pmatrix} C \\ M \\ Y \end{pmatrix}$$

bzw.

$$\begin{pmatrix} C \\ M \\ Y \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} - \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

# CMY-Farbmodells

## Problem

- schwarz ist am häufigsten verwendete Farbe von Schriftstücken
- schwarz im CMY-Farbmodell (1,1,1) ist in Praxis eher grau-braun
- durch Übereinanderdrucken aller drei Farbstoffe mit voller Intensität weicht Papier auf
- schwarze Farbe ist günstiger als Mischung von Cyan, Magenta und Gelb

# CMYK-Farbmodell

- CMYK-Farbmodell erweitert CMY um “Key“
  - Farbseparation von “Schwarz”, was ansonsten durch Absorption von Cyan, Magenta und Yellow erfolgen muss
- optimiert für Drucker

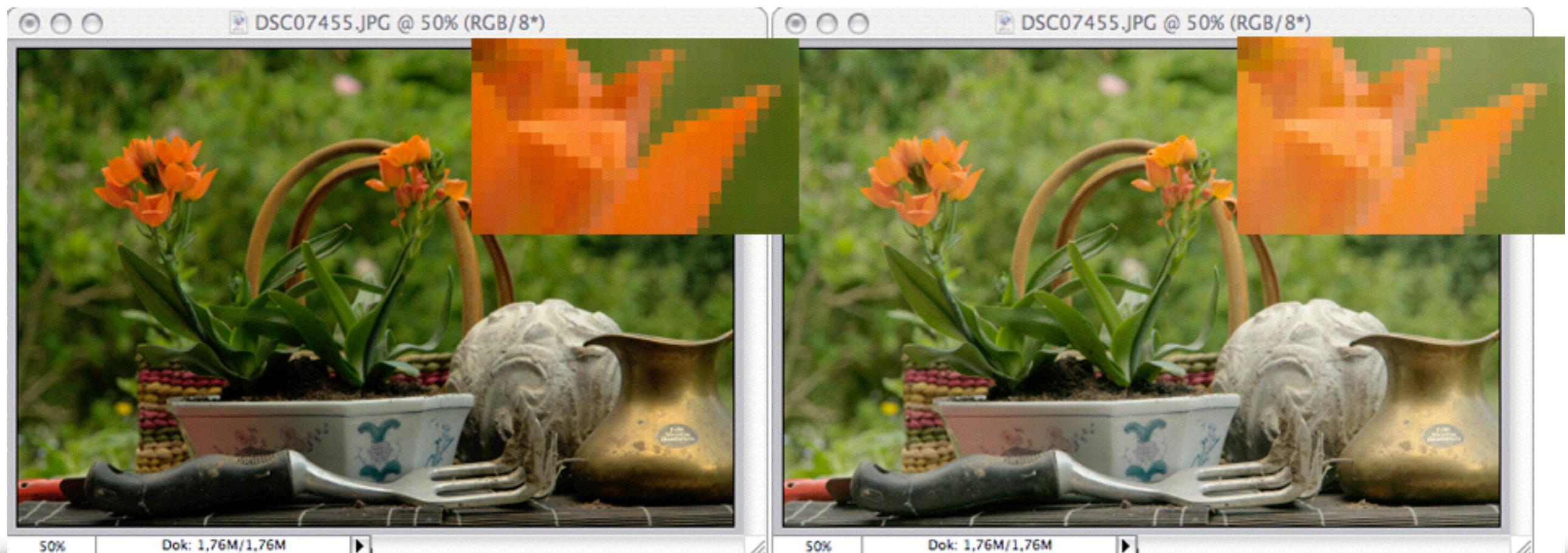
# Farbmodelle

## Benutzerorientiert

- Definition von Farben in additiven/ subtraktiven Farbmodelle nicht intuitiv
  - z.B. Aufhellen von einem definiertem Rot: Wie müssen R-, G- und B-Werte verändert werden?
- im Gegensatz zu den hardware-orientierten Farbmodellen RGB und CMY(K) orientiert sich HSV- und HLS-Farbmodell am Benutzer

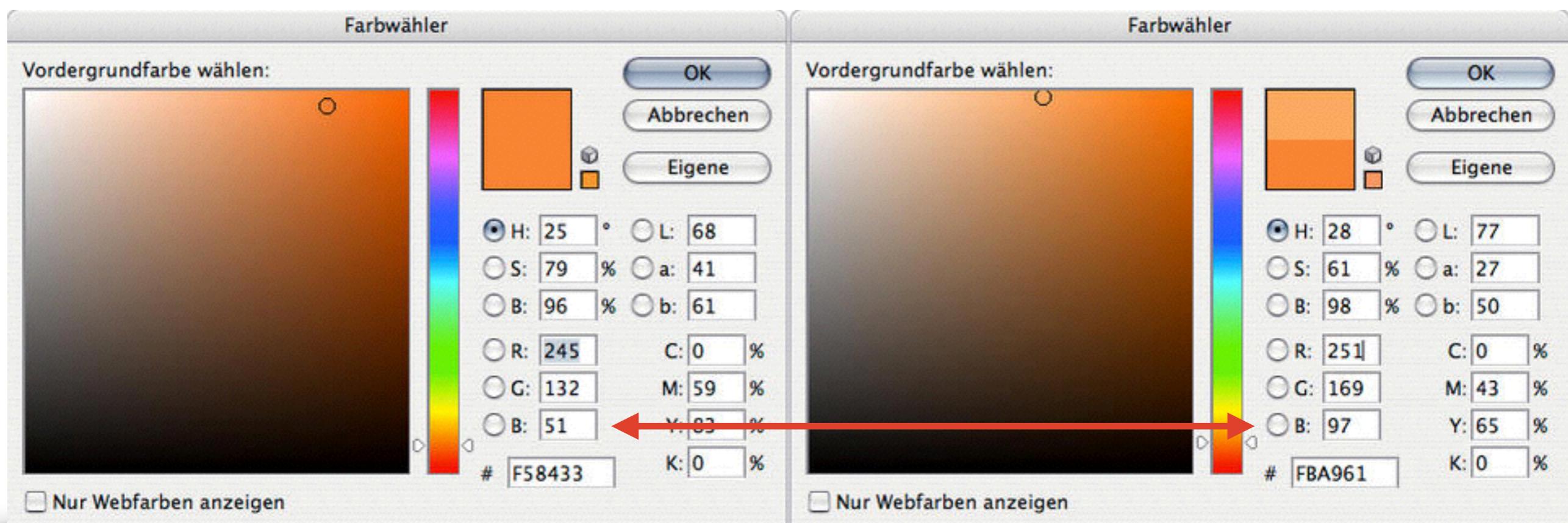
# RGB-Farbmodell

## Farbmanipulation



# RGB-Farbmodell

## Farbmanipulation

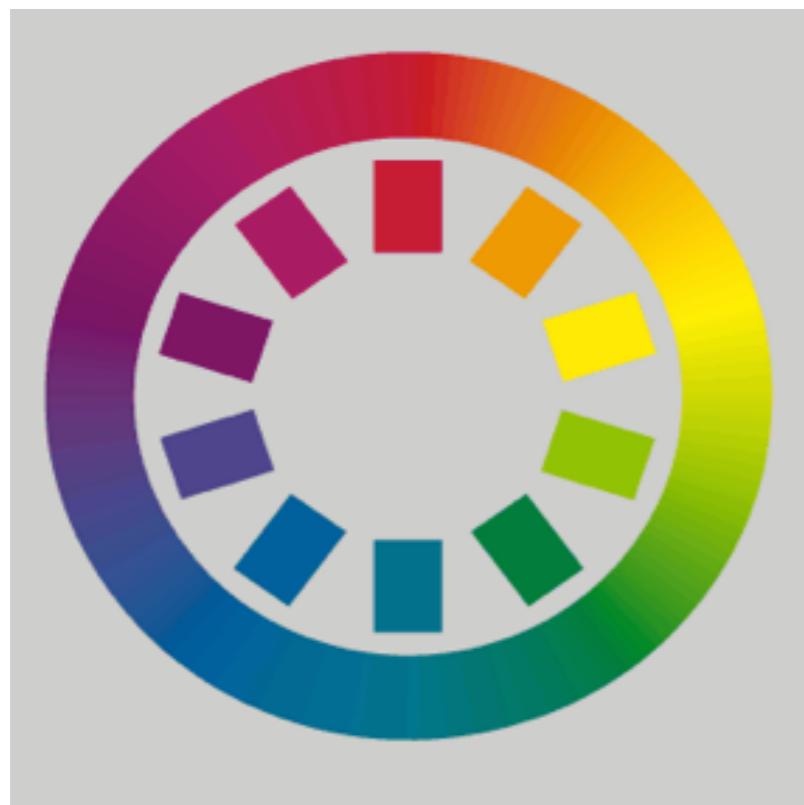




Albert H. Munsell: *Atlas of the Munsell Color System*, 1915

# Munsell-Notation

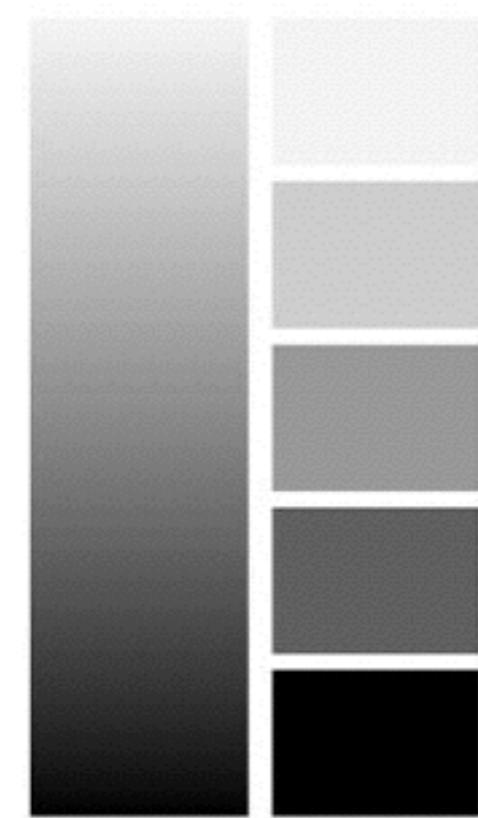
## Beispiel



Hue

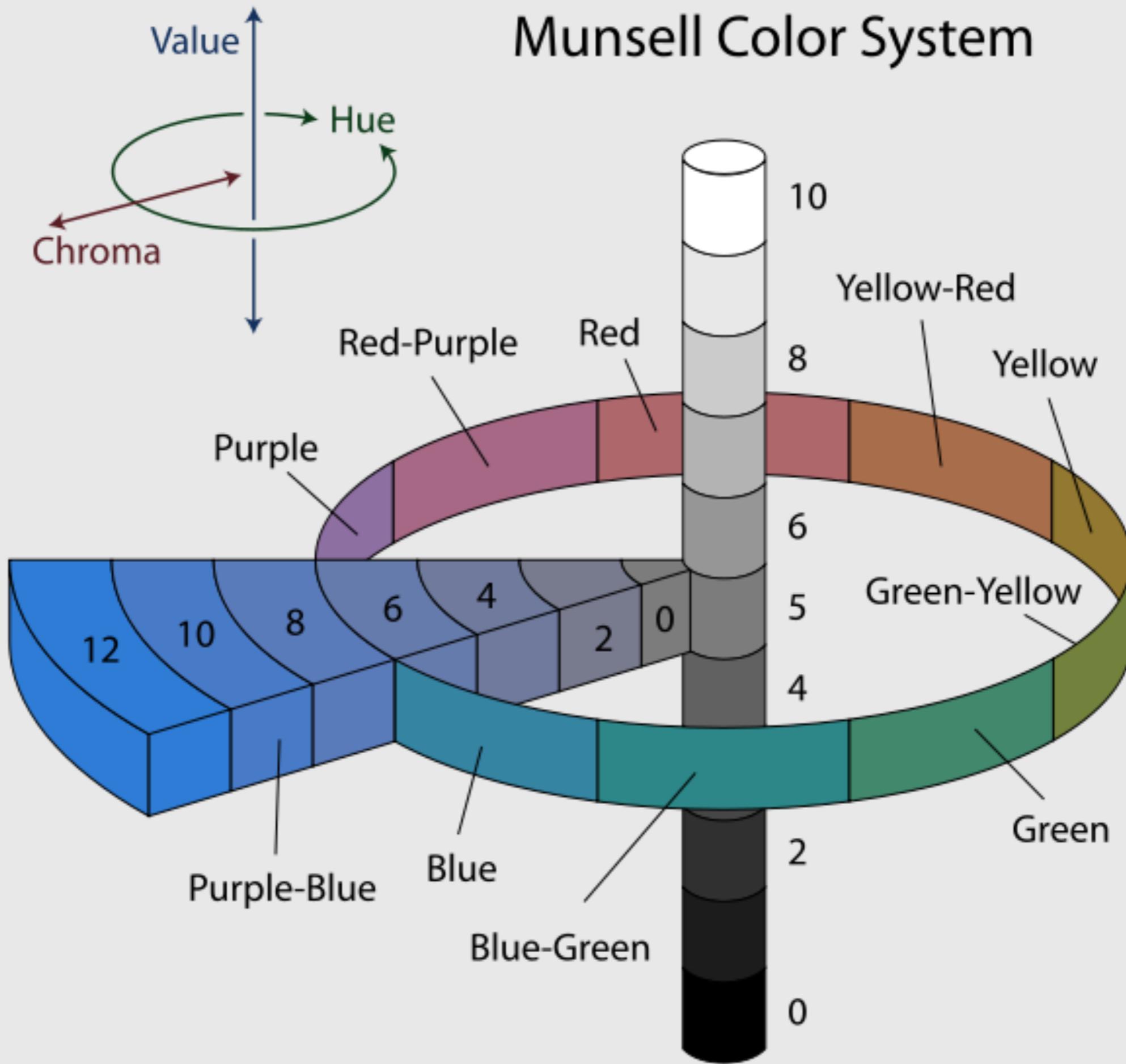


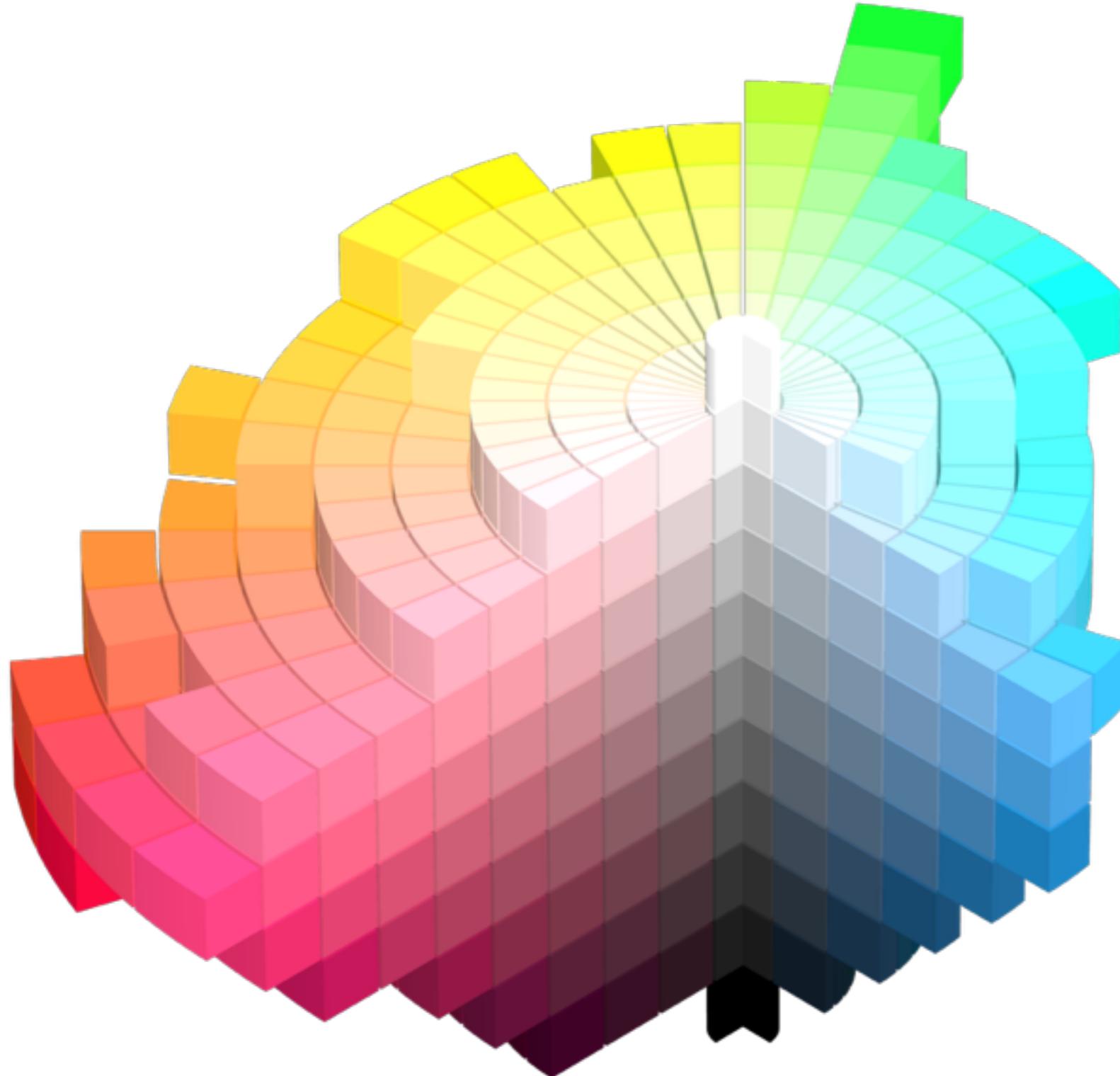
Chroma



Value

# Munsell Color System





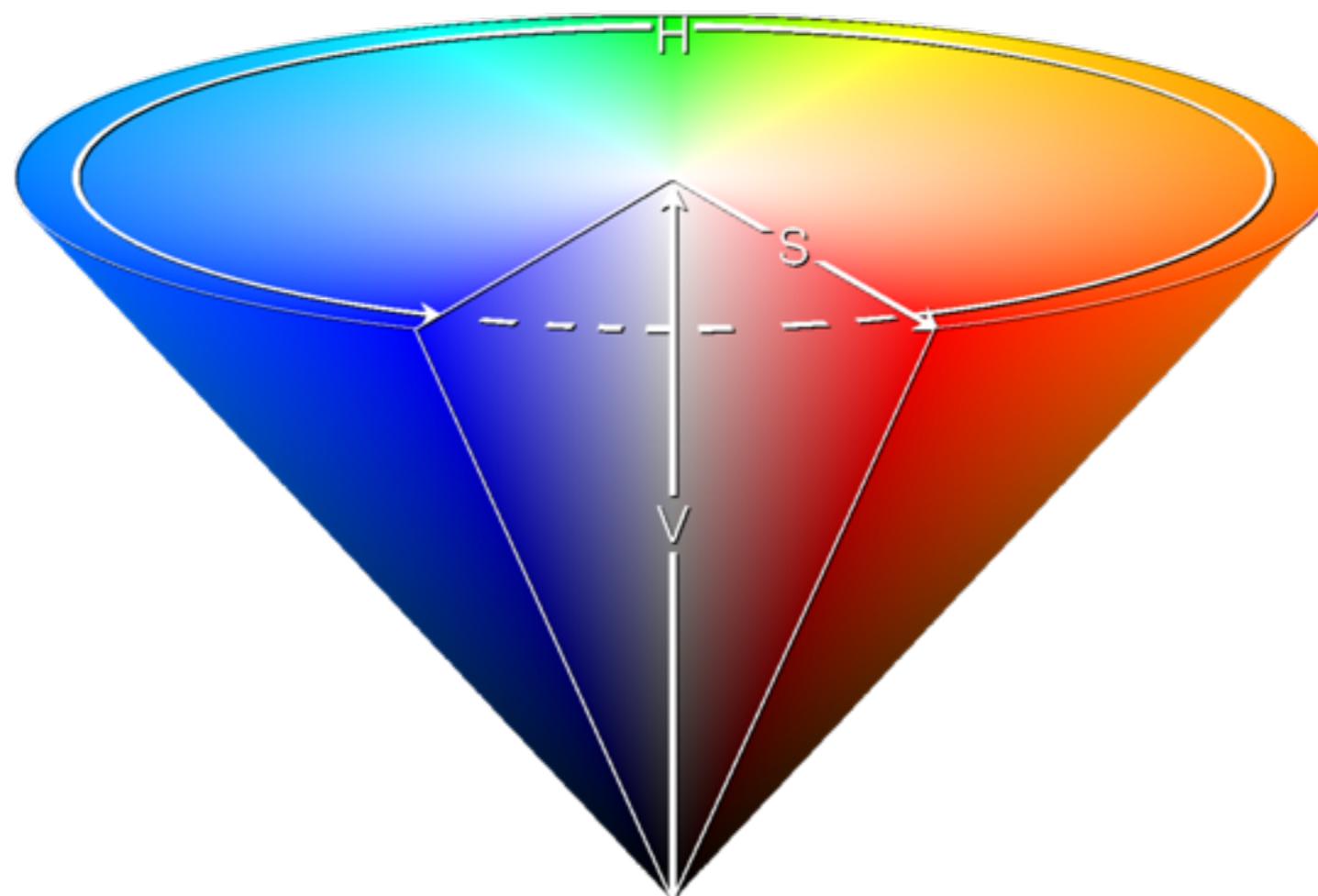
Munsell Color System

# HSV-Farbmodell

- **HSV-Farbmodell** ist benutzerorientiertes Farbmodell und basiert auf drei psychophysiologischen Größen
  1. **Hue (Farbton)**
  2. **Saturation/Chroma (Sättigung)**
  3. **Value/Brightness/Intensity/Lightness (Helligkeit)**

# HSV-Farbmodell

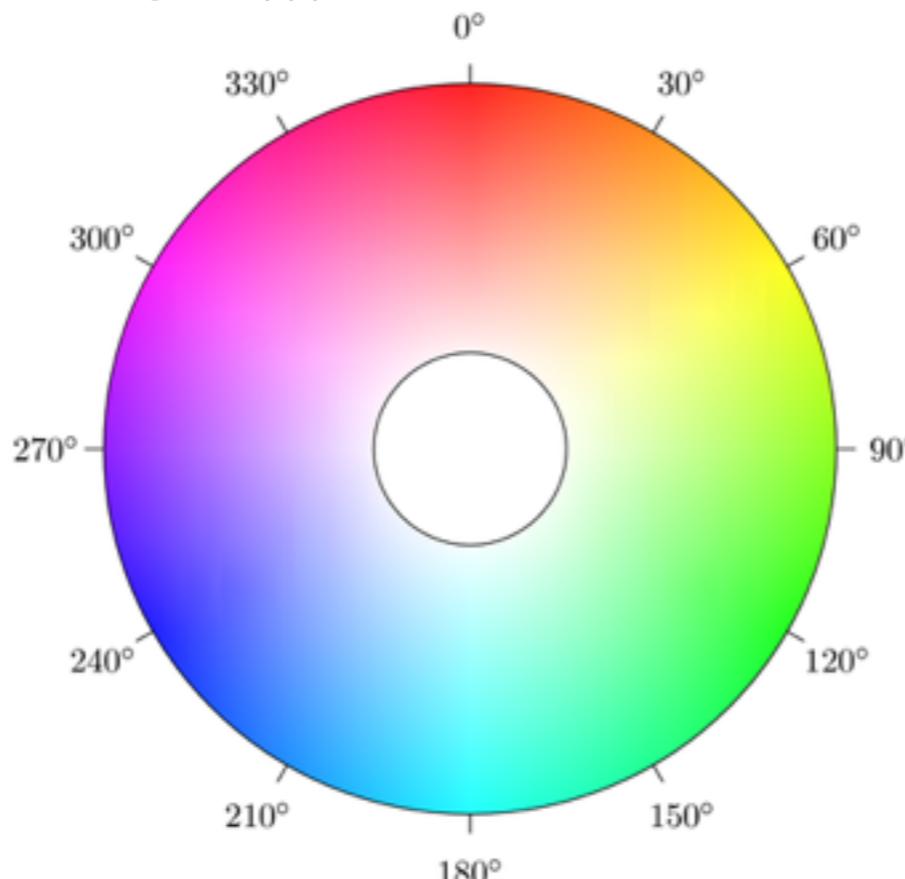
- Geometrisch besteht HSV-Farbmodell aus auf Spitze stehendem Kegel mit Grundfläche



# HSV-Farbmodell

## Farnton

- **Farnton** wird in Grad zwischen 0 und 360 angegeben und geometrisch als Winkel dargestellt, z.B. Rot hat Wert 0°, Gelb 60°, Grün 120° ...



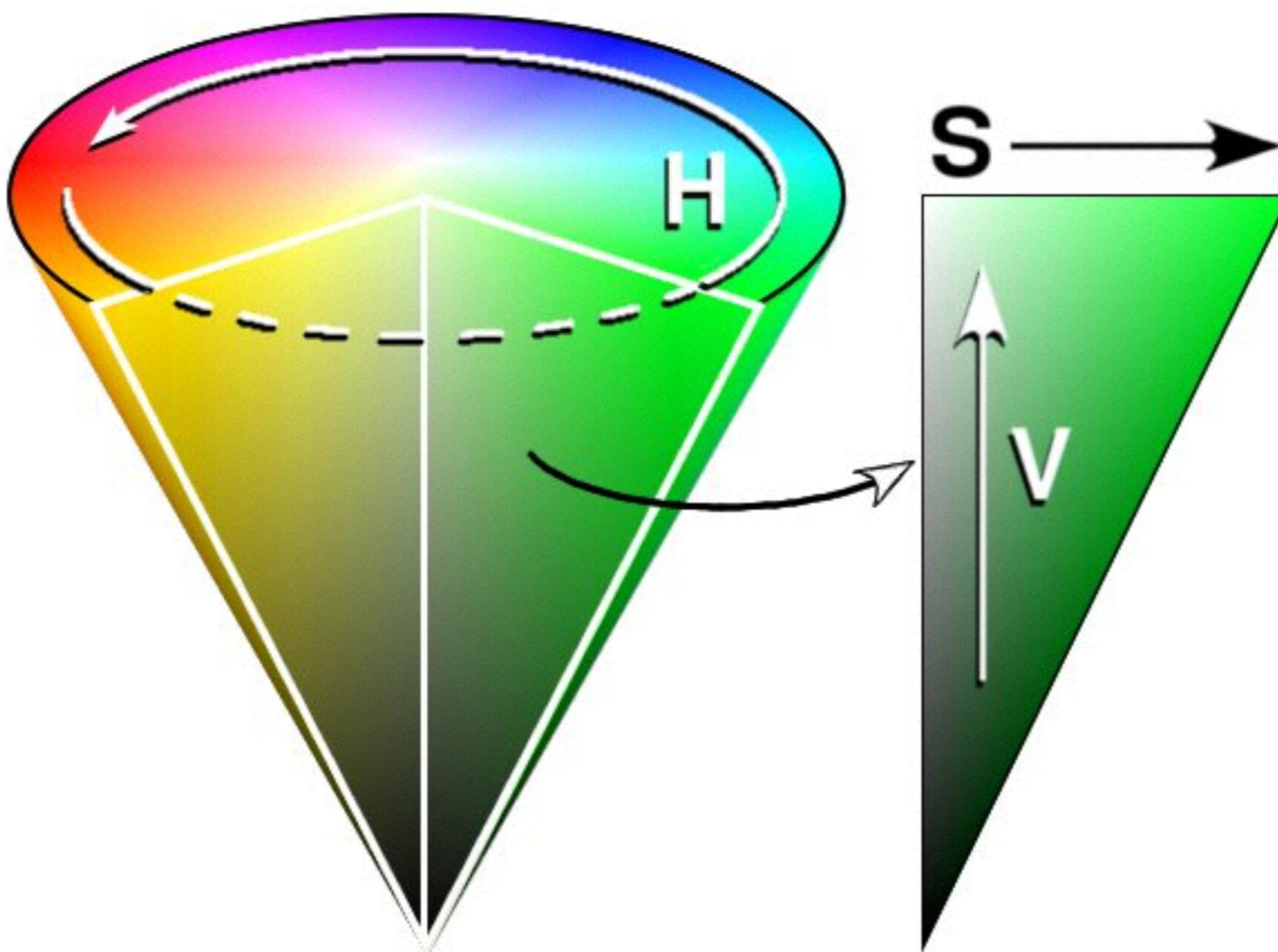
# HSV-Farbmodell

## Farnton

- Um Farben gleichen Farbtöns zu erhalten (z.B.  $H = 120$ ), wird senkrechte Scheibe von Schwarz-Weiß-Achse bis zum Mantel herausgeschnitten
- rechtwinkliges Dreieck enthält alle Farben konstanten Farbtöns

# HSV-Farbmodell

Beispiel: Farbtöne



# HSV-Farbmodell

## Helligkeit

- Helligkeit (engl. *Brightness, Lightness*)
  - wahrgenommene Intensität
    - ▶ *Lightness*: bei reflektierenden Objekten
    - ▶ *Brightness*: bei selbstleuchtenden Objekten

# HSV-Farbmodell

## Beispiel: Helligkeit

- **Helligkeit** wird durch Entfernung zwischen Spitze und Grundfläche der Pyramide beschrieben
- Werte zwischen 0 (Spitze des Kegels) und 1 (Grundfläche)



# HSV-Farbmodell

## Beispiel: Helligkeit

- Mittelpunkt der Grundfläche stellt Farbe “Weiß” und Spitze Farbe “Schwarz” dar
- **Grauachse** mit Grautönen zwischen Schwarz und Weiß



# HSV-Farbmodell

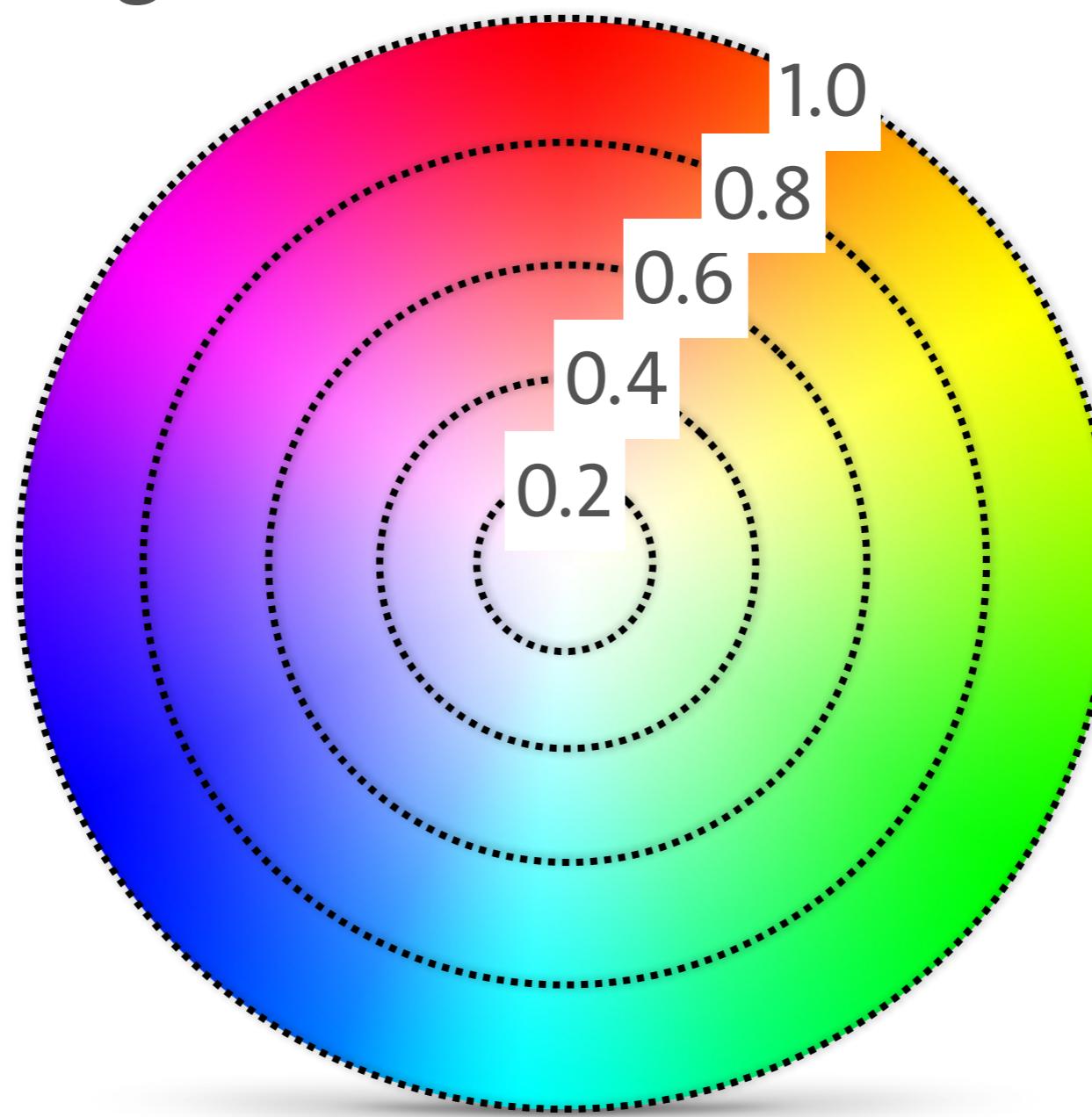
## Sättigung

- **Sättigung** wird durch relative Entfernung von Grauachse in horizontaler Ebene dargestellt
  - auf Grauachse ist Sättigung 0, am äußeren Rand ist sie 1
- Farben gleicher Sättigung (und konstanter Helligkeit) liegen somit auf Kreis, der parallel zum Rand der Grundfläche verläuft

# HSV-Farbmodell

## Beispiel: Sättigung

- Sättigung  $S = 0.2, 0.4, 0.6, 0.8$  und  $1.0$



# HSV-Farbmodell

## Vorgehen zur Farbauswahl

1. Farnton auswählen, indem Wert für H (bei V = 1 und S = 1) verändert wird
2. Wert für Sättigung festlegen (S)
3. Helligkeit der Farbe definieren (V)

# HSV-Farbmodell

## Beispiel: Farbton

- HSV-Modell unterscheidet zwischen Farb- und Helligkeitsinformation
  - Farbinformation wird durch Farbton und -sättigung definiert
  - Helligkeitsinformation getrennt davon



# Farbmodell

## Vergleich: HSV vs. RGB

Farbe	H	S	V	R	G	B
Schwarz	–	–	0 %	0 %	0 %	0 %
Rot	0°	100 %	100 %	100 %	0 %	0 %
Gelb	60°	100 %	100 %	100 %	100 %	0 %
Braun	24,3°	75 %	36,1 %	36 %	20 %	9 %
Weiß	–	0 %	100 %	100 %	100 %	100 %
Grün	120°	100 %	100 %	0 %	100 %	0 %
Dunkelgrün	120°	100 %	50 %	0 %	50 %	0 %
Cyan	180°	100 %	100 %	0 %	100 %	100 %
Blau	240°	100 %	100 %	0 %	0 %	100 %
Magenta	300°	100 %	100 %	100 %	0 %	100 %

# YCbCr-Farbmodell

- wurde für Digitalfernsehen nach Norm PAL entwickelt
  - auch beim digitalen NTSC-Fernsehen (siehe VL *Interaktive Medien - Videos*)
- teilt Farbinformation auf in **Grundhelligkeit Y** und zwei **Farbkomponenten Cb (Blue-Yellow Chrominance)** und **Cr (Red-Green Chrominance)**

# YCbCr-Farbmodell

## Beispiel



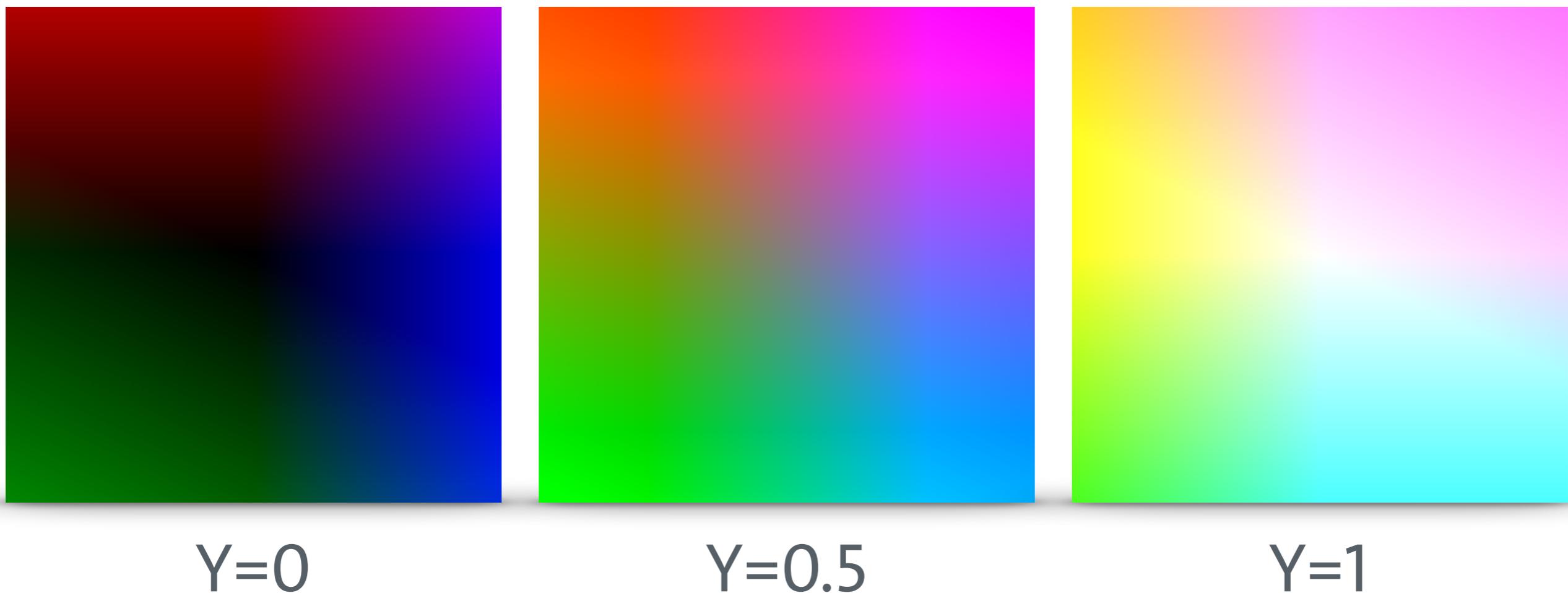
# YCbCr-Farbmodell

## Beispiel



# YCbCr-Farbmodell

Beispiel: Helligkeit



# YCbCr-Farbmodell

## Beispiel



# Diskussion



Diskutieren Sie die Vorteile des YCbCr-Farbmodells für die Fernsehtechnik!

# YCbCr-Farbmodell

- Y-Kanal überträgt quasi Schwarzweißbild
- Cb und Cr sind flächiger, d.h. benachbarte Bildpunkte haben häufig gleichen Wert
- mehr Informationen in Helligkeitswerten
  - meisten natürlichen Gegenstände sind einheitlich eingefärbt

# YCbCr-Farbmodell

- Umrechnung eines Ausgangsbilds (RGB-Kanäle mit 8-Bit) durch

$$\begin{bmatrix} Y' \\ Cb \\ Cr \end{bmatrix} \approx \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 0,299 & 0,587 & 0,114 \\ -0,168736 & -0,331264 & 0,5 \\ 0,5 & -0,418688 & -0,081312 \end{bmatrix} \cdot \begin{bmatrix} R'_d \\ G'_d \\ B'_d \end{bmatrix}$$

# Farbmodelle

## Geräteunabhängig

- RGB, CMY(K), HSV und YCbCr sind geräteabhängig
- Beispiel: im RGB-Farbmodell definierte Farbe sieht auf zwei verschiedenen Monitoren unterschiedlich aus, weil sich Leuchtstoffe und Primärfarben Rot, Grün und Blau von Monitor zu Monitor unterscheiden

# Farbmodelle

## Geräteunabhängig

- Internationale Beleuchtungskommission (Commission Internationale de l'Eclairage) (CIE) entwickelte geräteunabhängige Farbmodelle, um standardisierte Farbräume zu erhalten
- Modelle beruhen auf genau spezifizierten Primärfarben

# CIE XYZ Farbmodell

- drei Primärfarben Rot, Grün und Blau wurden bzgl. ihrer Wellenlänge durch CIE im Jahr 1931 standardisiert
  - Rot mit 700 nm
  - Grün mit 546,1 nm
  - Blau mit 435,8 nm

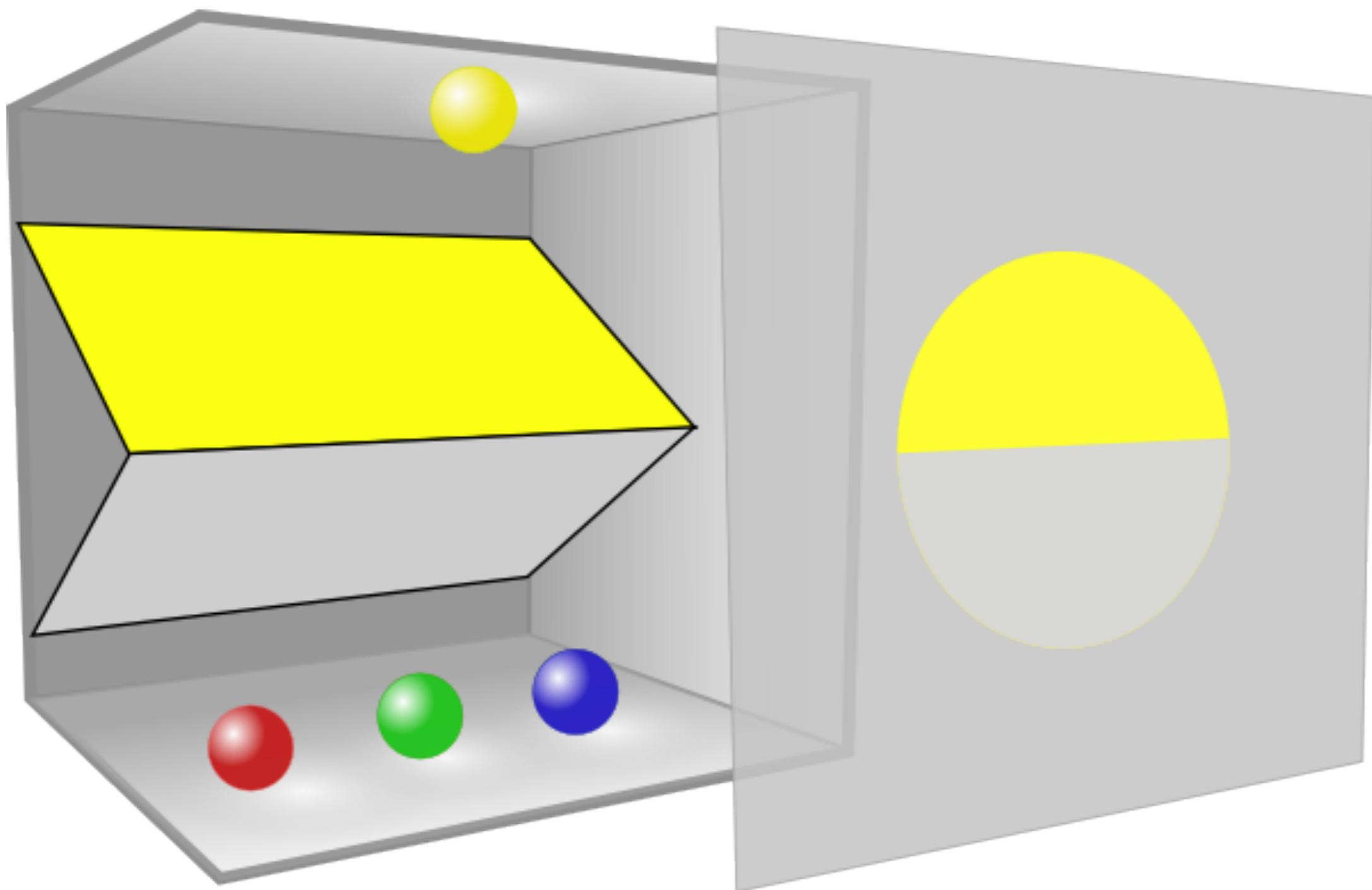
# CIE XYZ Farbmodell

## Experiment: Farbnachstellung

- Gruppe von ca. 20 Personen mussten (unter genau spezifizierten Bedingungen) monochromatisches Licht, d.h. Spektralfarben, die aus Licht einer einzigen Wellenlänge bestehen, mit Hilfe der drei definierten Primärfarben durch additive Farbmischung nachstellen

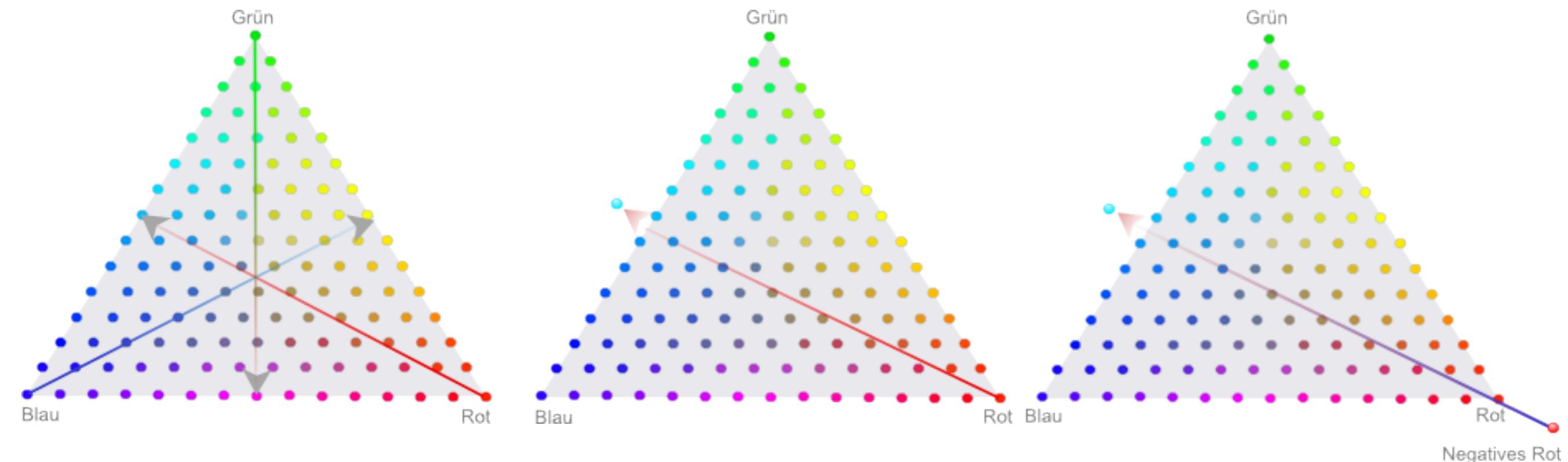
# CIE XYZ Farbmodell

Experiment: Farbnachstellung



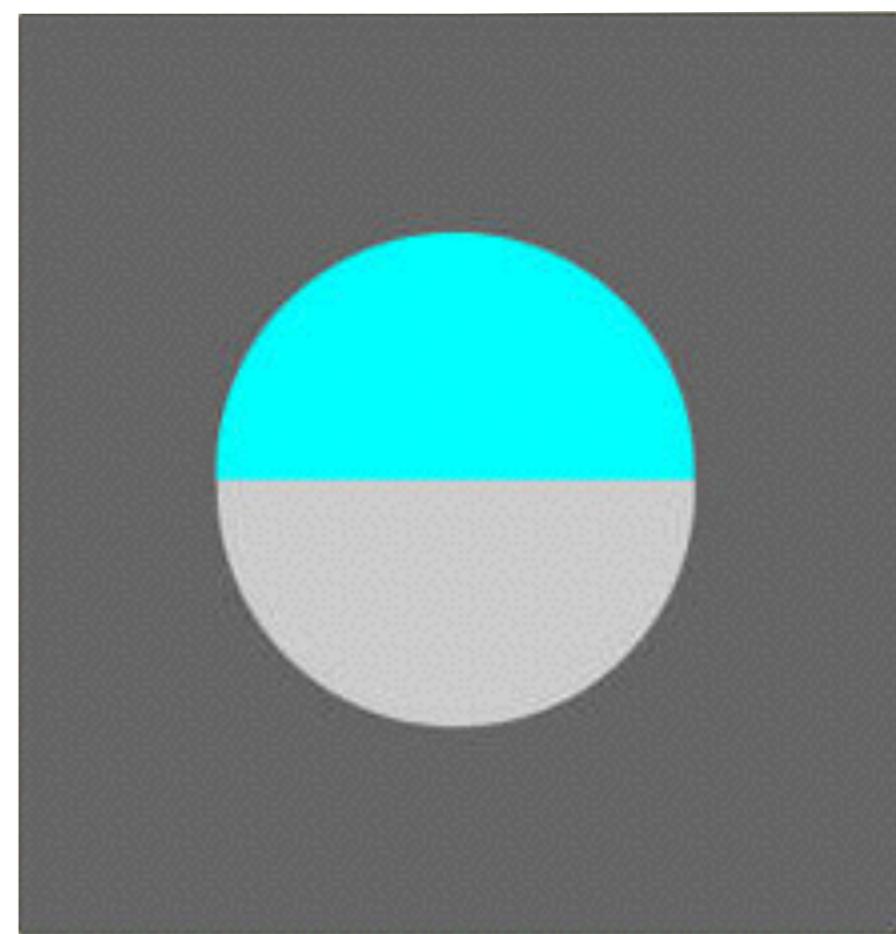
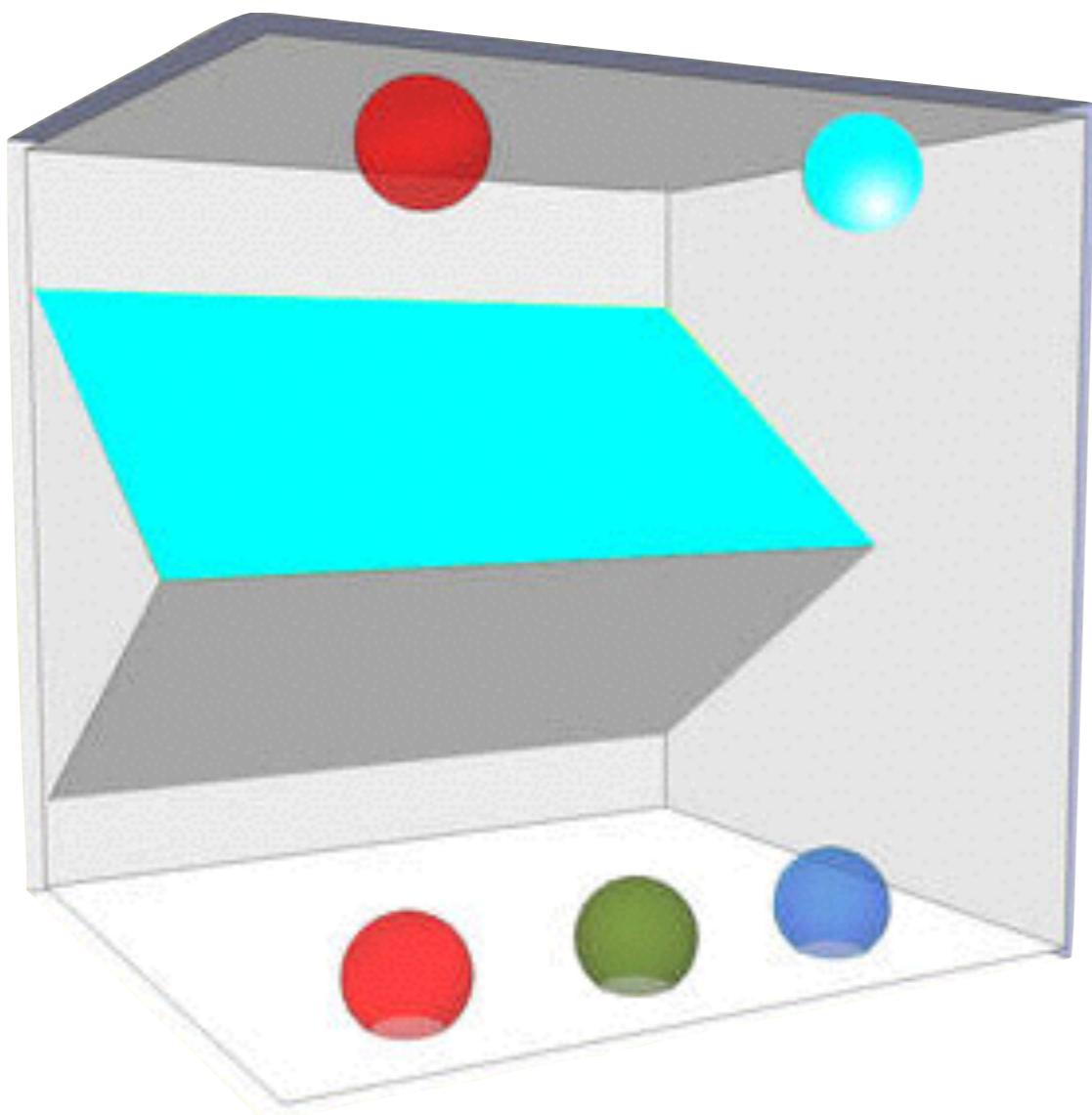
# Maxwell-Dreieck

## Illustration

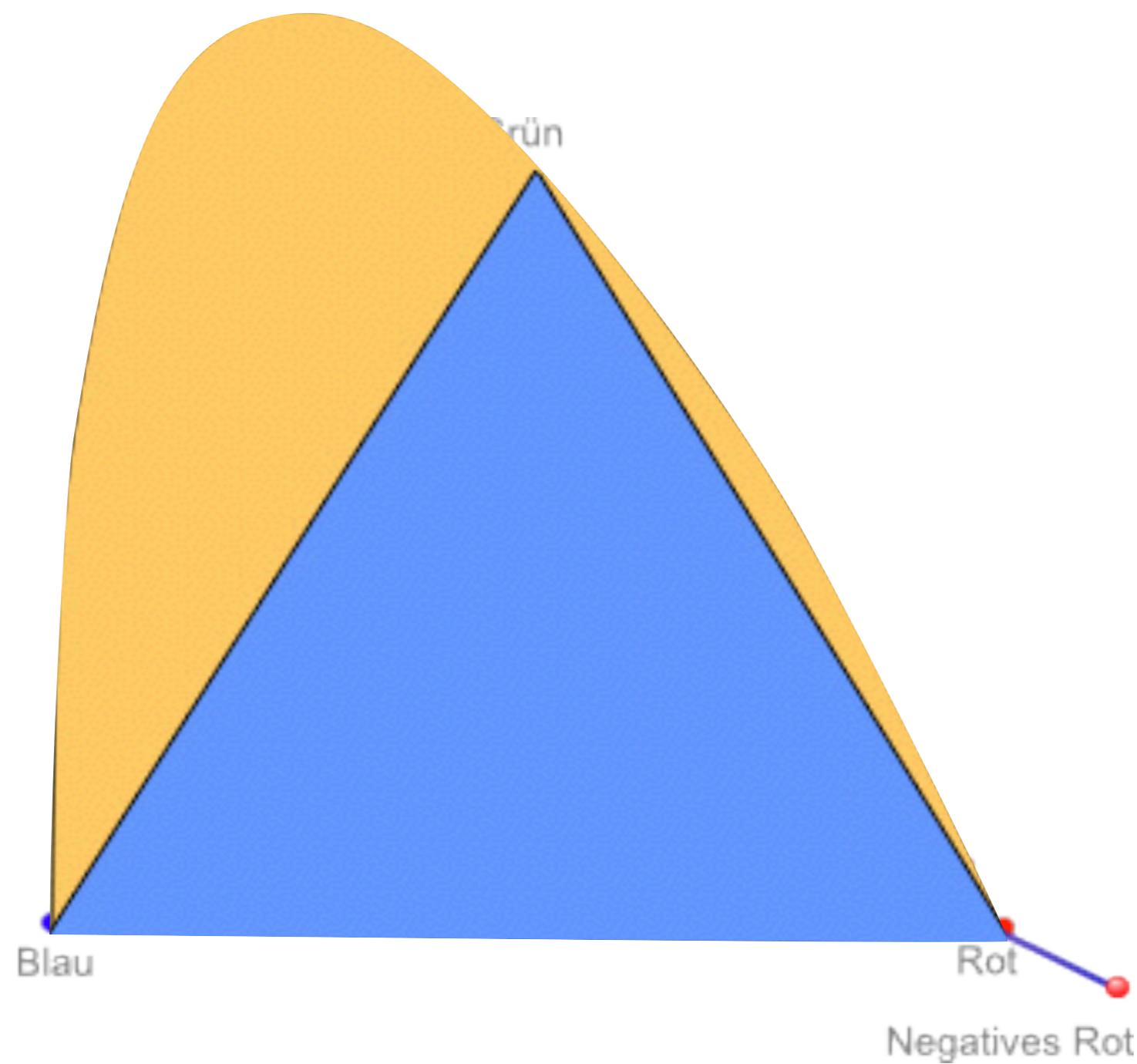


# CIE XYZ Farbmodell

## Experiment: Farbnachstellung

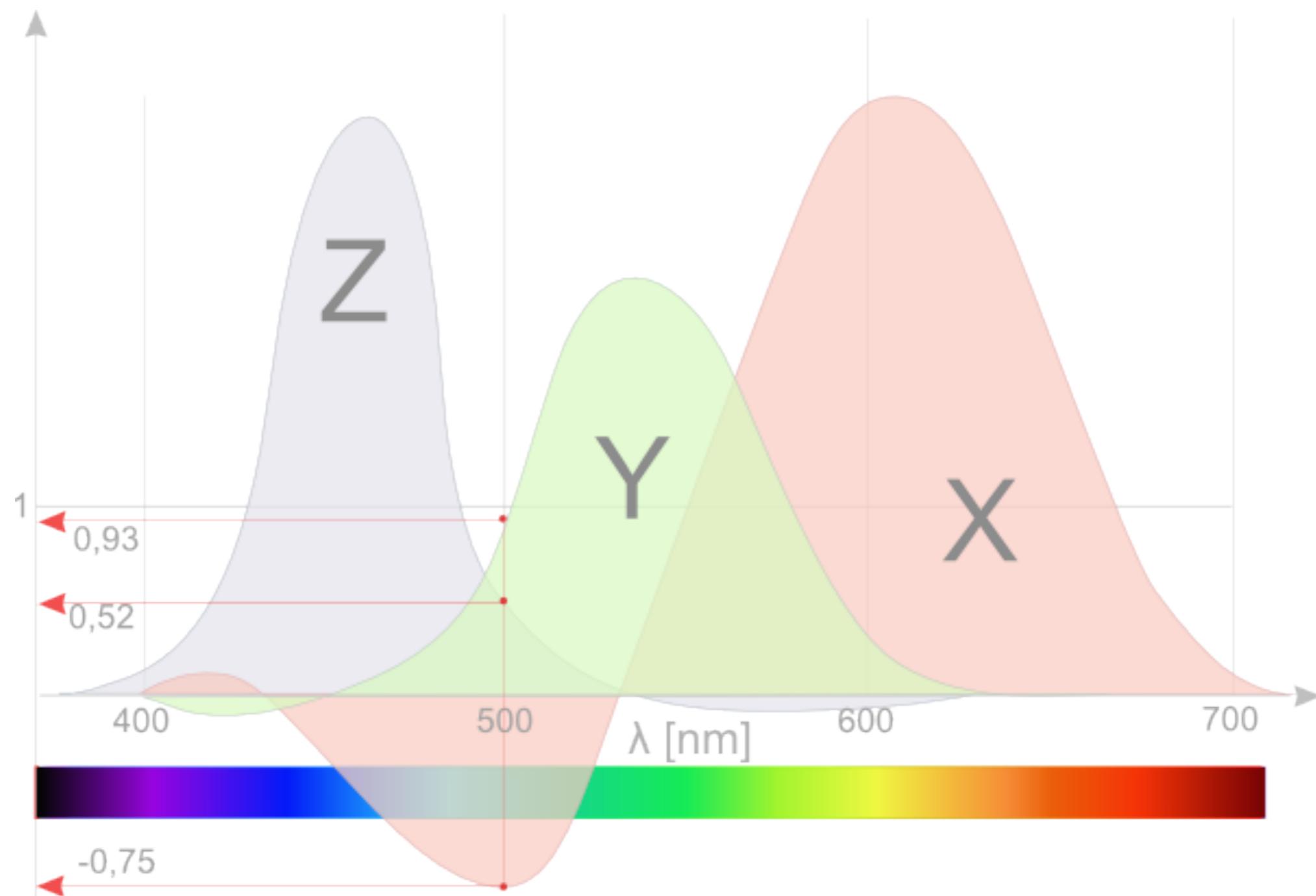


# Spektralfarbzug



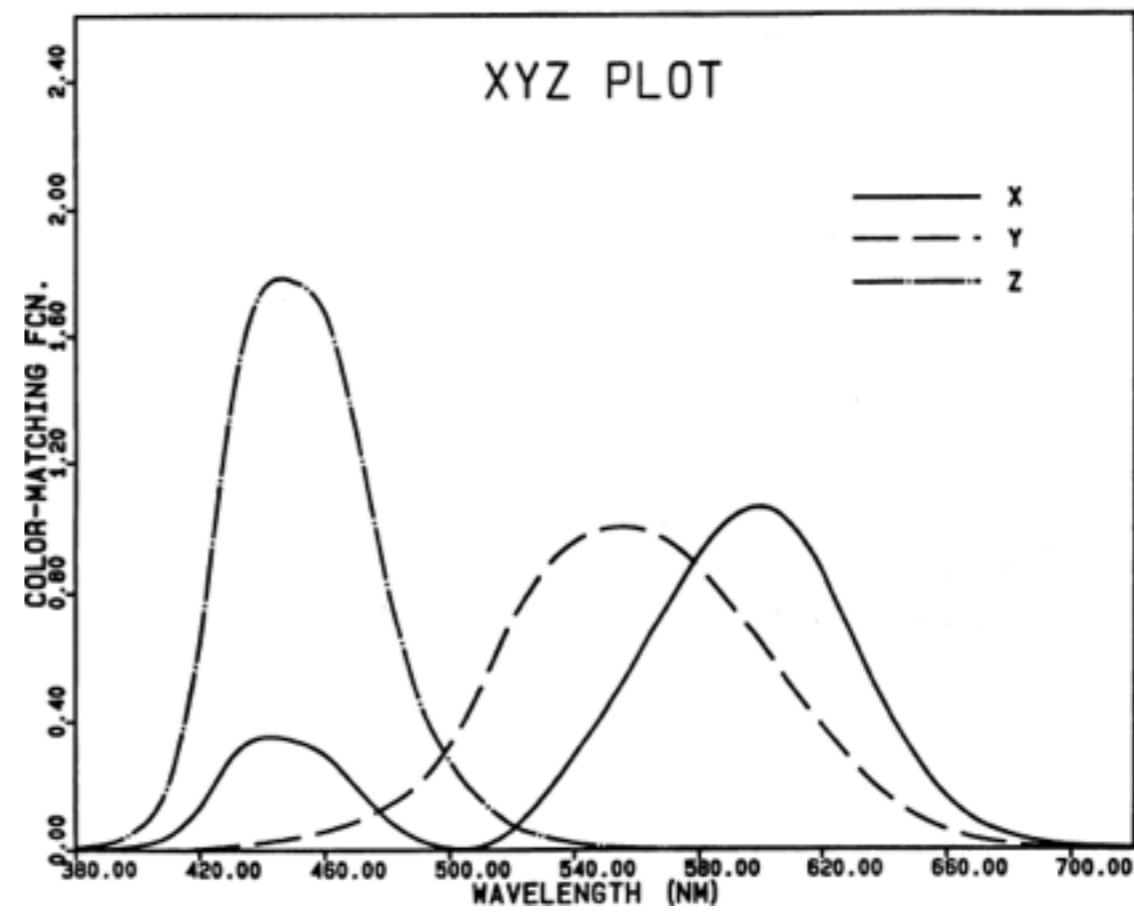
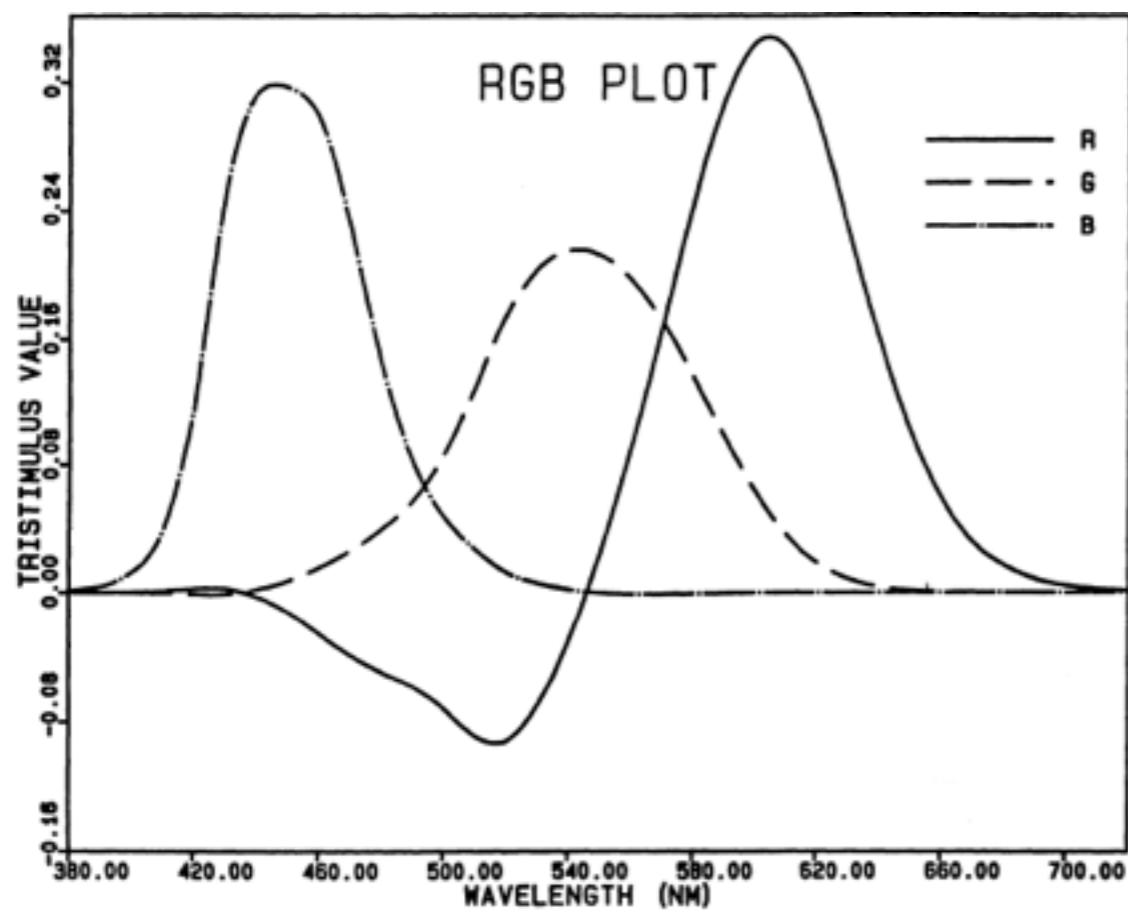
# CIE XYZ Farbmodell

## Experiment: Farbnachstellung



# CIE XYZ Farbmodell

## Experiment: Farbnachstellung



# CIE XYZ Farbmodell

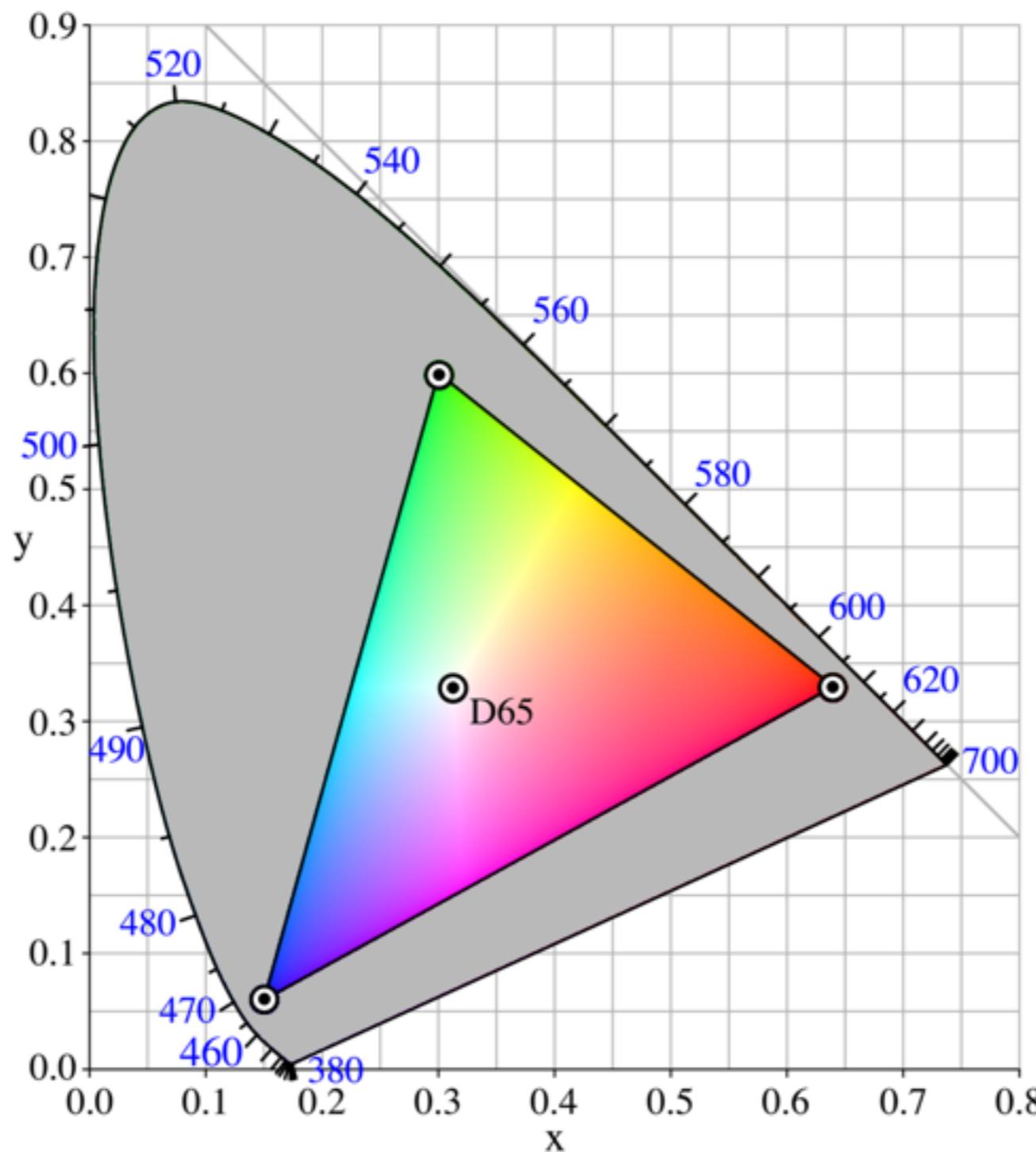
- CIE definiert drei Primärfarben X, Y und Z
- X, Y und Z beschreiben hypothetische, nicht realisierbare Farben, mit denen alle sichtbaren Farben erzeugt werden können
- Erzeugung einer Farbe kommt dabei ohne negative Werte aus

# CIE-Normfarbtafel

- **CIE-Normfarbtafel** ermöglicht 2D-Darstellung aller wahrnehmbarer Farben
- z lässt sich aus  $X + Y + Z = 1$  berechnen
- **Weißpunkt** wird durch Farbe ( $X=Y=Z=0.333\dots$ ) repräsentiert

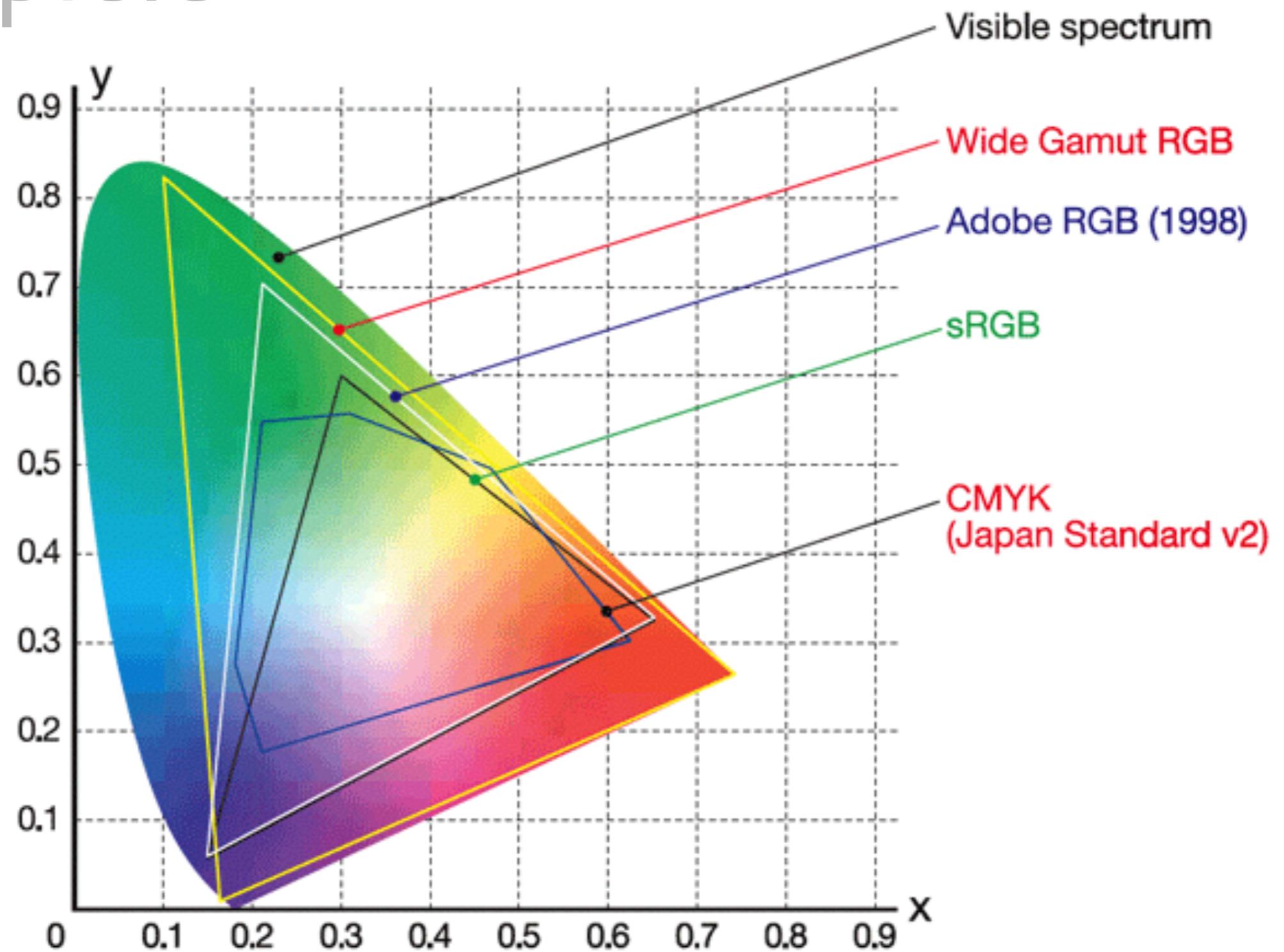
# CIE-Normfarbtafel

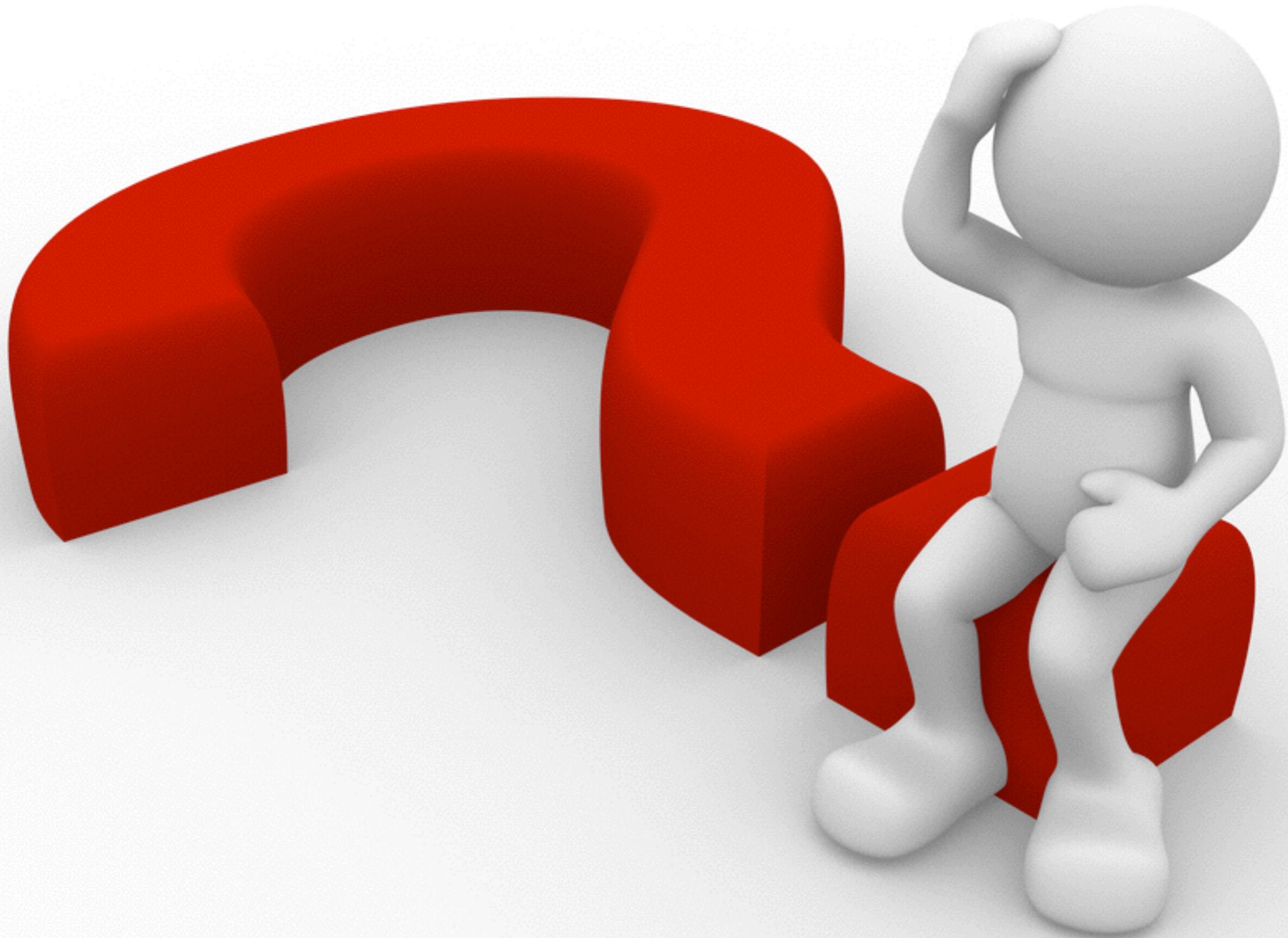
## Beispiel: RGB



# CIE-Normfarbtafel

## Beispiele







# Interaktive Medien

## Kapitel Bilder

### Kompression

# Farbtiefe

- **Farbtiefe** bestimmt Anzahl darstellbarer Farben
  - 2 Farben (1 Bit) - entspricht S/W
  - 256 Graustufen (8 Bit)
  - 16,7 Millionen Farben (24 Bit) - *True Color* (3 x 8 Bit pro Kanal (RGB))
  - ...

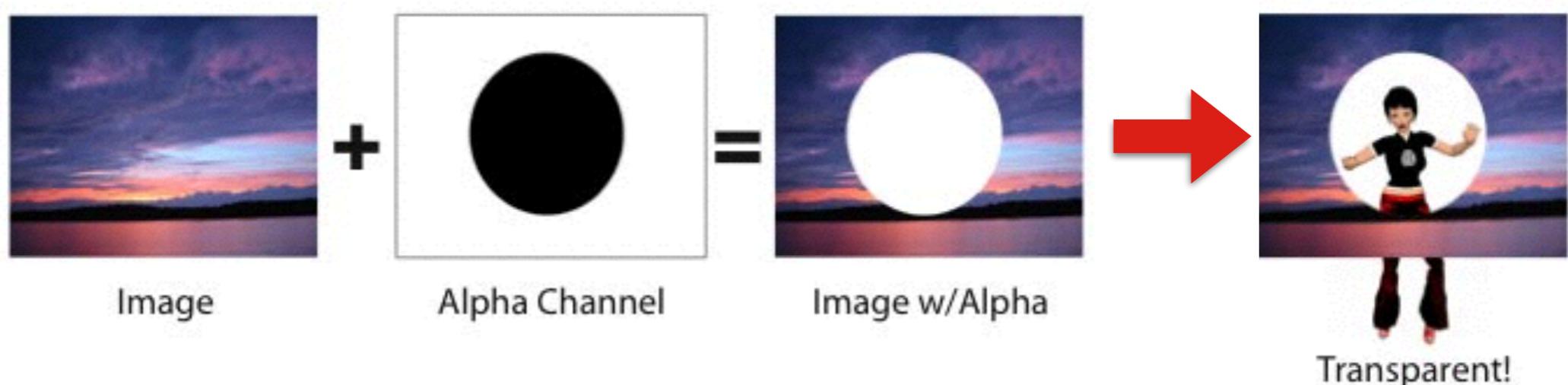
# Speicherbedarf

## Beispiele

	Auflösung			
Bits	800 x 600	1024 x 768	1280 x 800	1680 x 1050
1	60kB	98kB	128kB	220kB
8	480kB	786kB	1,0MB	1,8MB
16	960kB	1,5MB	2,0MB	3,4MB
24	1,4MB	2,3MB	3,1MB	5,3MB
32	1,9MB	3,1MB	4,1MB	7,0MB

# Alphakanal

- **Alphakanal** erlaubt Überblendung von Bildern
- **Alphakanal** gibt Transparenz des Pixels an
  - Beispiel: 24 Bit (RGB) + 8 Bit (Alpha)



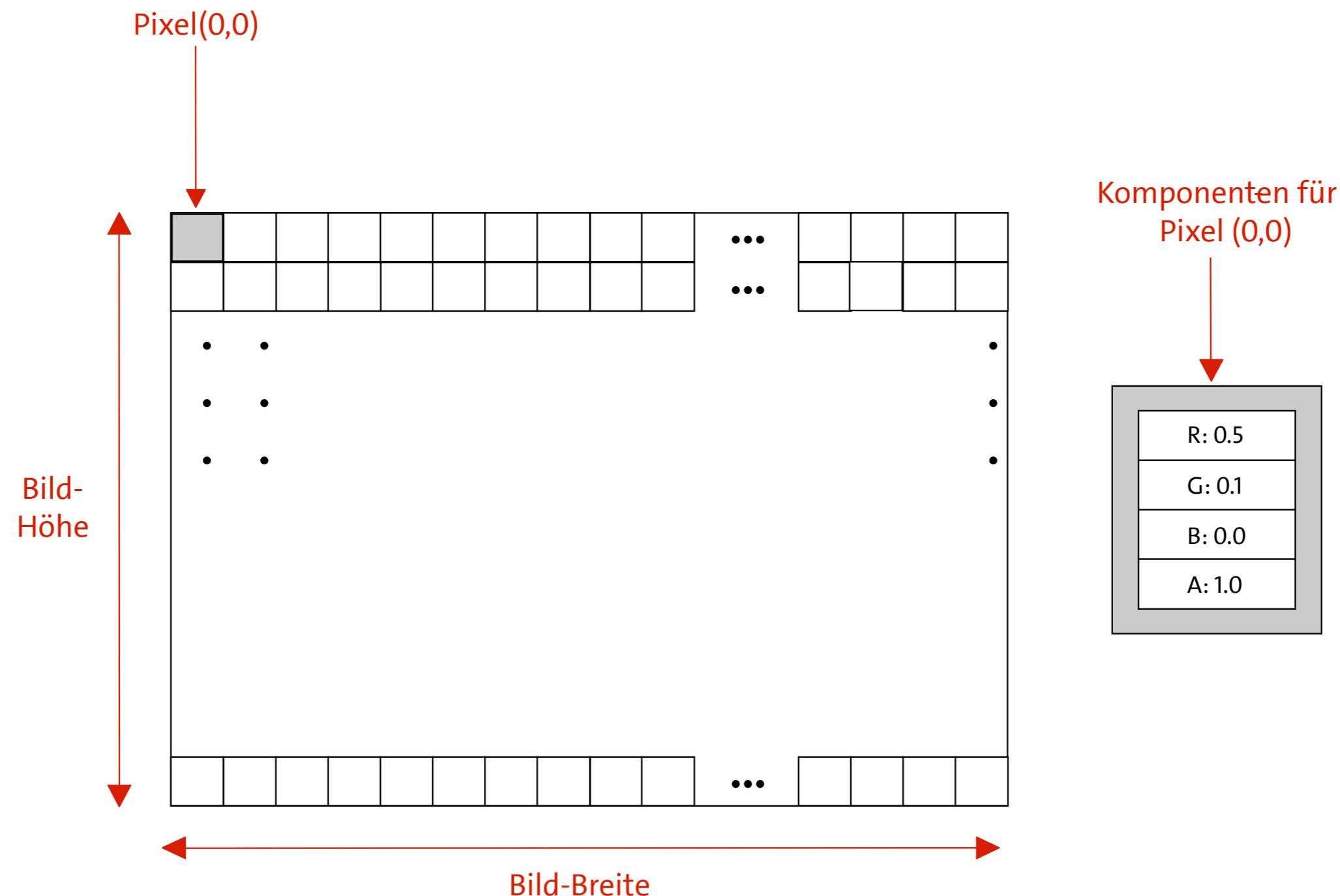
# Quantisierung

## Pixel

- Alternative Möglichkeiten zur Farbkodierung der Pixel
  1. Jeder Pixel beinhaltet Farbwert bezüglich des verwendeten Farbmodells
  2. Alle verwendeten Farben werden in Farbtabelle gespeichert und jeder Pixel enthält Verweis auf entsprechenden Eintrag

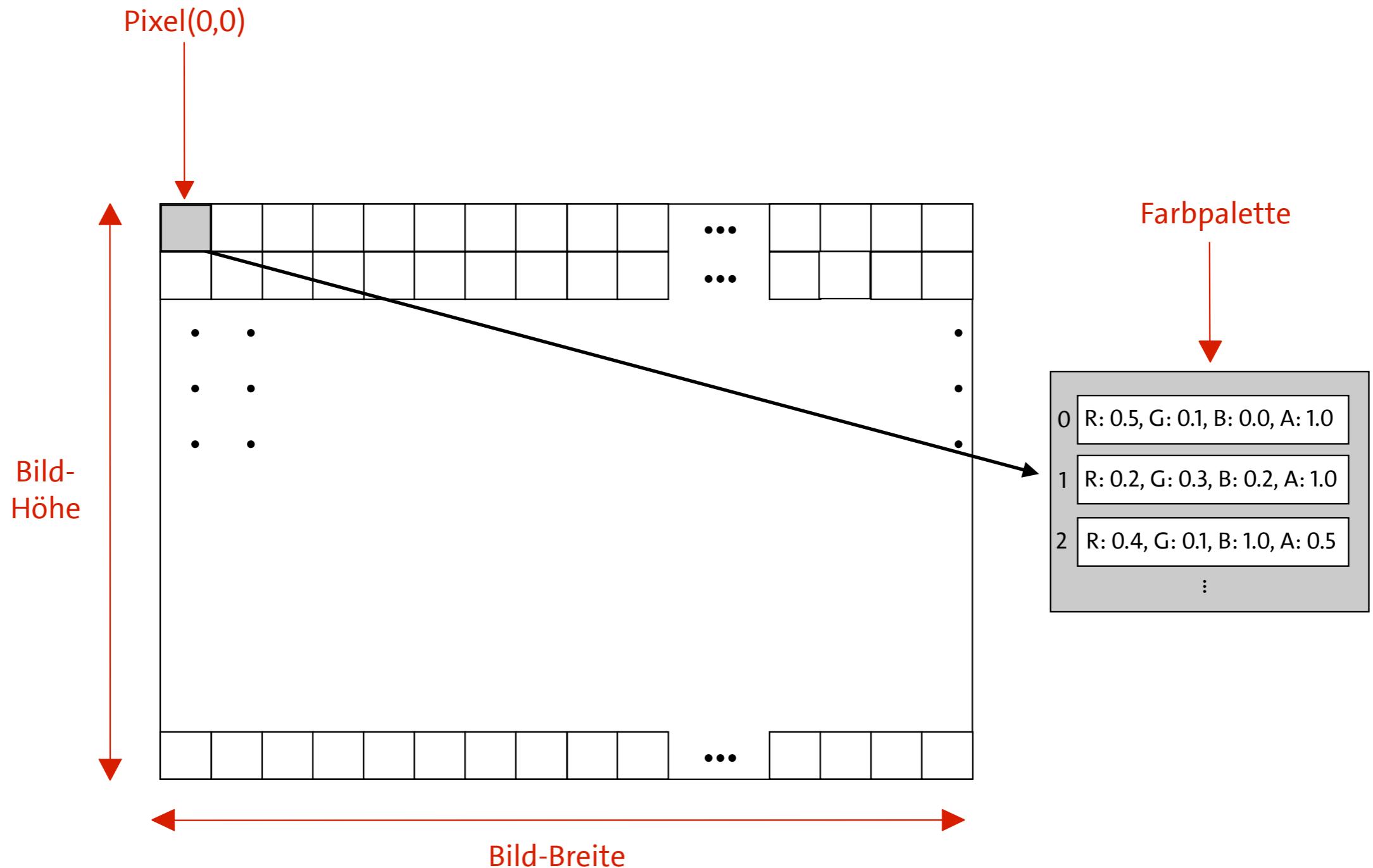
# Quantisierung

## Beispiel



# Quantisierung

## Beispiel



# Rastergrafiken

## Kompressionsverfahren

- Verlustfreie spezielle Verfahren
  - **Portable Network Graphics (PNG)**
  - **Graphics Interchange Format (GIF)**
  - ...
- Verlustbehaftete spezielle Verfahren
  - **Joint Photographic Experts Group (JPEG)**
  - ...

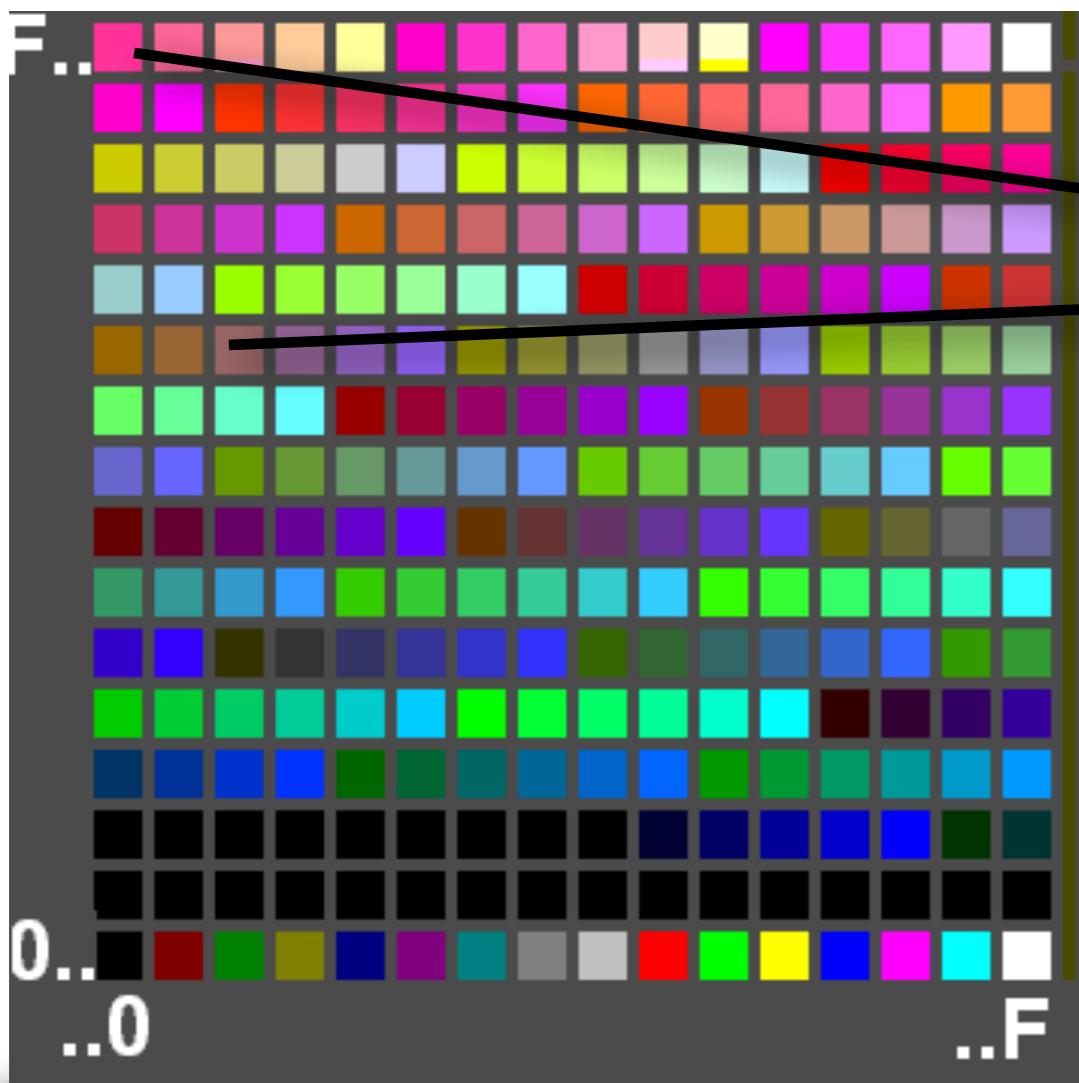
# Rastergrafikformate

## Beispiel: GIF

- **Graphics Interchange Format (GIF)** wurde 1987 von US-Online-Dienst CompuServe als Farbformat eingeführt
- GIF bietet verlustfreie Kompression für Bilder mit geringer Farbtiefe (8 Bit) und unterstützt 1-Bit Alphawert
- Farbinformationen werden in **Farbpalette** abgelegt

# GIF-Farbpalette

Beispiel: 8-Bit



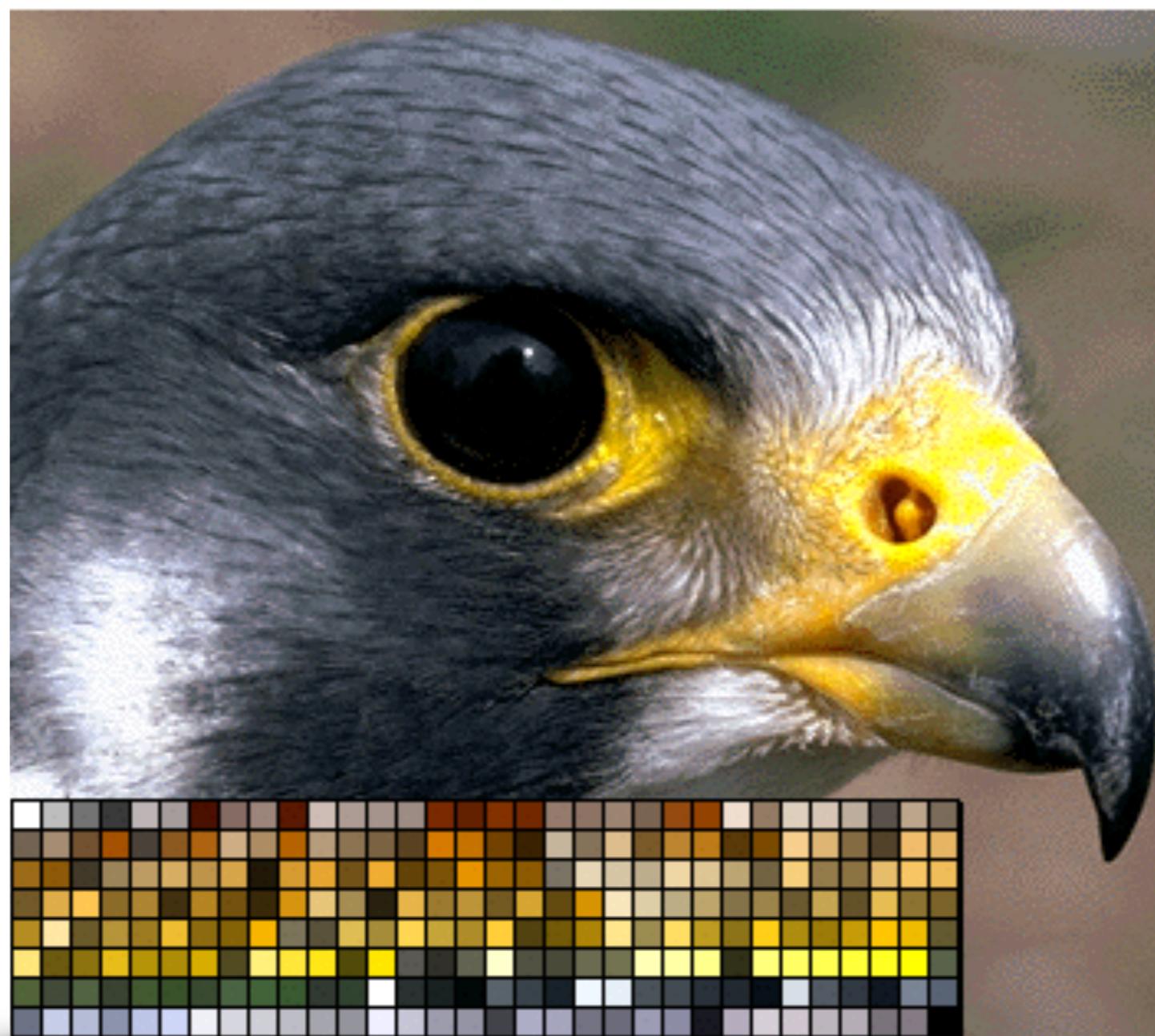
Farbpalette

0F	2A	35	A2
B3	B0	54	0A
BF	FF	0F	F0
12	22	C4	0C

Bild

# GIF-Farbpalette

## Beispiel



# GIF-Kompression

## Ablauf

- Farbinformationen werden horizontal von oben links bis unten rechts als Bitstrom gelesen und mittels LZW-Algorithmus (siehe VL *Interaktive Medien - Medien, Kanäle, Codes*) gespeichert

# Rastergrafikformate

- **Portable Network Graphics (PNG)**  
unterstützt neben unterschiedlichen Farbtiefen auch Transparenz
- PNG ist meistverwendete verlustfreie Grafikformat im Internet
- PNG nutzt prädiktive Kodierung mit **Vor-Filtern**

# PNG

## Vor-Filter

- **Vor-Filter** nutzen Korrelationen von benachbarten Farbwerte
- Bildpunkte werden über Differenz zu Nachbarpunkten beschrieben (**Delta-Kodierung**)
- resultierende Bits werden mit freier Variante des LZW-Algorithmus und Entropiekodierung nach Huffman kodiert

# PNG

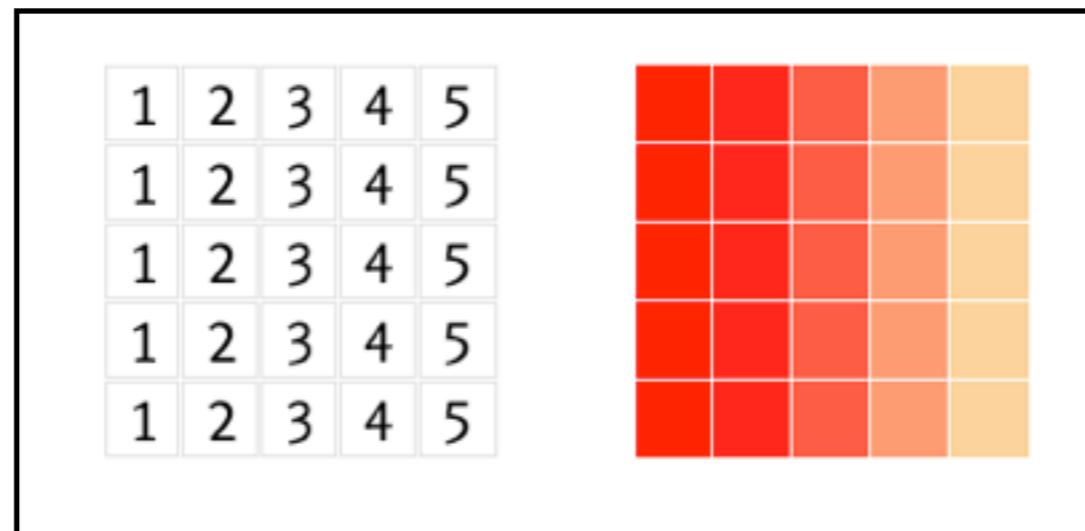
## Vor-Filter

- Zu jeder Bildzeile kann eine von fünf Filtermöglichkeiten bestimmt werden

Nummer	Name	Beschreibung
0	None	Keine Vorfilterung
1	Sub	Es werden Differenzen zu dem jeweils links benachbarten Pixel verarbeitet
2	Up	Es werden Differenzen zu dem jeweils darüber liegenden Pixel verarbeitet
3	Average	Es wird Differenz zum Mittelwert aus darüber liegenden und dem links benachbarten Pixel gebildet
4	Paeth	Es wird aus links benachbarten, dem darüber liegenden und schräg links oben benachbarten Pixel sogenannter <i>Paeth-Predictor</i> -Wert (links+oben-linksoben) berechnet, welcher Pixel für Differenzbildung bestimmt

# PNG

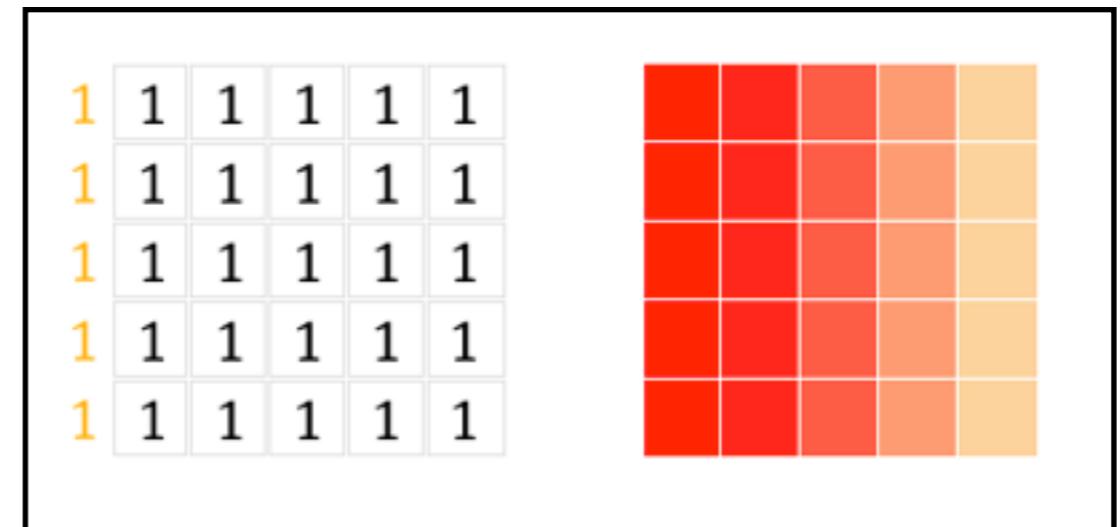
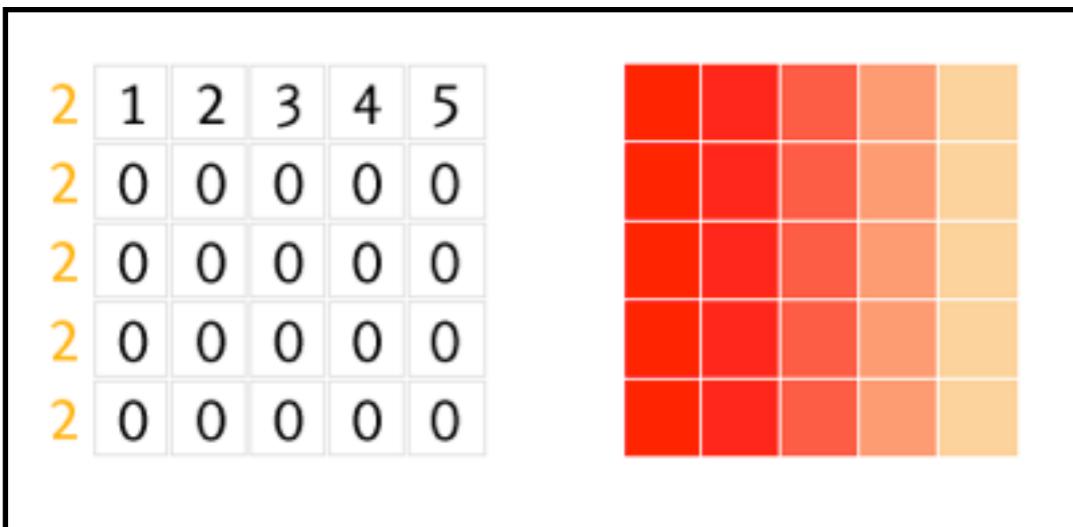
## Beispiel: Vor-Filter



2: Up



1: Sub



# Bildkompression

## Verlustbehaftet

- Menge der Daten eines digitalisierten Bildes hängt ab von
  - **Abtasthäufigkeit** (Diskretisierung, d.h. Anzahl der Pixel)
  - **Abtastgenauigkeit** (Quantisierung, d.h. Farbtiefe)

# Bildkompression

## Verlustbehaftet

- Kompression durch **Reduktion** der Anzahl der Abtastungen und/oder Genauigkeit der Abtastungen
- **Reduktion** allerdings nur an Stellen wo es visuell kaum wahrnehmbar ist

# Rastergrafikformate

## Beispiele



PNG: 47.775 Byte



GIF: 18.279 Byte



JPEG: 8.982 Byte

# JPEG-Kompression

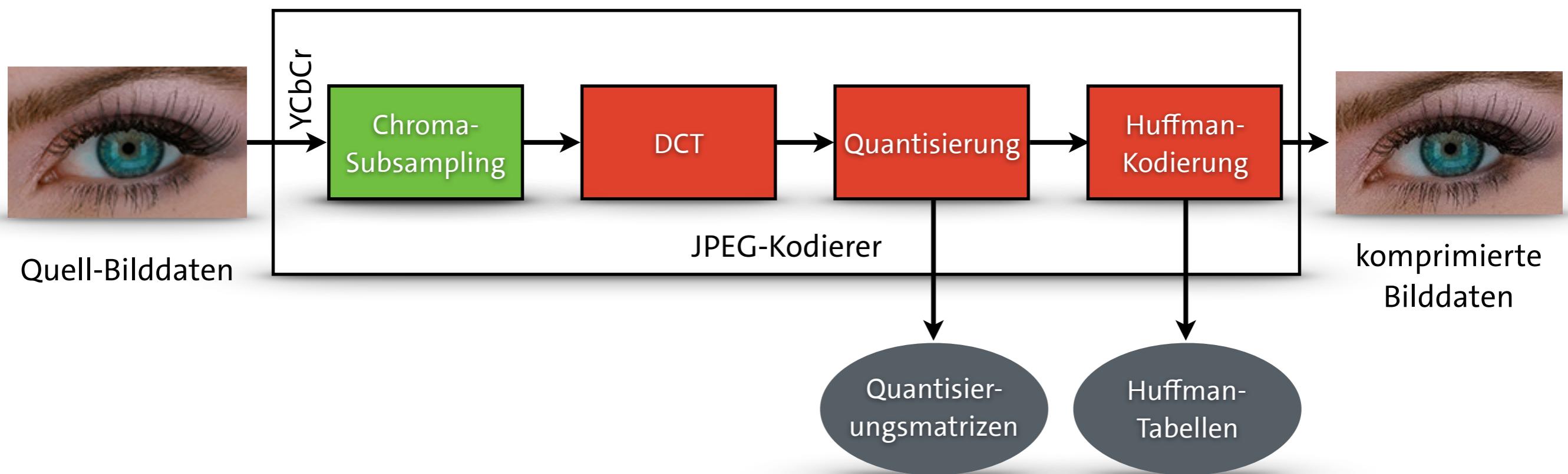
- **JPEG-Kompression** geht auf Gremium **Joint Photographic Experts Group (JPEG)** zurück und beschreibt verschiedene Methoden der Bildkompression
- JPEG ist gebräuchliche Bezeichnung für 1992 vorgestellte Norm

# JPEG-Kompression

- **JPEG-Kompression** geschieht nach Farbraumumrechnung ins YCbCr-Farbmodell in folgenden Schritten
  1. Chroma-Subsampling
  2. DCT
  3. Quantisierung
  4. Huffman-Kodierung

# JPEG-Kompression

## Algorithmus



# Chroma-Subsampling

- beim verlustbehafteten Kompressionsschritt  
**Chroma-Subsampling** wird Abtastrate für Teile des Bildes verringert
  - benachbarte ähnliche Pixel werden durch gleichen Farbwert repräsentiert
- Reduktion der Abtastrate wird beim Subsampling nur auf Farbkanäle ( $C_b$ ,  $C_r$ ), aber nicht auf Helligkeit ( $Y$ ) angewendet

# Chroma-Subsampling



Y - Helligkeit

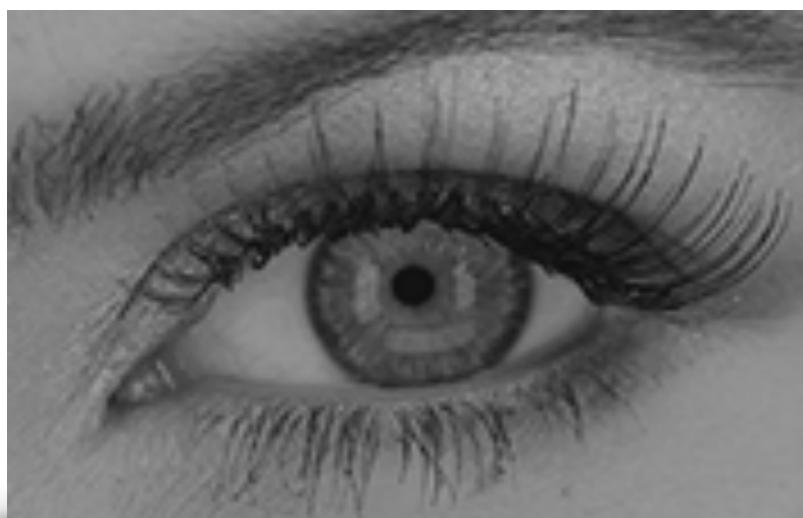


Cb - Farbkanal Blau/Gelb



Cr - Farbkanal Rot/Grün

# Chroma-Subsampling



Y - Helligkeit



Cb - Farbkanal Blau/Gelb



Cr - Farbkanal Rot/Grün

# Chroma-Subsampling



Y - Helligkeit



Cb - Farbkanal Blau/Gelb



Cr - Farbkanal Rot/Grün

# Chroma-Subsampling



Y - Helligkeit



Cb - Farbkanal Blau/Gelb

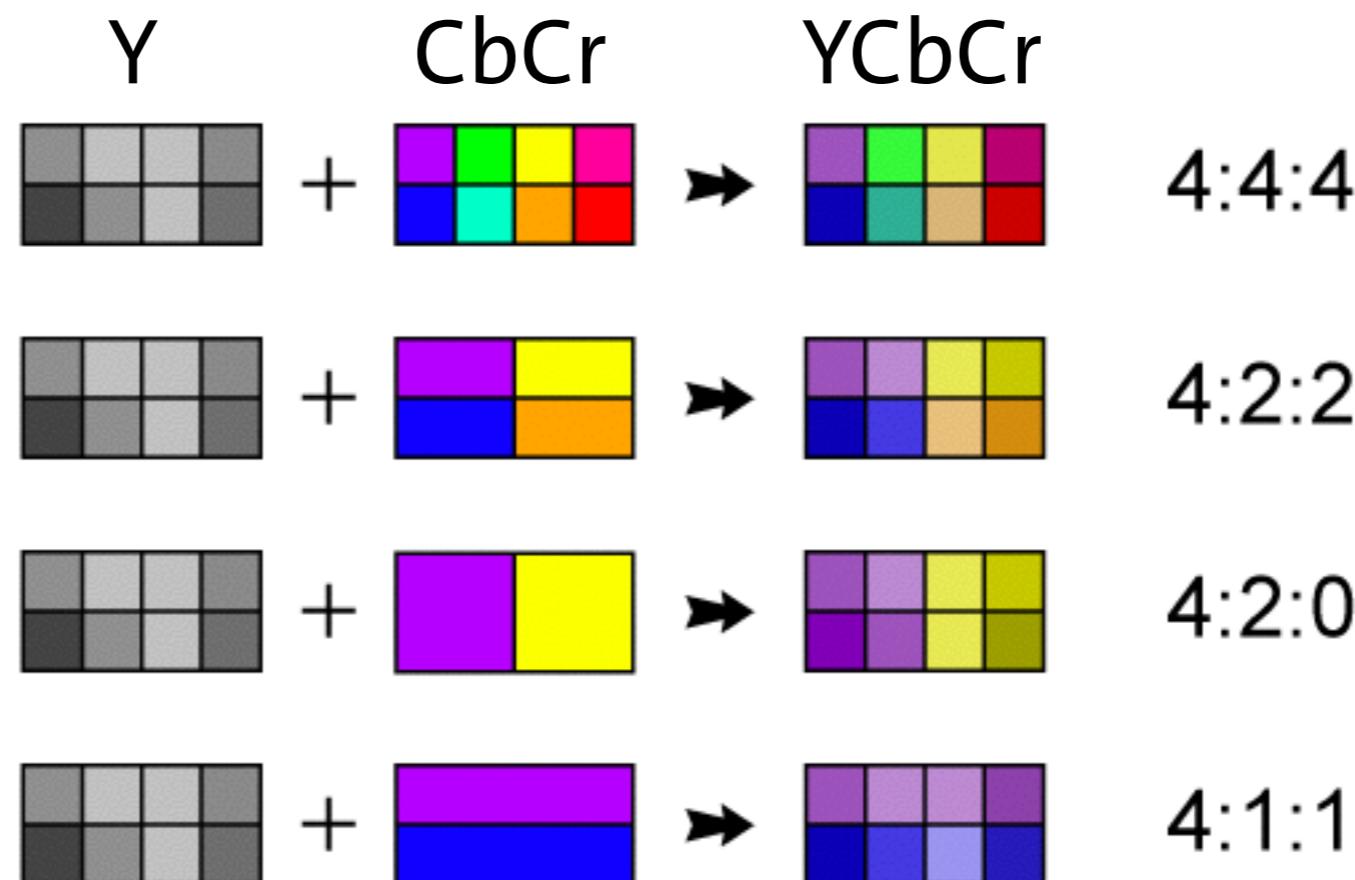


Cr - Farbkanal Rot/Grün

# Chroma-Subsampling

## Beispiele

- Unterabtastung der Farbsignale wird in Relation zum Helligkeitssignal beschreiben





A



B

In welchem Bild wurde die Farbauflösung halbiert?

A

B



A



B

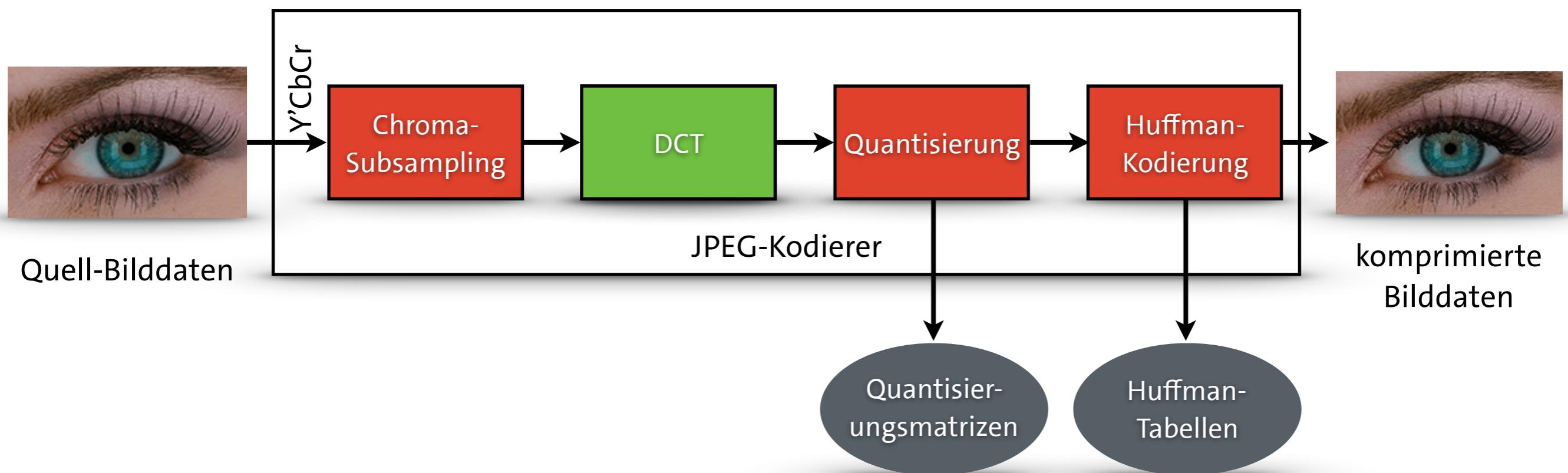
In welchem Bild wurde die Farbauflösung halbiert?

A

B

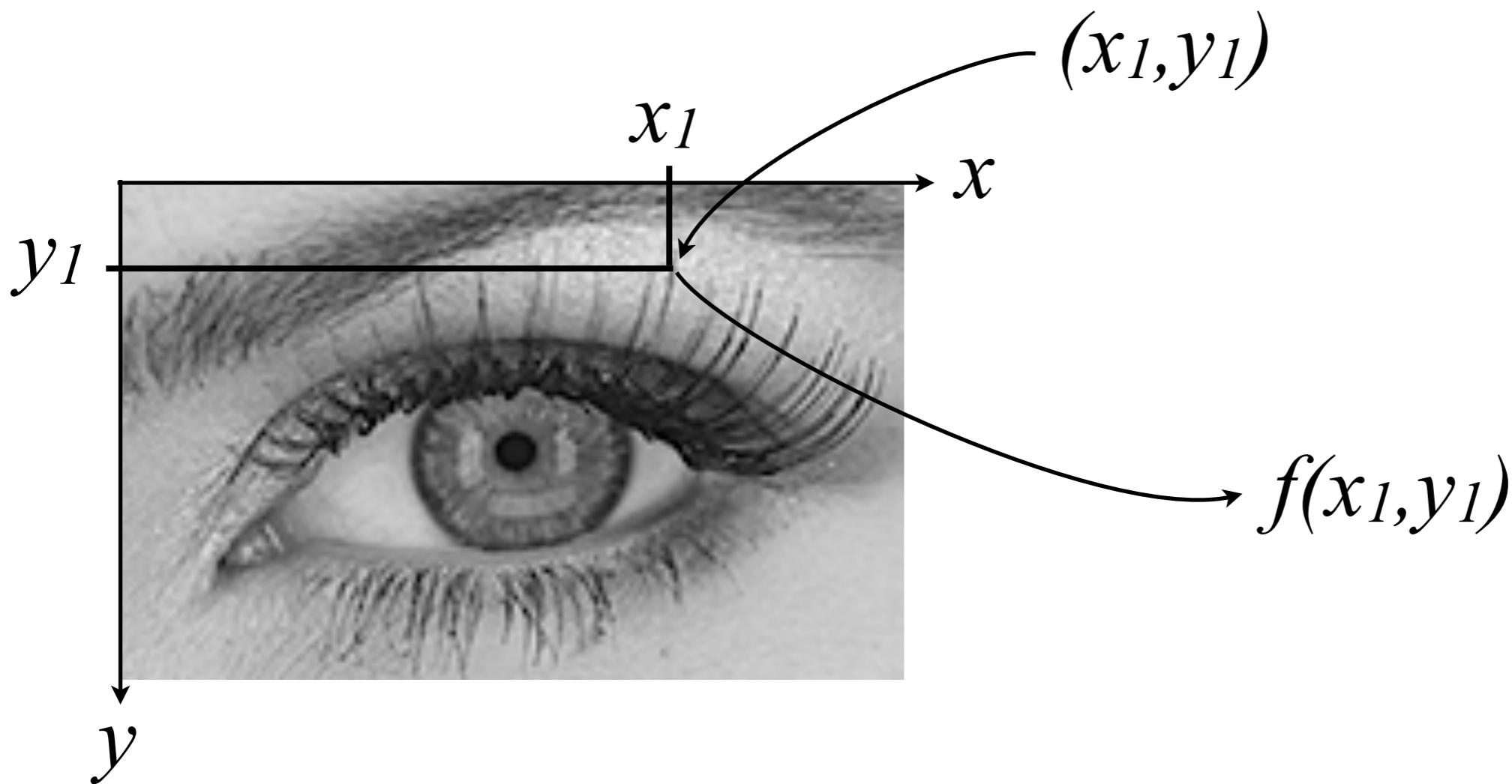
# JPEG-Kompression

## Algorithmus



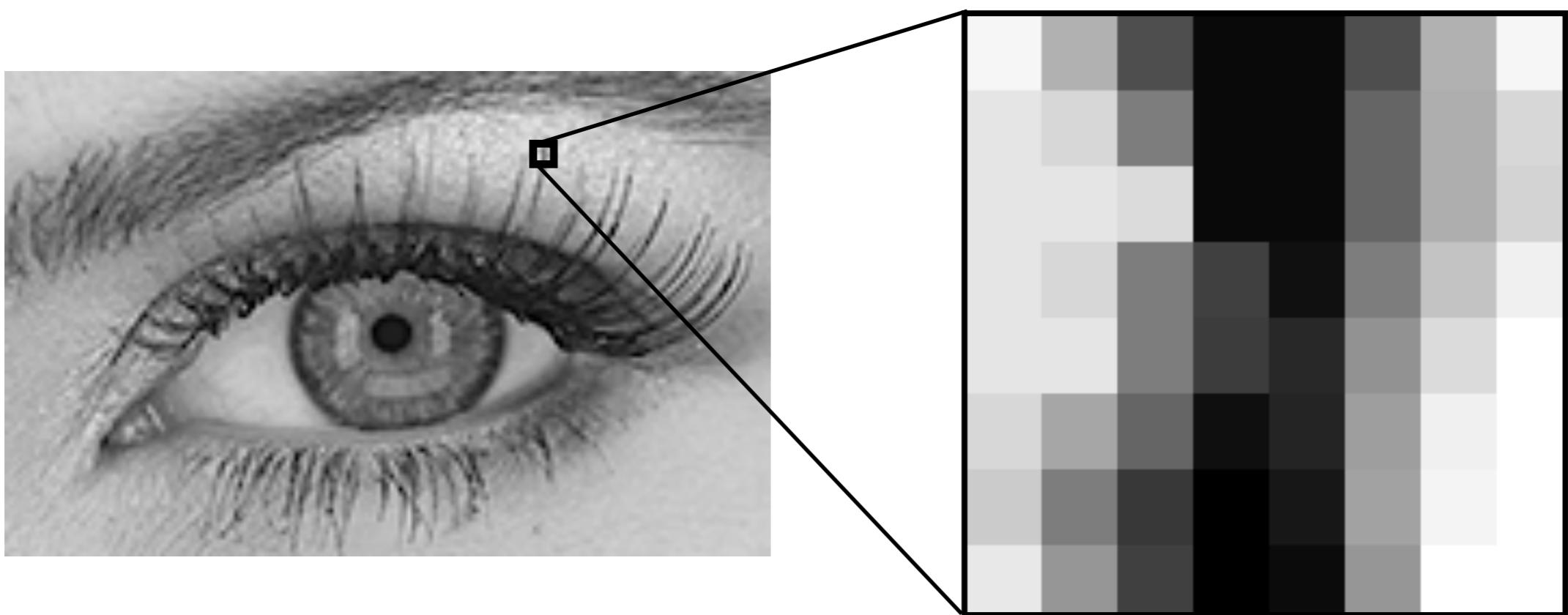
# Bildraum Illustration

- Wert  $f(x_1, y_1)$  für Pixel an Position  $(x_1, y_1)$



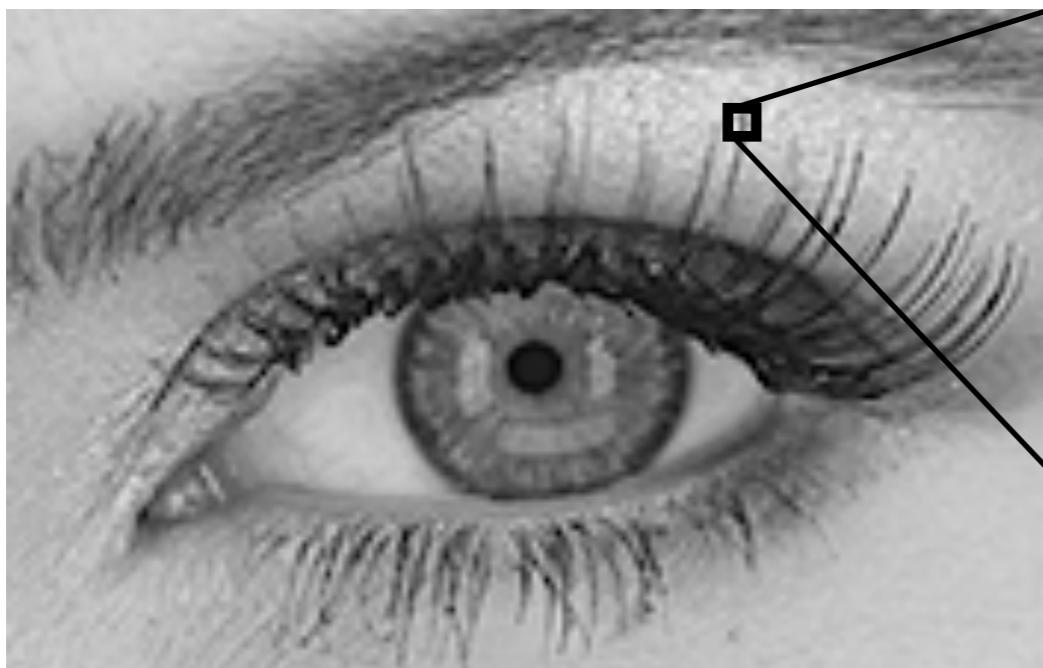
# Bildraum Illustration

- Unterteilung des Bildes in Blöcke aus jeweils 8x8-Pixel



# Bildraum Illustration

- Darstellung Wertebereich  $f(x_1, y_1)$  bei 8 Bit von -128 (schwarz) bis 127 (weiß)



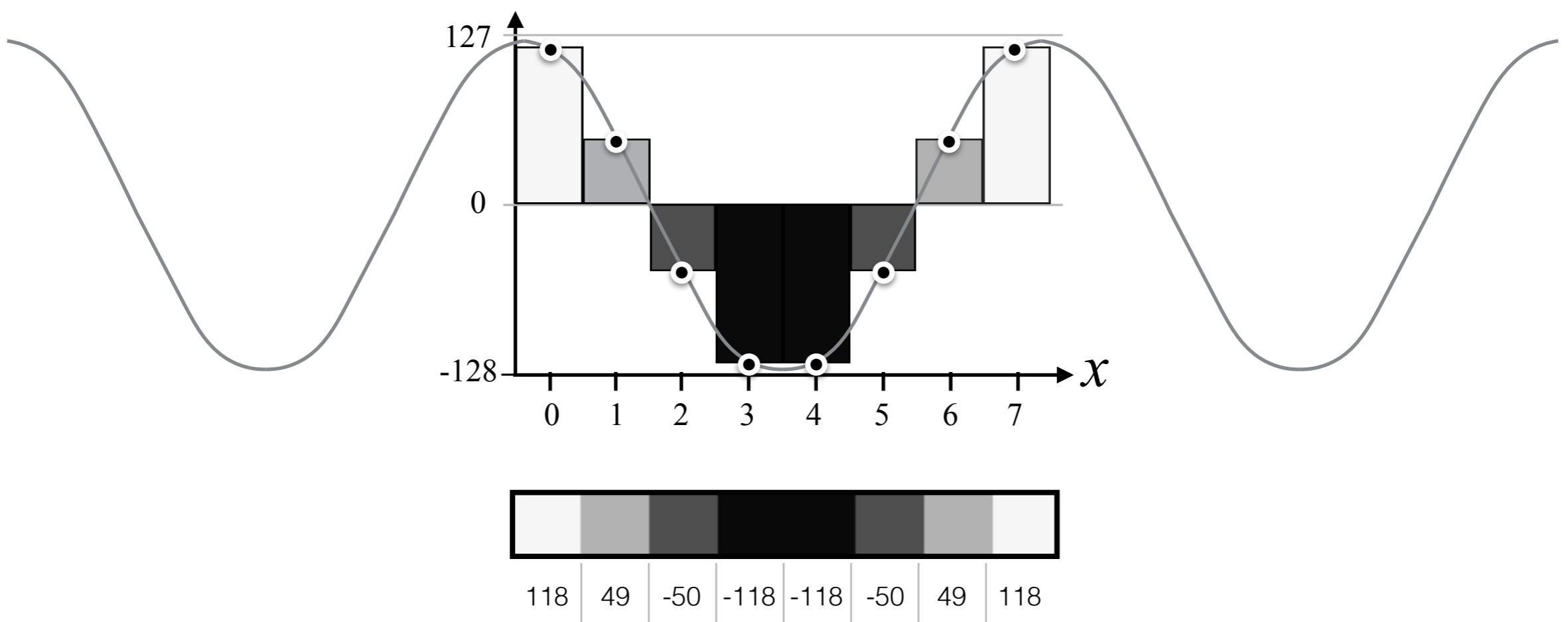
118	49	-50	-118	-118	-50	49	118
101	87	-3	-118	-118	-27	46	87
101	101	91	-118	-118	-27	46	83
101	87	-3	-64	-112	-3	67	112
101	101	-3	-68	-88	18	91	127
87	38	-27	-113	-92	30	112	127
75	-3	-72	-128	-105	34	116	127
104	22	-64	-128	-117	22	127	127

# Wellendarstellung

## Beispiel

- Helligkeitsverlauf darstellbar als Welle

$$f(x) = 127 \cdot \cos\left(\frac{(2x+1)2\pi}{16}\right)$$

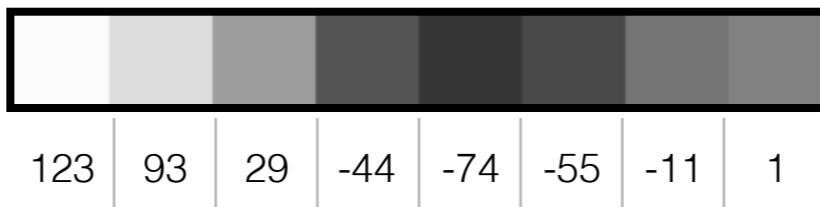
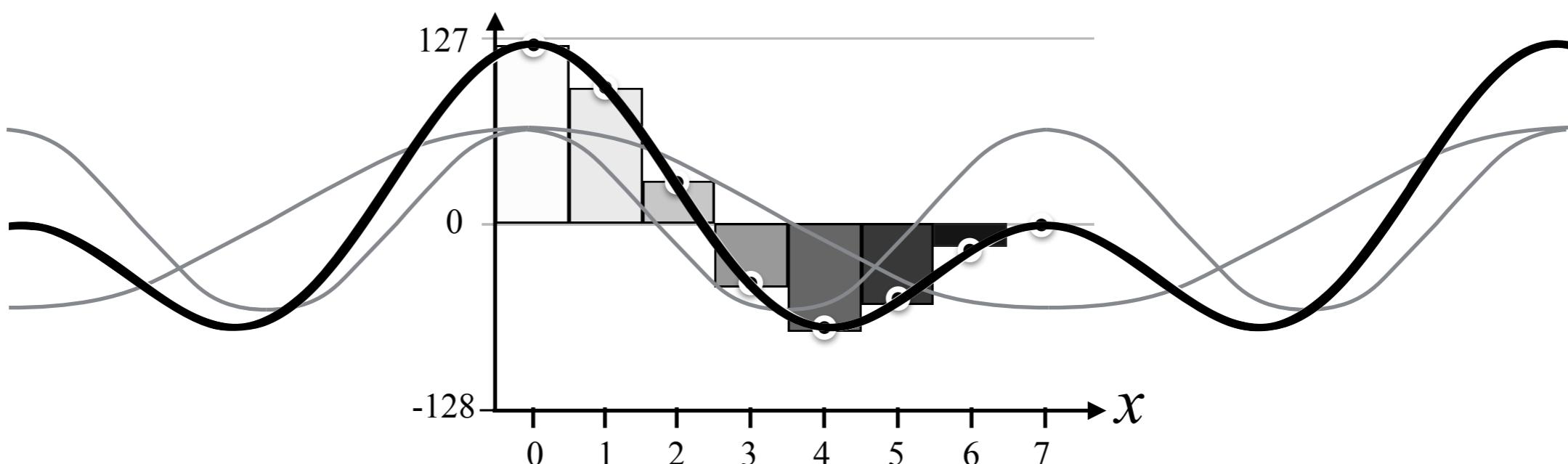


# Wellendarstellung

## Beispiel

- Überlagerung mehrerer Wellen

$$f(x) = \boxed{63 \cdot \cos\left(\frac{(2x+1)2\pi}{16}\right)} + 63 \cdot \cos\left(\frac{(2x+1)\pi}{16}\right)$$



# DCT

- **Diskrete Kosinustransformation (DCT)**  
beschreibt Verlauf von 8 Pixel durch  
Überlagerung von 8 Basisfrequenzen:



# DCT

- Den 8 Pixeln werden 8 neue Werte (**DCT-Koeffizienten**) zugeordnet, die angeben, mit welchem Anteil Basisfunktionen vorkommen

$$\begin{aligned} & F(0) \cdot \cos\left(\frac{(2x+1)0\pi}{16}\right) + F(1) \cdot \cos\left(\frac{(2x+1)1\pi}{16}\right) + F(2) \cdot \cos\left(\frac{(2x+1)2\pi}{16}\right) + F(3) \cdot \cos\left(\frac{(2x+1)3\pi}{16}\right) \\ & + F(4) \cdot \cos\left(\frac{(2x+1)4\pi}{16}\right) + F(5) \cdot \cos\left(\frac{(2x+1)5\pi}{16}\right) + F(6) \cdot \cos\left(\frac{(2x+1)6\pi}{16}\right) + F(7) \cdot \cos\left(\frac{(2x+1)7\pi}{16}\right) \end{aligned}$$

# DCT

- Darstellung als Summenformel:

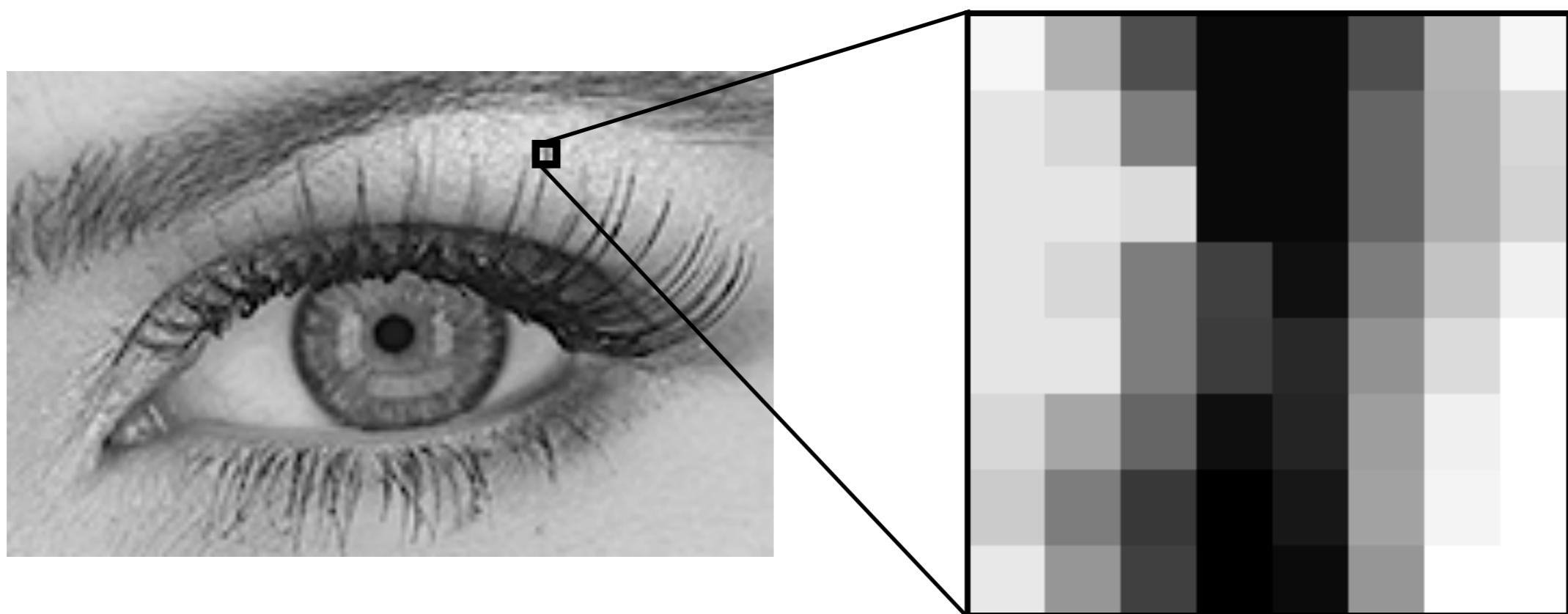
$$f(x) = \frac{1}{4} c_x \sum_{u=0}^7 F(u) \cdot \cos\left(\frac{(2x+1)u\pi}{16}\right), \quad c_k = \begin{cases} \frac{1}{\sqrt{2}}, & \text{falls } k=0 \\ 1, & \text{sonst} \end{cases}$$

- Berechnung der DCT-Koeffizienten  $F(u)$ :

$$F(u) = \frac{1}{4} c_u \sum_{x=0}^7 f(x) \cdot \cos\left(\frac{(2x+1)u\pi}{16}\right), \quad c_k = \begin{cases} \frac{1}{\sqrt{2}}, & \text{falls } k=0 \\ 1, & \text{sonst} \end{cases}$$

# 2-D DCT

- Erweiterung der DCT auf zwei Dimensionen für Blöcke aus 8x8 Pixel



# Berechnung der 2D-DCT

- Berechnung der 64 DCT-Koeffizienten  $F(u, v)$  aus 64 Pixeln  $f(x, y)$ :

$$F(u, v) = \frac{1}{4} c_u c_v \sum_{y=0}^7 \sum_{x=0}^7 f(x, y) \cdot \cos \left( \frac{(2x+1)u\pi}{16} \right) \cdot \cos \left( \frac{(2y+1)v\pi}{16} \right)$$

$$c_k = \begin{cases} \frac{1}{\sqrt{2}}, & \text{falls } k = 0 \\ 1, & \text{sonst} \end{cases}$$

# Berechnung der 2D-DCT

- $F(u, v)$  effizient zu berechnen, da  $\cos$ - und  $c_k$ -Anteile unabhängig von  $f(x, y)$

$$F(u, v) = \sum_{x=0}^7 \sum_{y=0}^7 T(x, y, u, v) \cdot f(x, y)$$

- Parameter  $T(x, y, u, v)$  vorberechenbar:

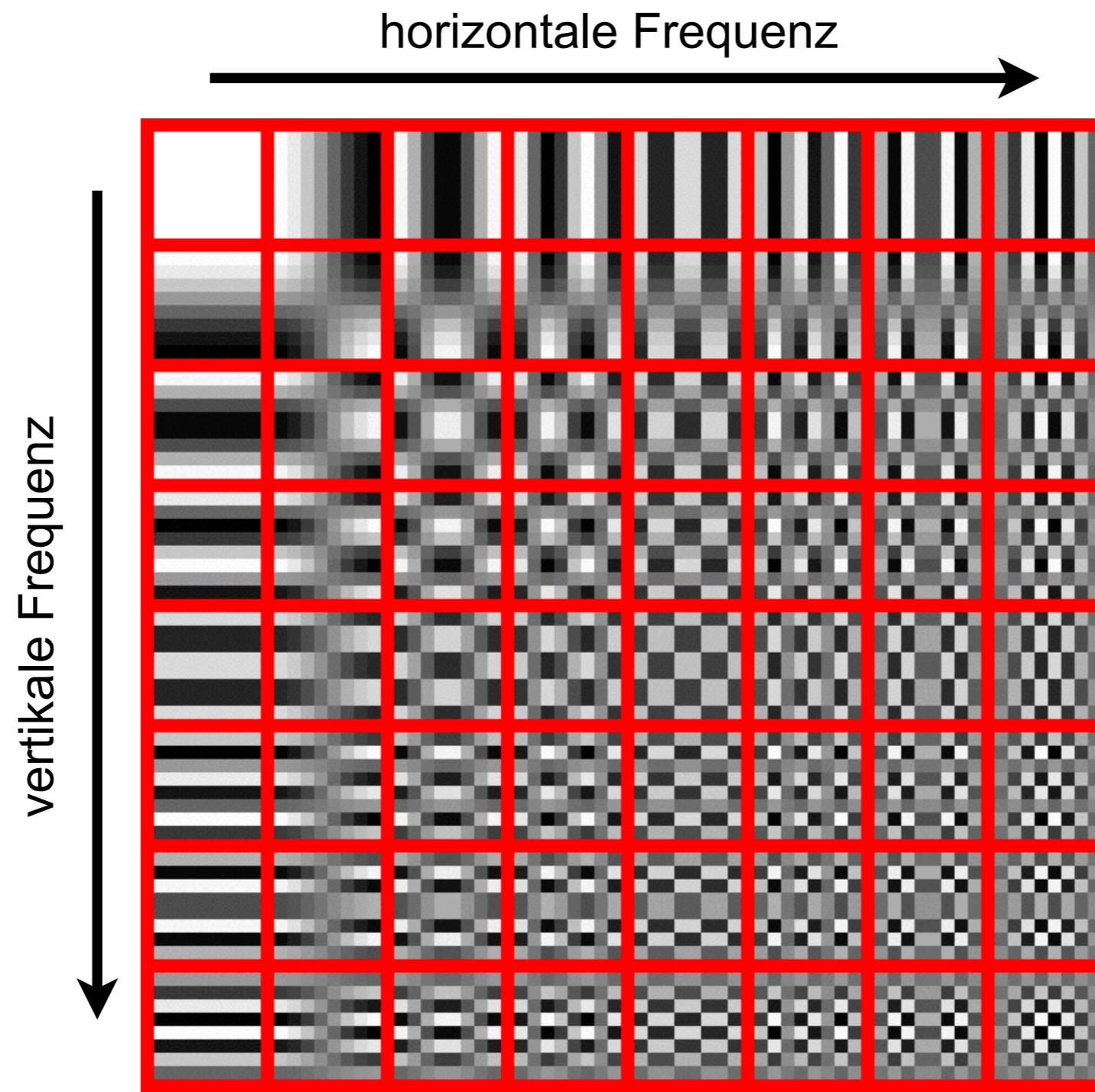
$$T(x, y, u, v) = \frac{1}{4} c_u c_v \cos\left(\frac{(2x+1)u\pi}{16}\right) \cdot \cos\left(\frac{(2y+1)v\pi}{16}\right)$$

# Bildrekonstruktion

- $F(u, v)$  gibt an, wie stark Muster aus horizontalen und vertikalen Frequenzen in Teilbild vorkommen
  - Rekonstruktion eines Bildes aus DCT-Koeffizienten durch Zusammensetzung der Basismuster
- Inverse Transformation mittels  $T(x, y, u, v)$

$$f(x, y) = \sum_{u=0}^7 \sum_{v=0}^7 T(x, y, u, v) \cdot F(u, v)$$

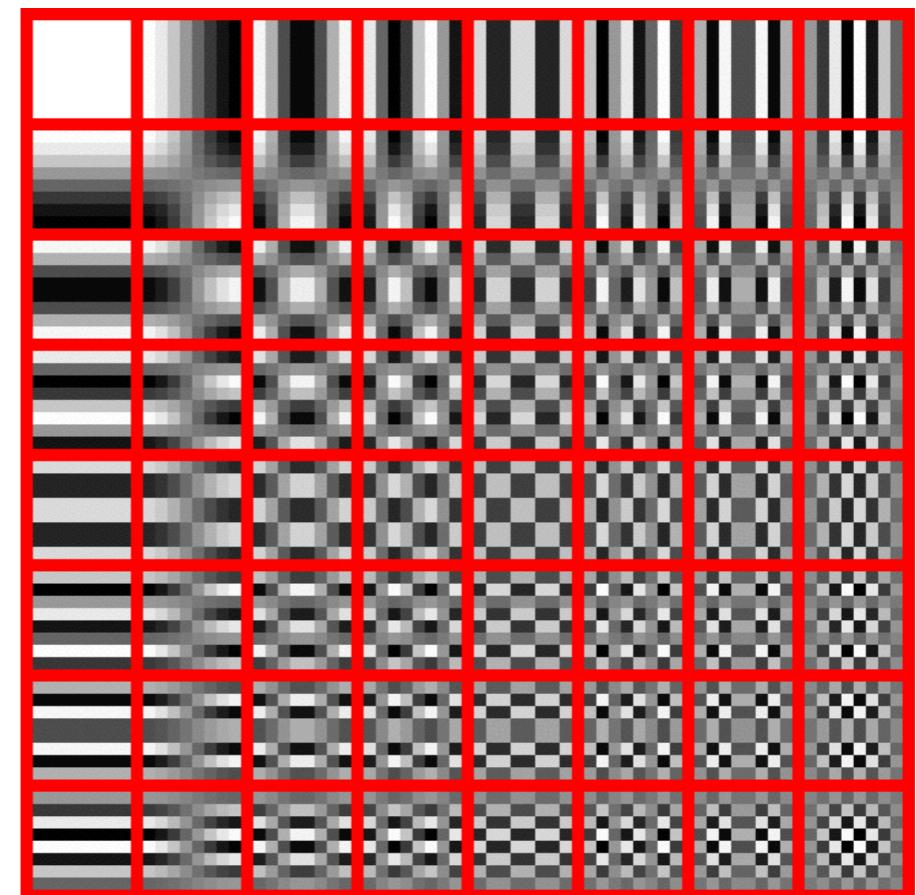
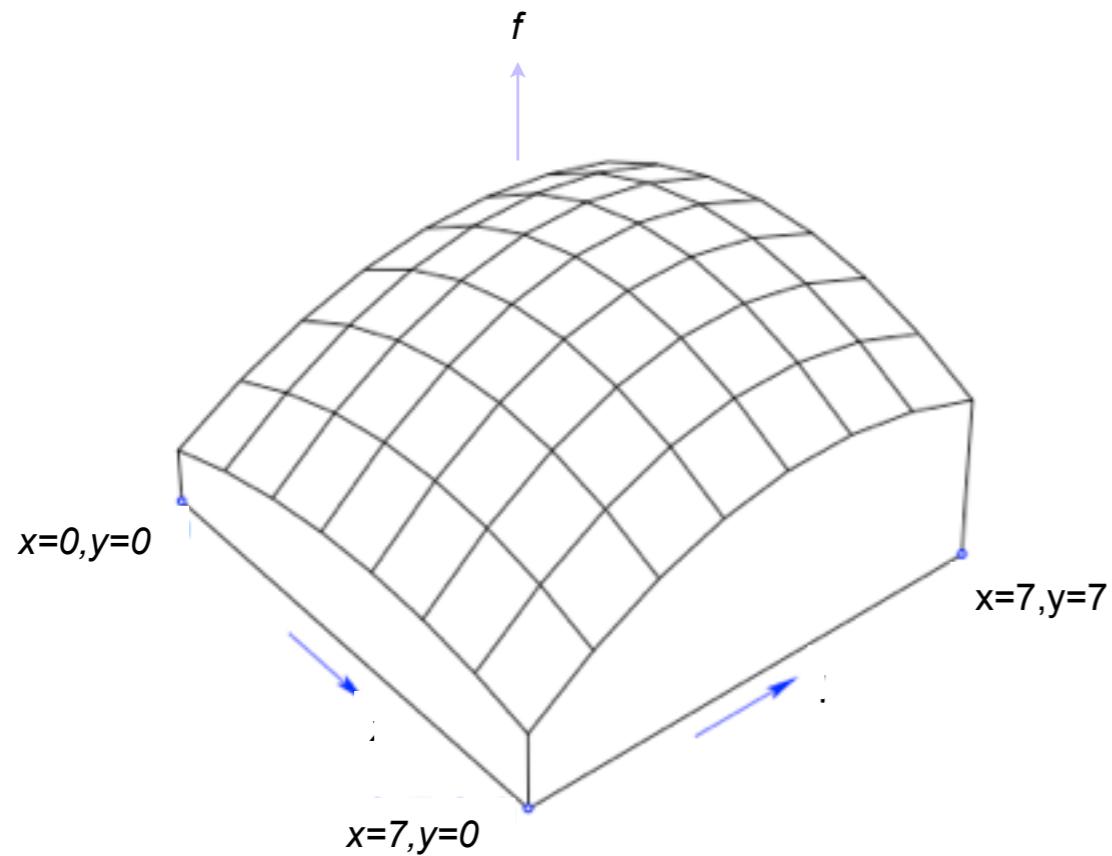
# 64 Basismuster



# Frequenztransformation

## Beispiel

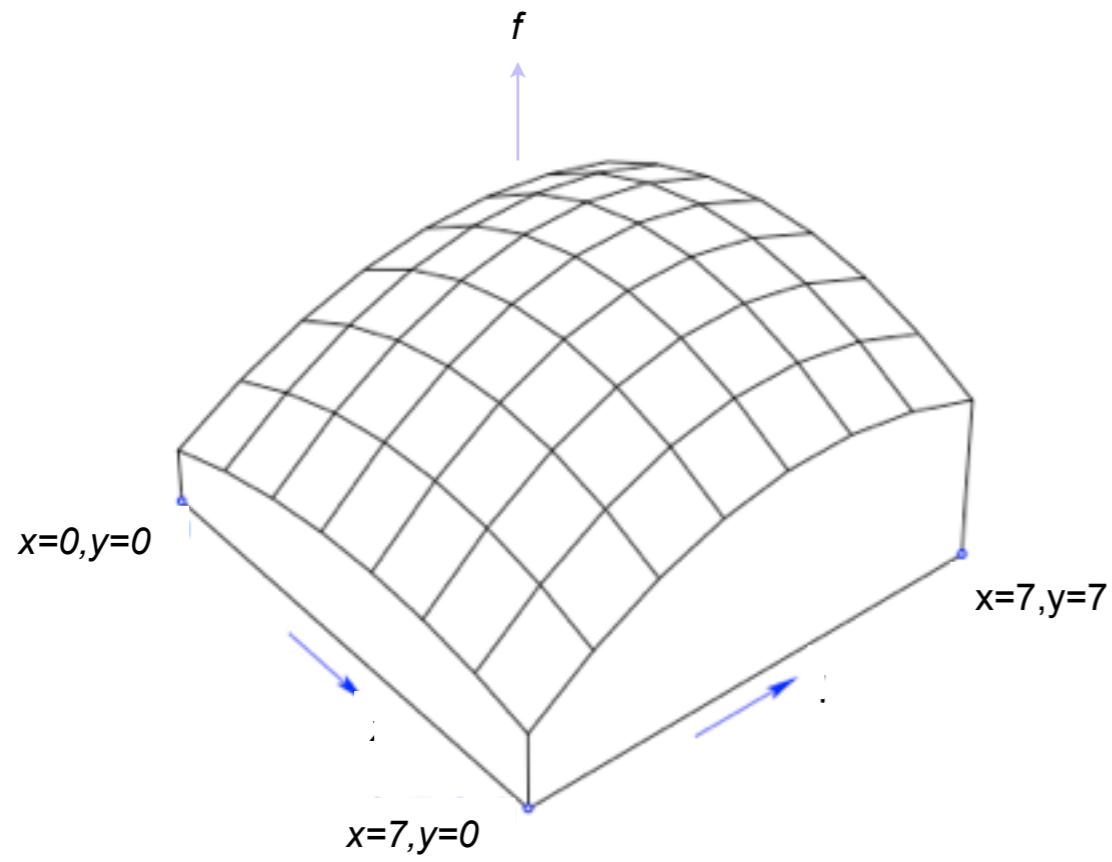
Funktionswerte  $f(x,y)$



# Frequenztransformation

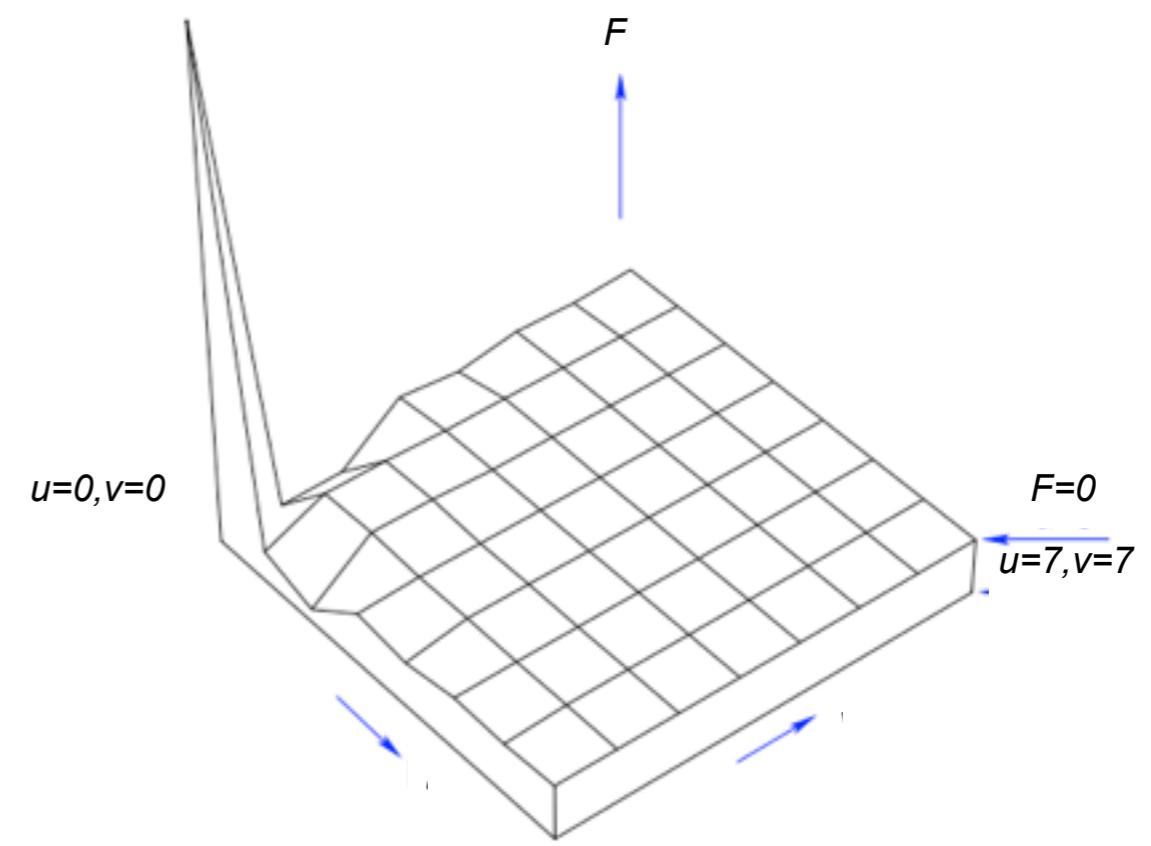
## Beispiel

Funktionswerte  $f(x,y)$



Ortsraum

DCT-Koeffizienten  $F(u,v)$



Frequenzraum

# Berechnung der DCT

## Beispiel

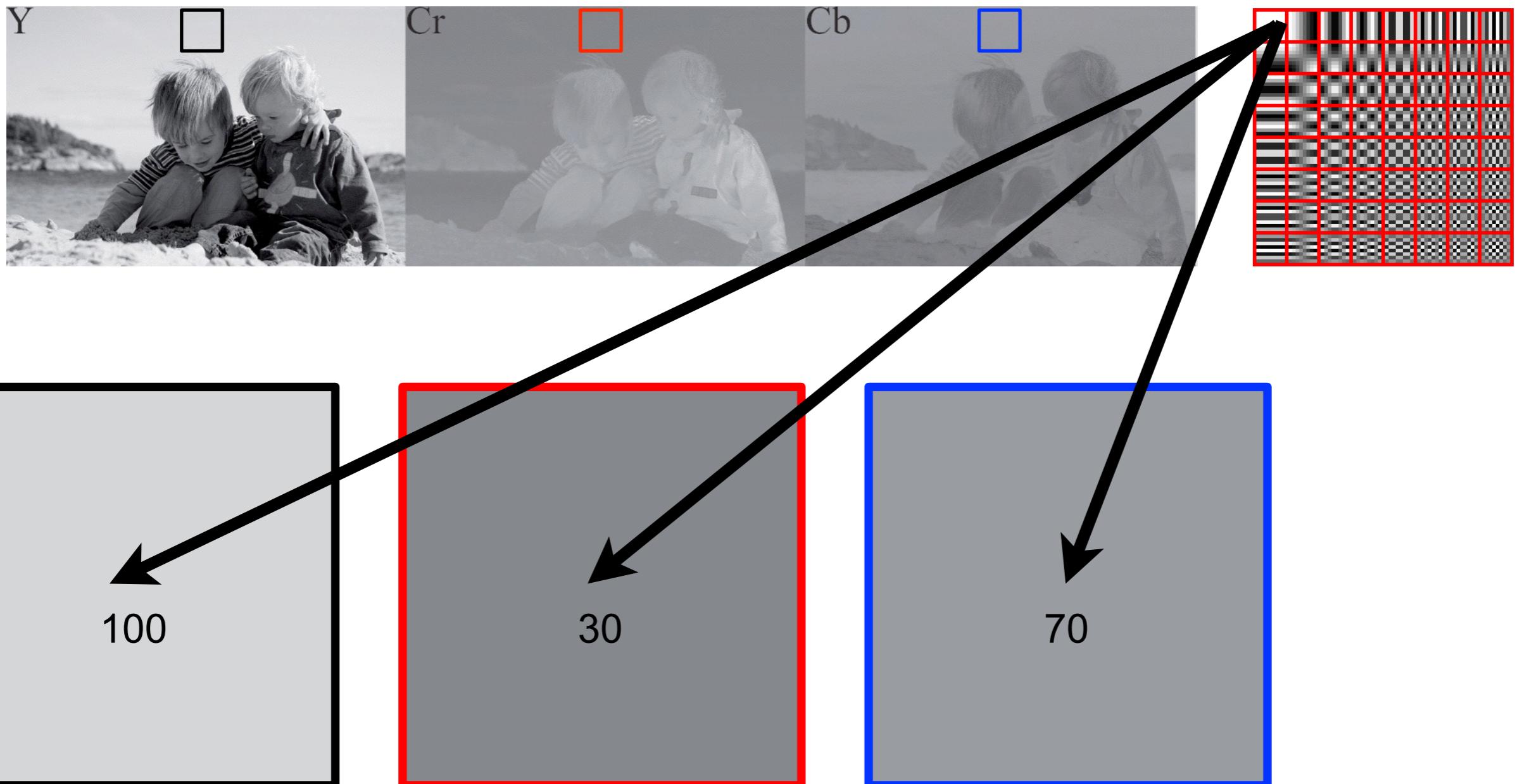
- 8x8 Pixel aus beliebigen Kanal (YCrCb)

$$f(0,0) \rightarrow \begin{bmatrix} 10 & 15 & 25 & 37 & 51 & 66 & 82 & 100 \\ 15 & 19 & 28 & 39 & 52 & 67 & 83 & 101 \\ 25 & 28 & 35 & 45 & 58 & 72 & 88 & 105 \\ 37 & 39 & 45 & 54 & 66 & 79 & 94 & 111 \\ 51 & 52 & 58 & 66 & 76 & 89 & 103 & 119 \\ 66 & 67 & 72 & 79 & 89 & 101 & 114 & 130 \\ 82 & 83 & 88 & 94 & 103 & 114 & 127 & 142 \\ 100 & 101 & 105 & 111 & 119 & 130 & 142 & 156 \end{bmatrix}$$

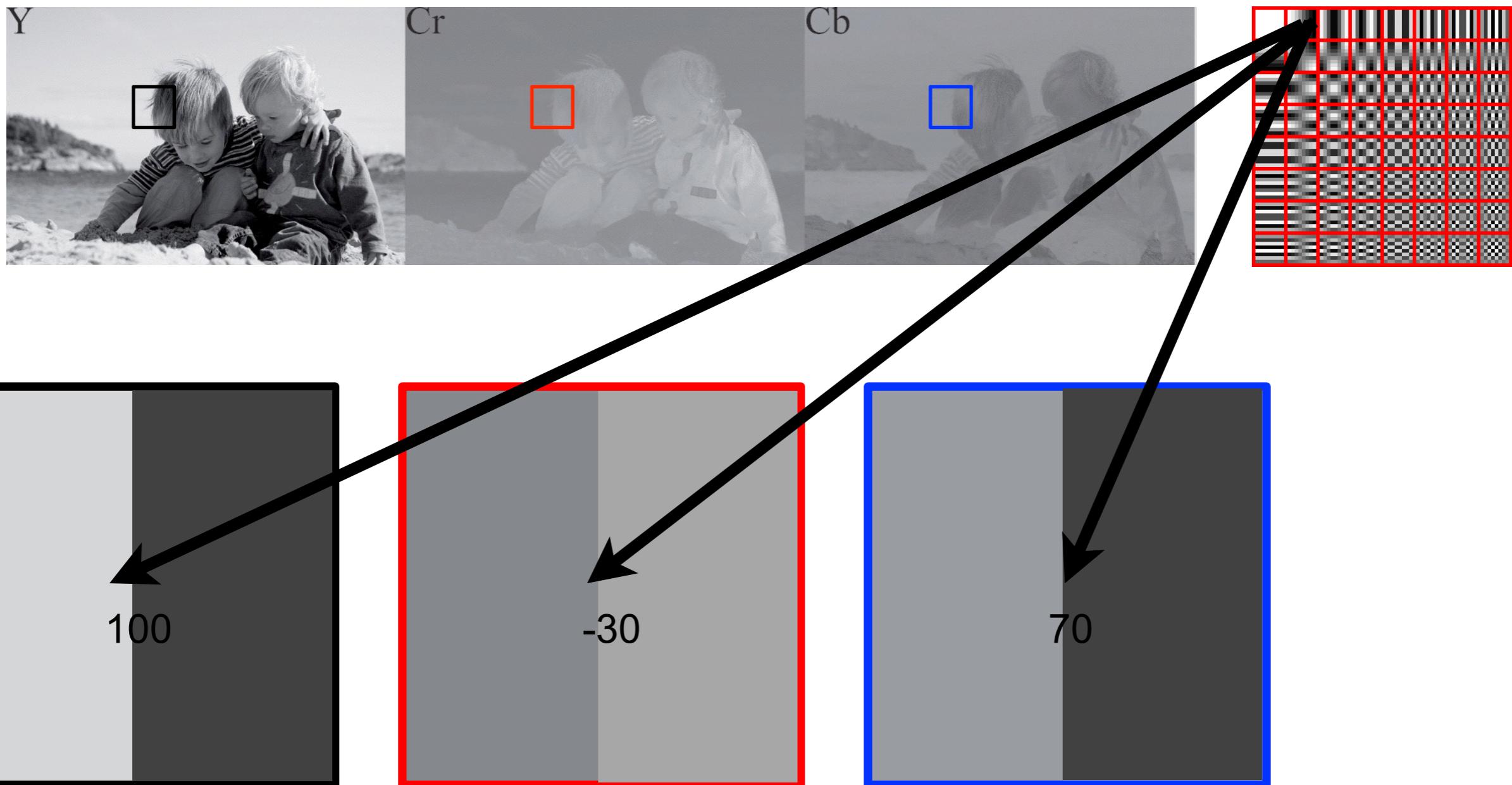
- Berechnung der  $F(u,v)$

$$F(0,0) \rightarrow \begin{bmatrix} 782,91 & 44,93 & 172,52 & -35,28 & -20,58 & 35,93 & 2,88 & -3,85 \\ -122,35 & -75,46 & -7,52 & 55,00 & 30,72 & -17,73 & 8,29 & 1,97 \\ -2,99 & -32,77 & -57,18 & -30,07 & 1,76 & 17,63 & 12,23 & -13,57 \\ -7,98 & 0,66 & 2,41 & -21,28 & -31,07 & -17,20 & -9,68 & 16,94 \\ 3,87 & 7,07 & 0,56 & 5,13 & -2,47 & -15,09 & -17,70 & -3,76 \\ -3,77 & 0,80 & -1,46 & -3,50 & 1,48 & 4,13 & -6,32 & -18,47 \\ 1,78 & 3,28 & 4,63 & 3,27 & 2,39 & -2,31 & 5,21 & 11,77 \\ -1,75 & 0,43 & -2,72 & -3,05 & 3,95 & -1,83 & 1,98 & 3,87 \end{bmatrix}$$

# DCT Veranschaulichung

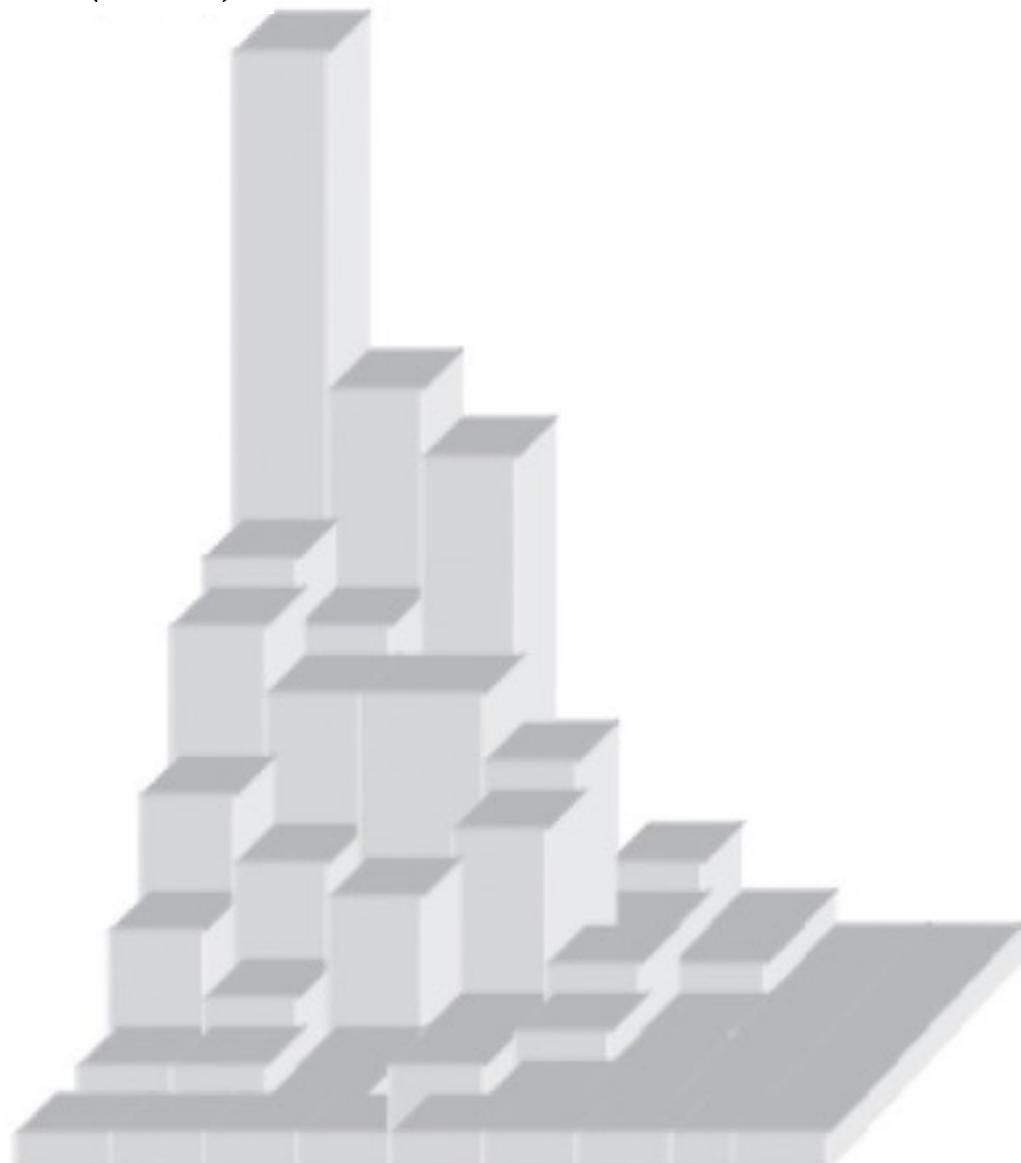


# DCT Veranschaulichung



# Berechnung der DCT

$F(0,0)$



$F(7,7)$

- **DC-Komponente**  $F(0,0)$  ist Maß für Mittelwert aller Pixel innerhalb eines Teilbildes (siehe Übungen)
- anderen Komponenten  $F(u,v)$  mit  $u,v \neq 0$  werden **AC-Komponenten** genannt

# Berechnung der DCT

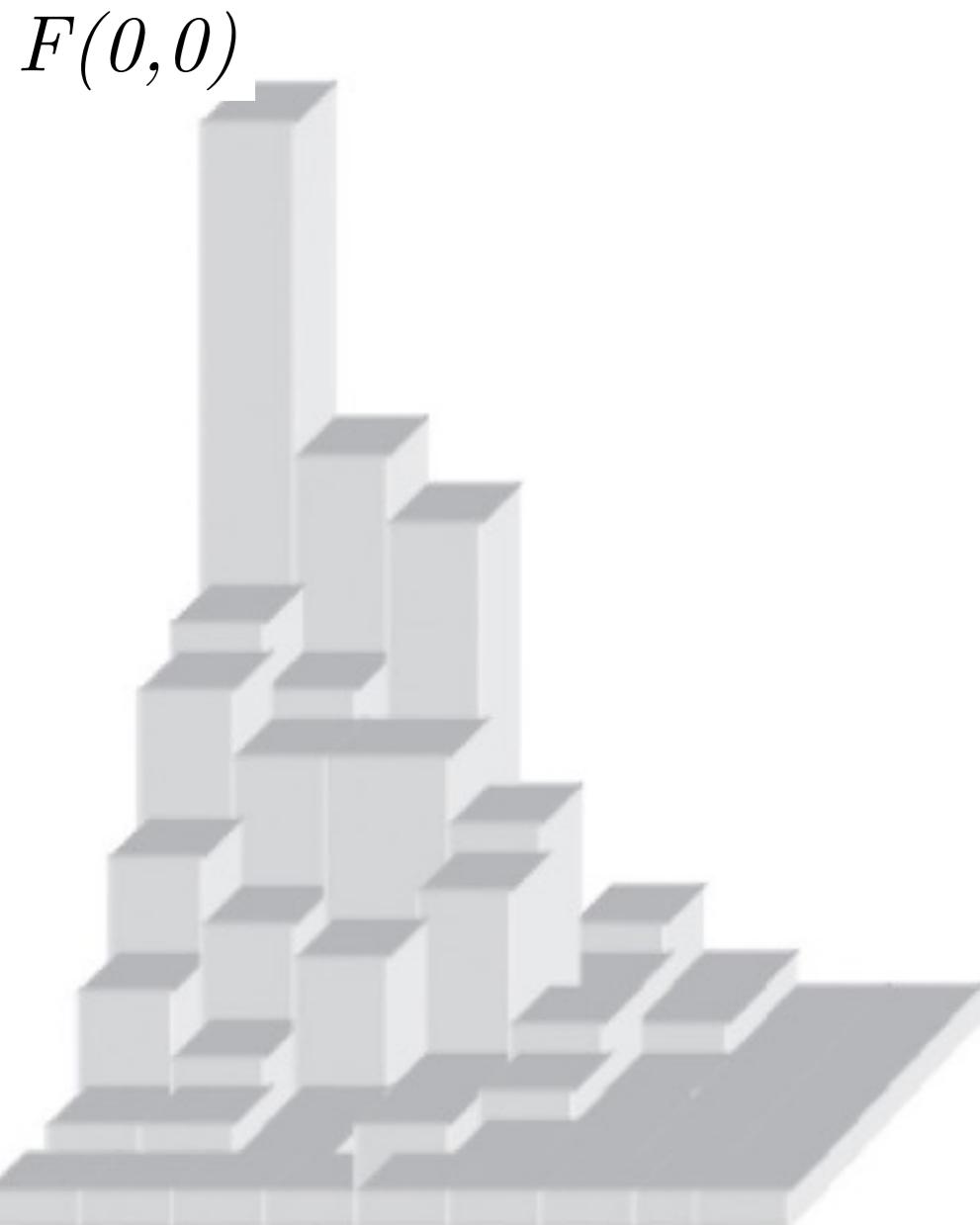
- DCT reduziert Speicherplatzbedarf nicht, da statt 64 Pixelwerte 64 Koeffizienten gespeichert werden
- Aber:
  - hohe Frequenzanteile sind oft weniger stark vorhanden als niedrigere
  - hohe Frequenzen spielen bei Wahrnehmung des Bildes kaum eine Rolle

# Diskussion

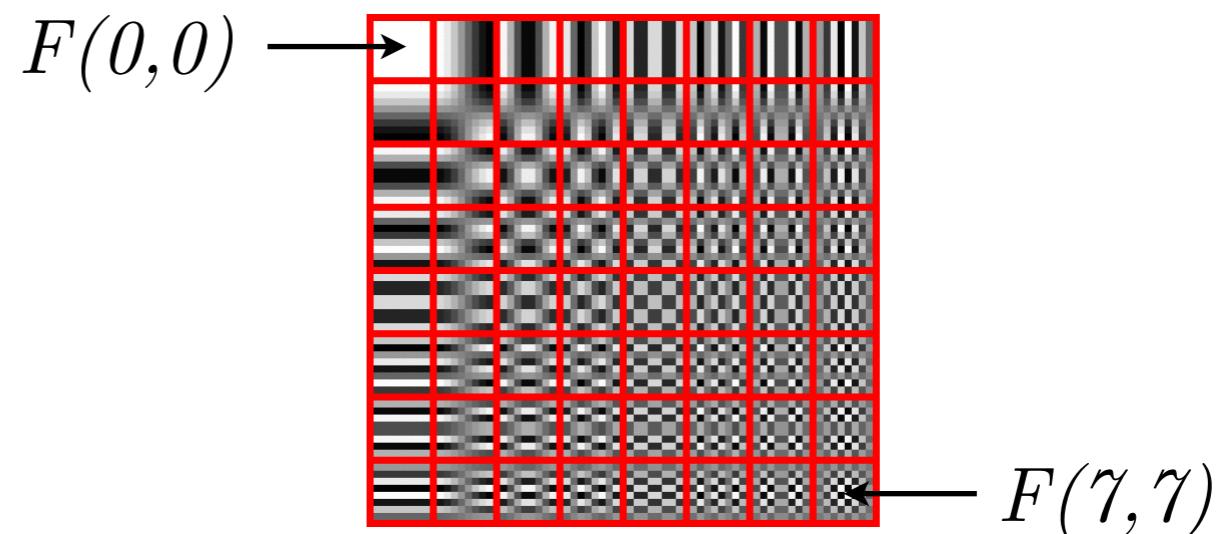


Wie lassen sich DCT-Koeffizienten komprimieren?

# Kompression der DCT



- Werte nahe bei Null können auf 0 gesetzt werden
- Anteile hoher Frequenz können mit weniger Bits codiert werden



# Quantisierung

- Quantisierung wird mit **Quantisierungsmatrix** durchgeführt
- jeder DCT-Koeffizienten wird durch Wert  $Q(u,v) \geq 1$  geteilt

$$F'(u,v) = Round\left(\frac{F(u,v)}{Q(u,v)}\right)$$

- falls  $Q(u,v)=1$  gehen weniger Informationen verloren als  $Q(u,v)>1$

# Quantisierung

## Beispiel

$$F = \begin{bmatrix} 782,91 & 44,93 & 172,52 & -35,28 & -20,58 & 35,93 & 2,88 & -3,85 \\ -122,35 & -75,46 & -7,52 & 55,00 & 30,72 & -17,73 & 8,29 & 1,97 \\ -2,99 & -32,77 & -57,18 & -30,07 & 1,76 & 17,63 & 12,23 & -13,57 \\ -7,98 & 0,66 & 2,41 & -21,28 & -31,07 & -17,20 & -9,68 & 16,94 \\ 3,87 & 7,07 & 0,56 & 5,13 & -2,47 & -15,09 & -17,70 & -3,76 \\ -3,77 & 0,80 & -1,46 & -3,50 & 1,48 & 4,13 & -6,32 & -18,47 \\ 1,78 & 3,28 & 4,63 & 3,27 & 2,39 & -2,31 & 5,21 & 11,77 \\ -1,75 & 0,43 & -2,72 & -3,05 & 3,95 & -1,83 & 1,98 & 3,87 \end{bmatrix}$$

$$Q = \begin{bmatrix} 10 & 15 & 25 & 37 & 51 & 66 & 82 & 100 \\ 15 & 19 & 28 & 39 & 52 & 67 & 83 & 101 \\ 25 & 28 & 35 & 45 & 58 & 72 & 88 & 105 \\ 37 & 39 & 45 & 54 & 66 & 79 & 94 & 111 \\ 51 & 52 & 58 & 66 & 76 & 89 & 103 & 119 \\ 66 & 67 & 72 & 79 & 89 & 101 & 114 & 130 \\ 82 & 83 & 88 & 94 & 103 & 114 & 127 & 142 \\ 100 & 101 & 105 & 111 & 119 & 130 & 142 & 156 \end{bmatrix} \xrightarrow{\text{Red Arrow}} F' = \begin{bmatrix} 78 & 3 & 7 & -1 & 0 & 1 & 0 & 0 \\ -8 & -4 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & -1 & -2 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

# Quantisierung

## Beispiel



# Kompression

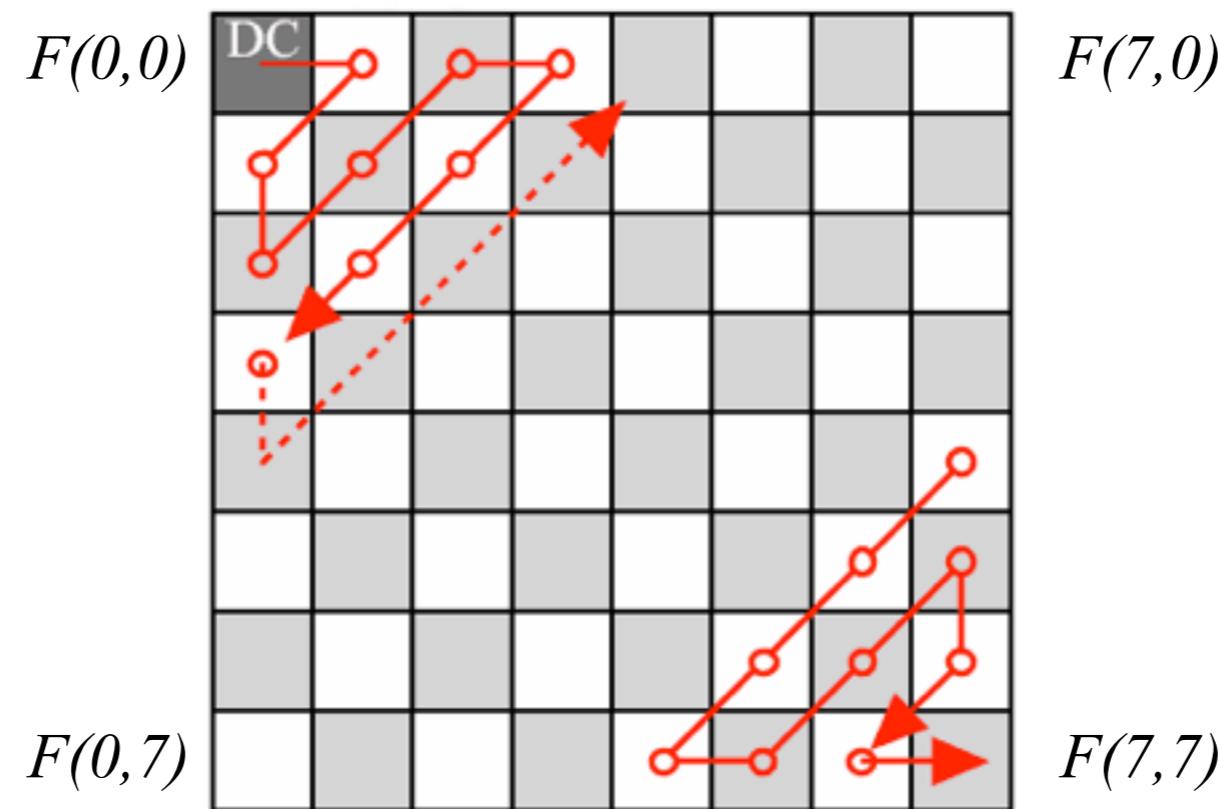
## DC-Komponenten

- verlustfreie Kompression der DC- und AC-Komponenten  $F'(0,0)$  bzw.  $F'(u,v)$
- DC-Komponenten der 8x8-Blöcke werden prädiktiv kodiert, da benachbarte Blöcke oft ähnliche Grundtöne haben
  - weniger Bits für kleinere Differenzen
- DC-Komponenten ergeben Vorschaubild

# Kompression

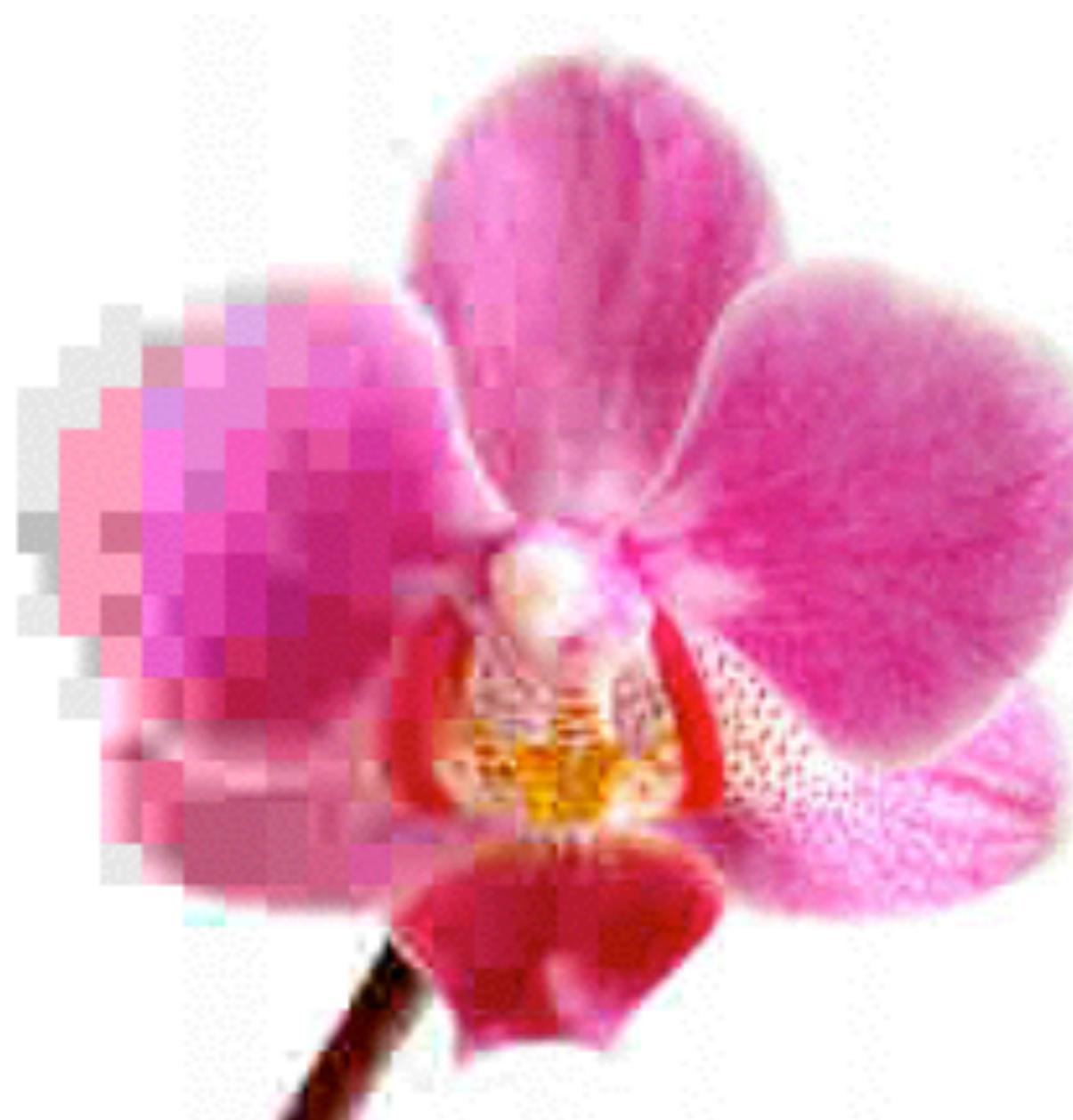
## AC-Komponenten

- AC-Komponenten werden im Zick-Zack-Muster mit Lauflängencodierung ausgelesen
- anschließend Entropiekodierung



# JPEG

## Beispiel: Qualität



→ Qualität und Dateigröße

# Rastergrafikformate

## Beispiele



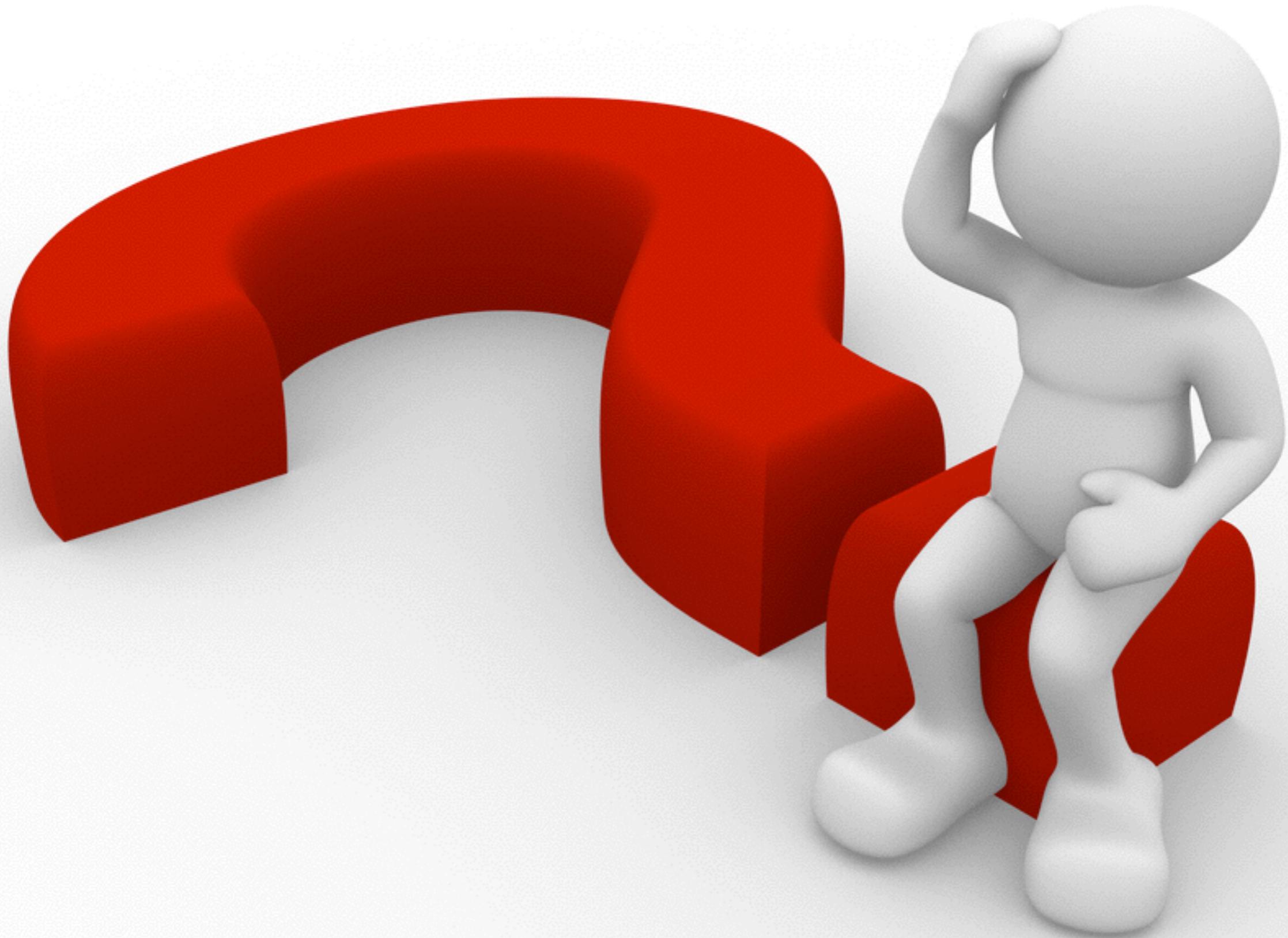
JPG: 1,3 MB



JPG: 250 KB



JPG: 60 KB





# Interaktive Medien

## Kapitel Bilder

### Bildverarbeitung

# Bildverarbeitung

- nach Kodierung, Kompression und Speicherung fängt eigentliche Arbeit an
- Beispiele:
  - Skalierung oder Drehung
  - Entzerrung und Filterung
  - Aufbereitung, Schnitt, Montage und Manipulation
  - ...

# Bildverarbeitung

- **Bildverarbeitung** bezeichnet Verarbeitung von Signalen, die Bilder repräsentieren
- Resultat einer Bildverarbeitung ist wieder Bild (**Bildbearbeitung**) oder Menge von Merkmalen des Eingabebildes (**Bilderkennung**)

# Bildbearbeitung

- **Bildbearbeitung** beschäftigt sich mit **Manipulation** von Bildern zur anschließenden Darstellung
- Beispiele
  - Anti-Aliasing, z.B. Weichzeichnen von Objekträndern, Kantenhervorhebung, Farbmanipulationen, Ausschneiden ...

# Bildbearbeitung

## Beispiele

- Anwendung von Bildoperatoren zur Erzeugung verschiedener Effekte



Original



Negativ-Bild



Solarisation



Bleistiftzeichnung



Relief

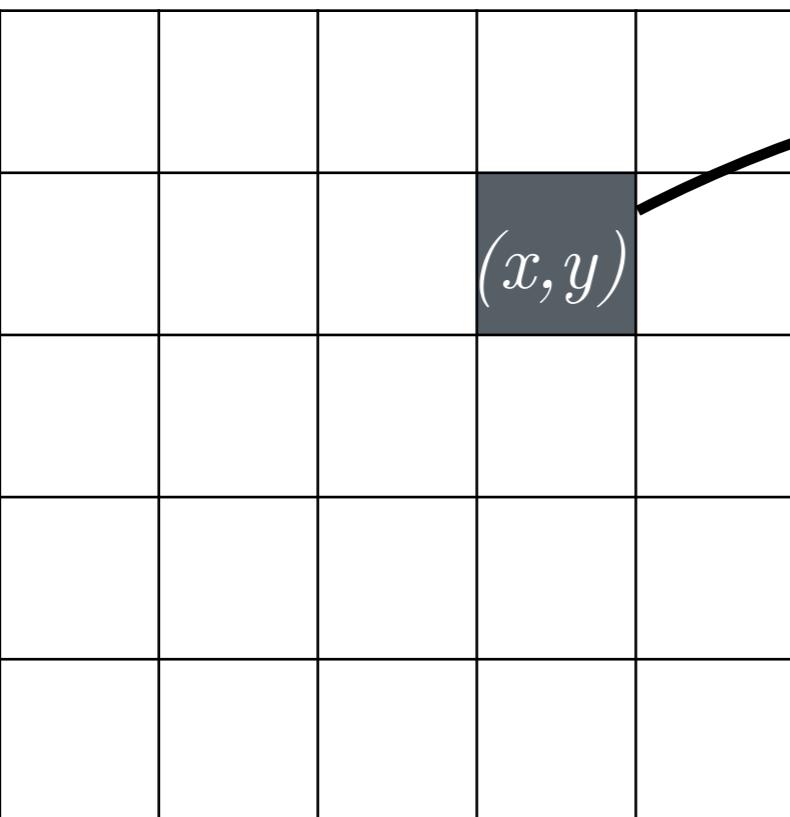
# Punkt-Operation

- **Punkt-Operation** ändert Werte einzelner Pixel ohne Nachbarpixel zu betrachten
  - bei **homogenen Punkt-Operationen** werden Koordinaten nicht berücksichtigt
  - bei **inhomogenen Punkt-Operationen** werden Koordinaten der Punkte berücksichtigt, d.h. *adaptive Tonwertkorrektur*

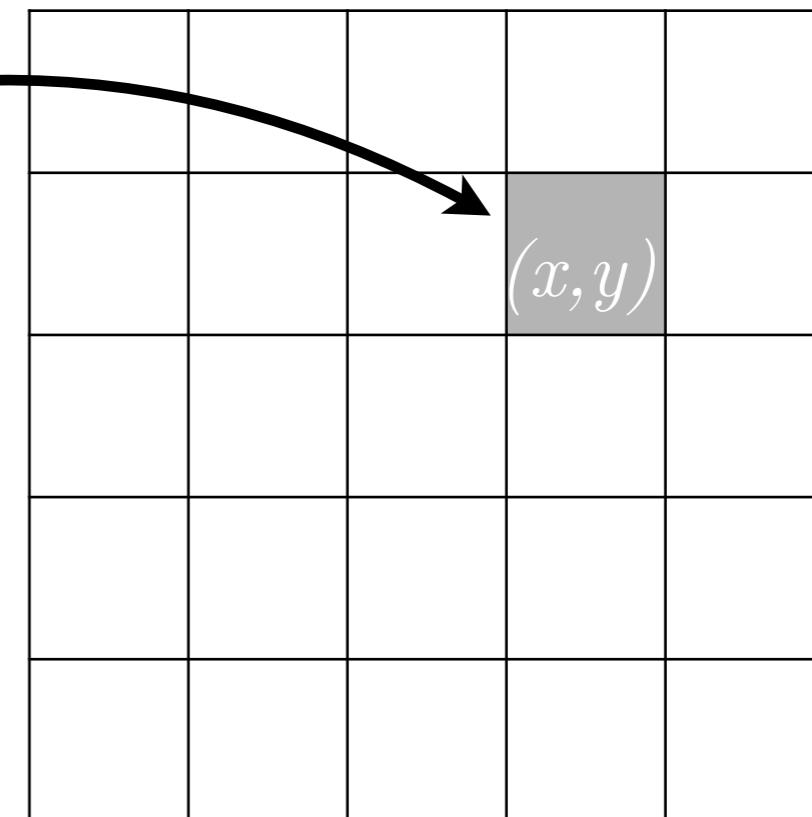
# Punkt-Operation

## Beispiel

Eingabebild



Ausgabe



# Punkt-Operation

## Homogen

- **Homogene Punkt-Operation** ist typische Operation zur Verbesserung des Bildkontrastes
- Beispiel: Umwandlung von Grauwertbild  $f(x,y)$  in Schwarzweißbild  $f'(x,y)$  mit Schwellwert  $w$  (z.B.  $w=0.5$ ):

$$f'(x,y) = \begin{cases} 1, & \text{falls } f(x,y) > w \\ 0, & \text{sonst} \end{cases}$$

# Punkt-Operation

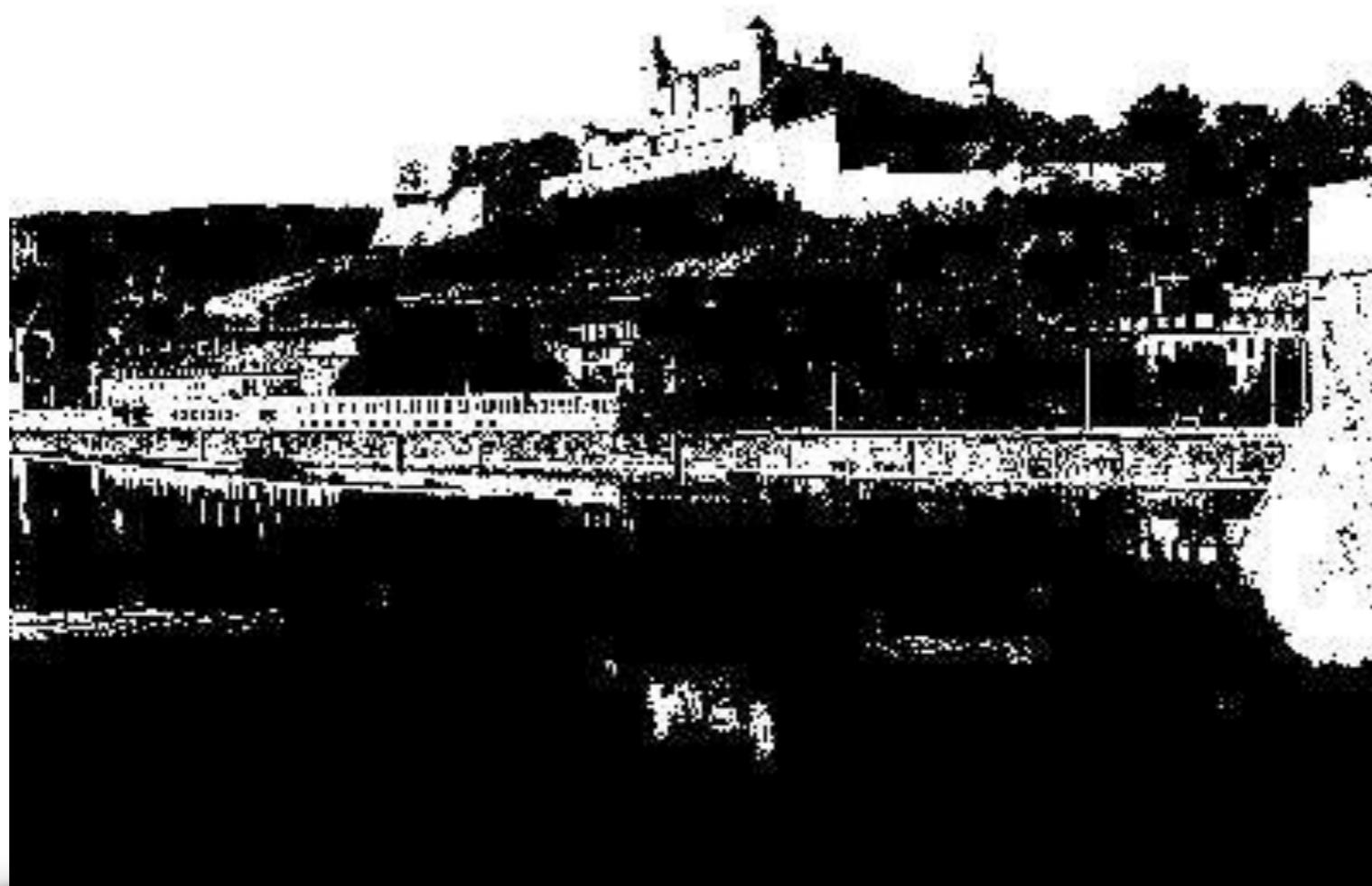
## Beispiel



24-Bit (Farbe)

# Punkt-Operation

## Beispiel



24-Bit (Farbe) → 1-Bit S/W

# Punkt-Operation

## Beispiel



24-Bit (Farbe) → 8-Bit Graustufen

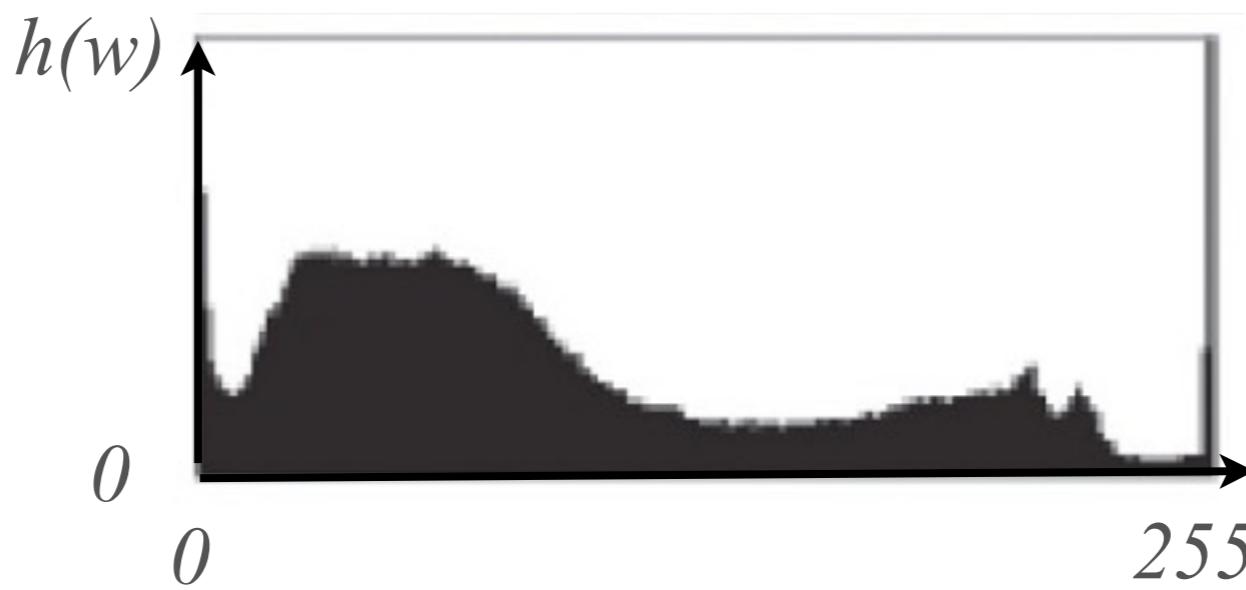
# Histogramme

- **Histogramm** gibt Häufigkeitsverteilung metrisch skalierter Merkmale, z.B. Pixelwerte, an
- **Histogramm** enthält nützliche Informationen über Bilder und deren Pixelverteilungen
  - Beispiel: Kontrastumfang eines Bildes lässt sich aus Histogramm ablesen

# Histogramm

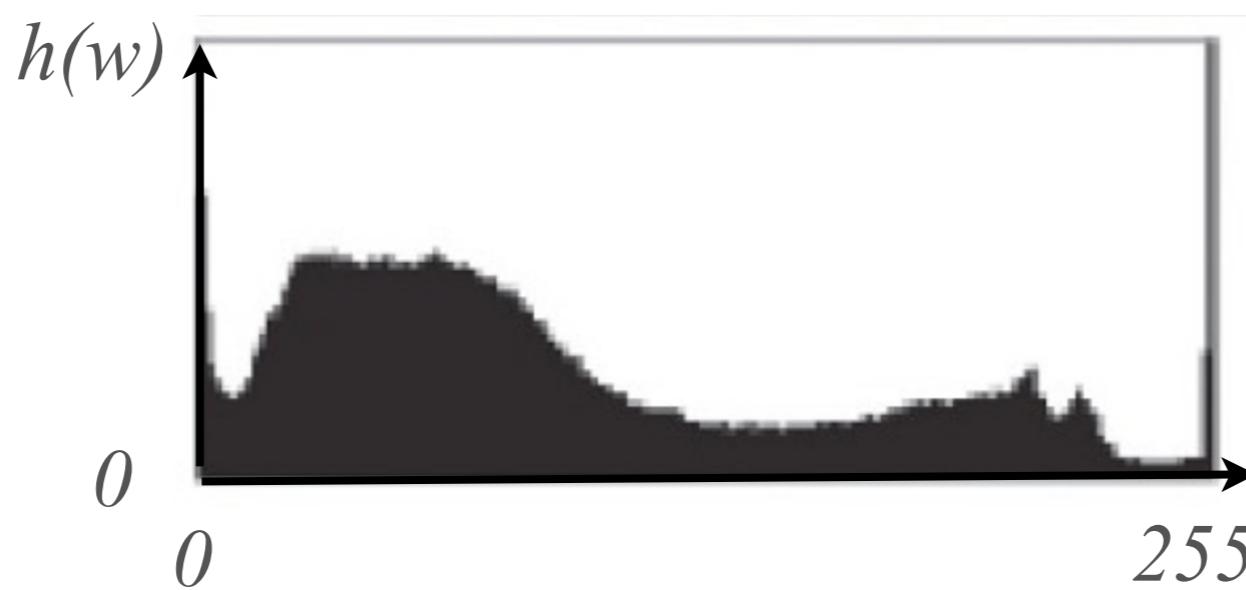
## Beispiel

- Histogramm eines 8-Bit Grauwertbilds (256 Grautöne) lässt sich durch Funktion  $h(w)$ ,  $w=0,\dots,255$  repräsentieren
- $h(w)$  gibt Anzahl Pixel mit Wert  $w$  an



# Histogramm

## Beispiel



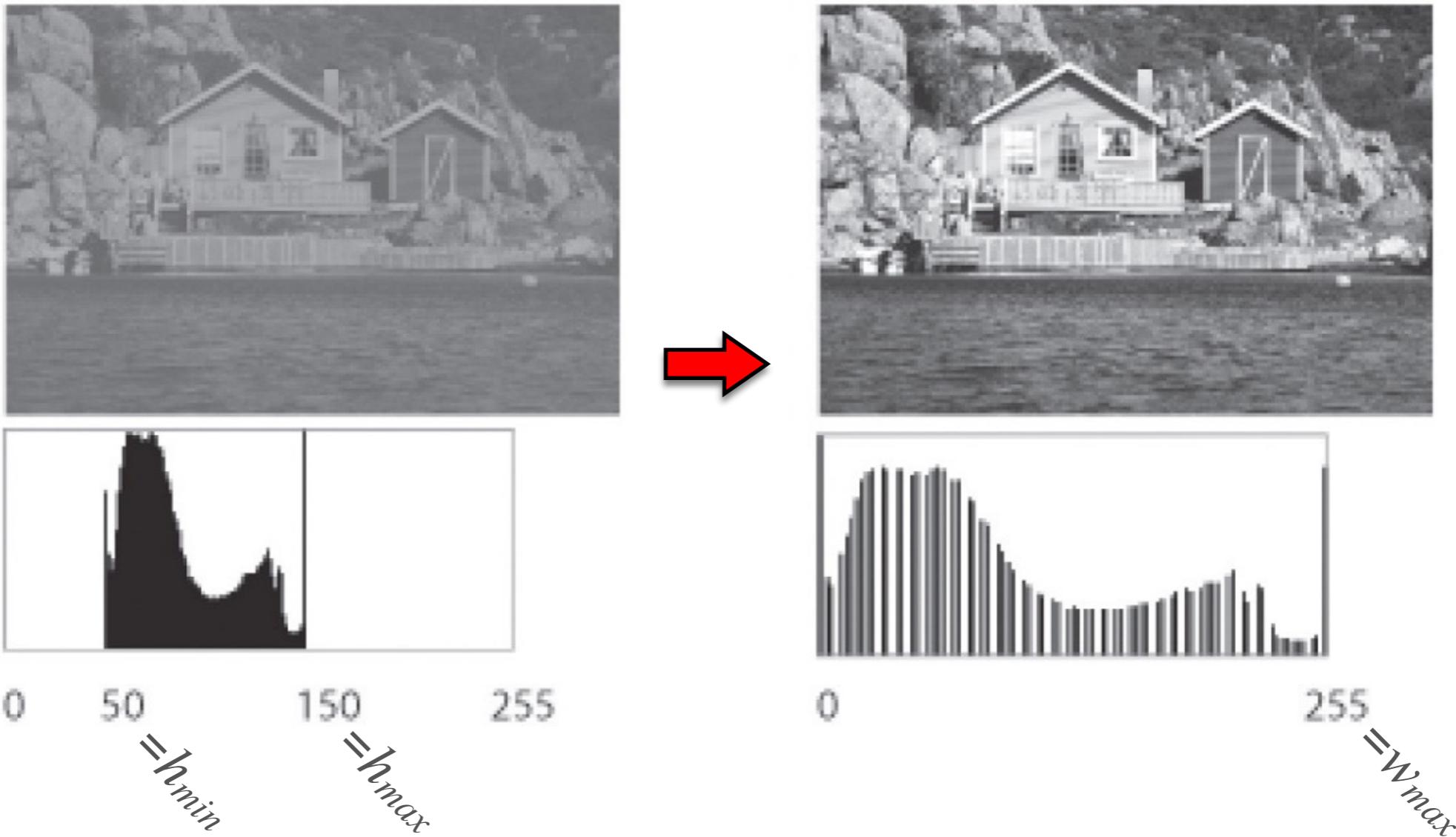
# Histogrammausgleich

## Beispiel

- Sei  $f(x,y)$  Grauwertbild mit 8 Bit nur Werte zwischen 50 ( $h_{min}$ ) und 150 ( $h_{max}$ ) verwendet,
- Histogrammausgleich erlaubt **Spreizen** der Werte von 0 bis 255 ( $w_{max}$ ) führt zu besserem Kontrast

# Histogrammausgleich

## Beispiel



# Gruppenarbeit

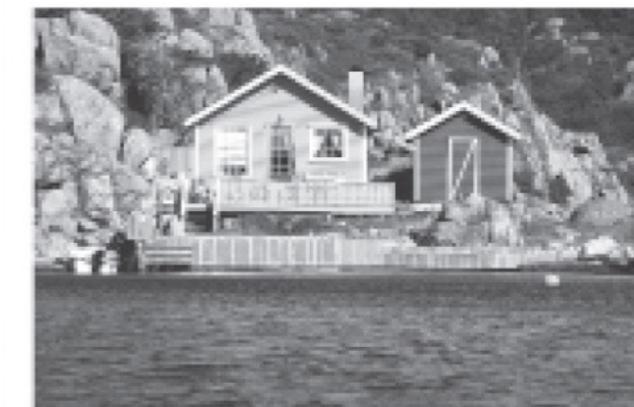
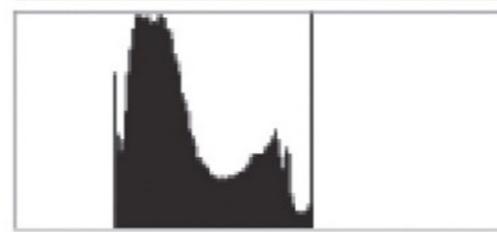
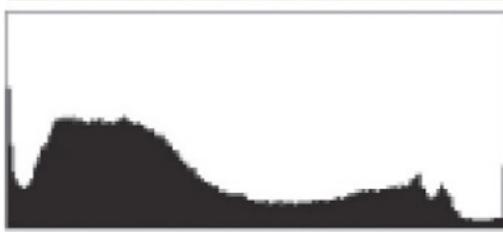
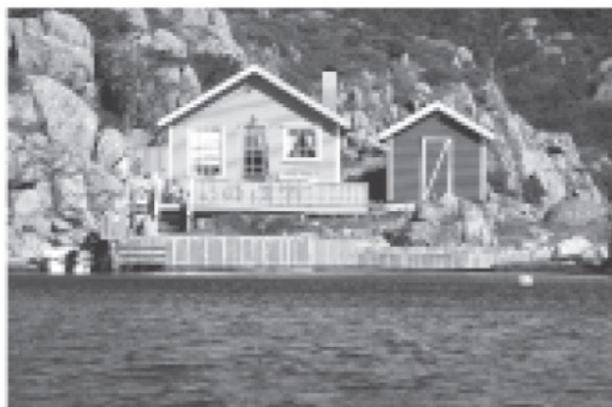


Wie lassen sie die Werte spreizen?

# Histogrammausgleich

## Beispiel

$$f'(x,y) = \frac{f(x,y) - h_{\min}}{h_{\max} - h_{\min}} w_{\max}$$



0 255

0 50 150 255

0 255

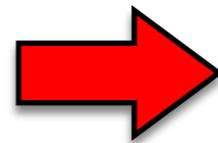
# Punkt-Operationen

## Inhomogen

- **Inhomogene Punkt-Operationen**  
manipulieren Pixel in Abhängigkeit ihrer Koordinaten oder Eigenschaften
- Beispiel: Korrektur von Abschattungen, visuelle Effekte, Shading-Korrekturen ...

# Punkt-Operationen

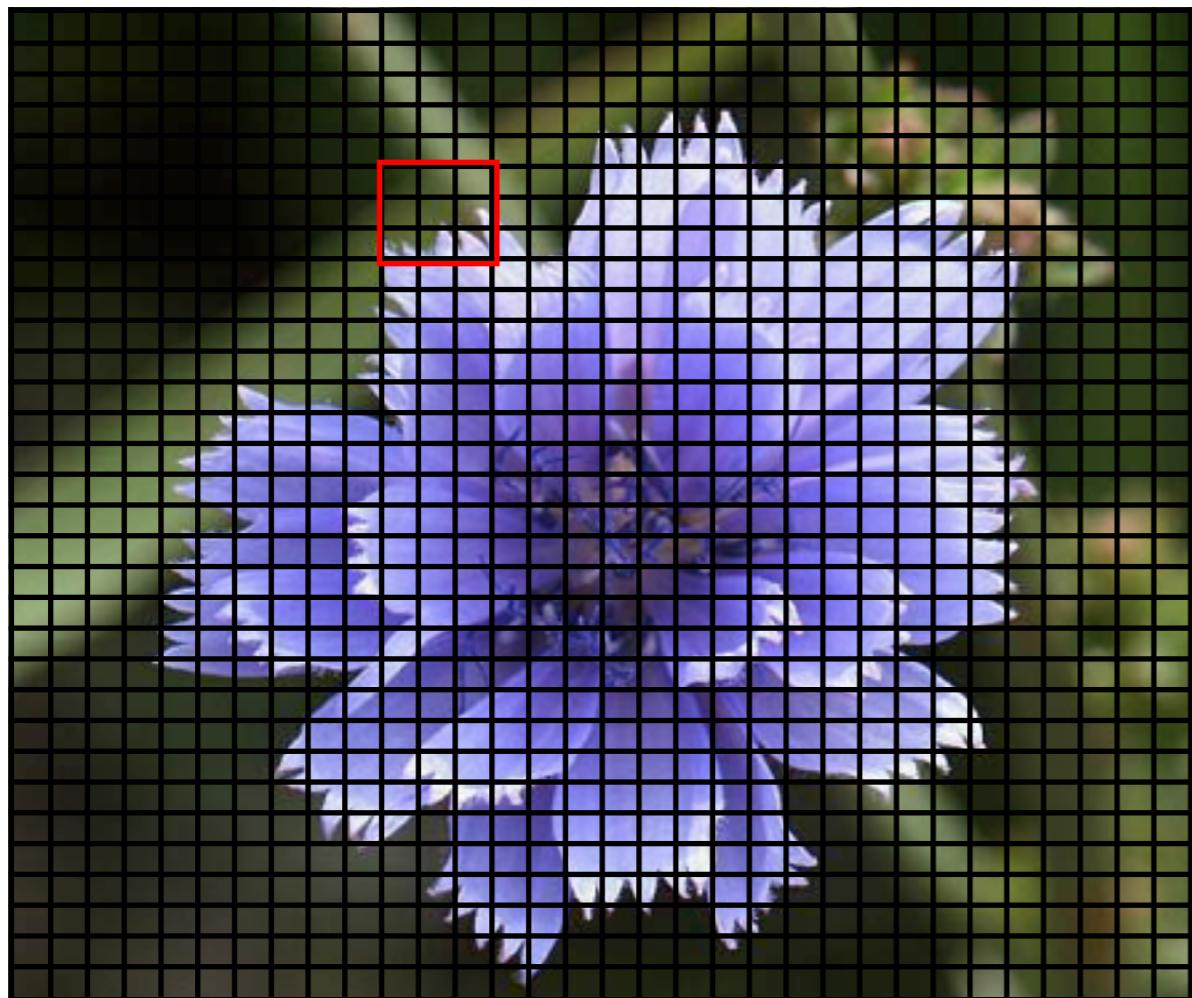
## Inhomogen



# Filter

- **Filter** manipulieren Pixel und beziehen Informationen von mehreren Pixeln (meist aus Nachbarschaft) im Originalbild mit ein
- Beispiele: Glättungs-/Weichzeichner, Scharfzeichner ...

# Filter Beispiel



$f(x-1,y-1)$	$f(x,y-1)$	$f(x+1,y-1)$
$f(x-1,y)$	$f(x,y)$	$f(x+1,y)$
$f(x-1,y+1)$	$f(x,y+1)$	$f(x+1,y+1)$

# Faltungsfilter

- **Faltungsfilter** lassen sich mit **Faltungsfunktion** realisieren

$$f'(x, y) = \sum_{i=-n}^n \sum_{j=-m}^m g(i, j) \cdot f(x + i, y + j)$$

- **Faltung** eines Bildsignals  $f$  mit **Maske (Kernel)**  $g$

# Filter

## Bsp: Weichzeichner

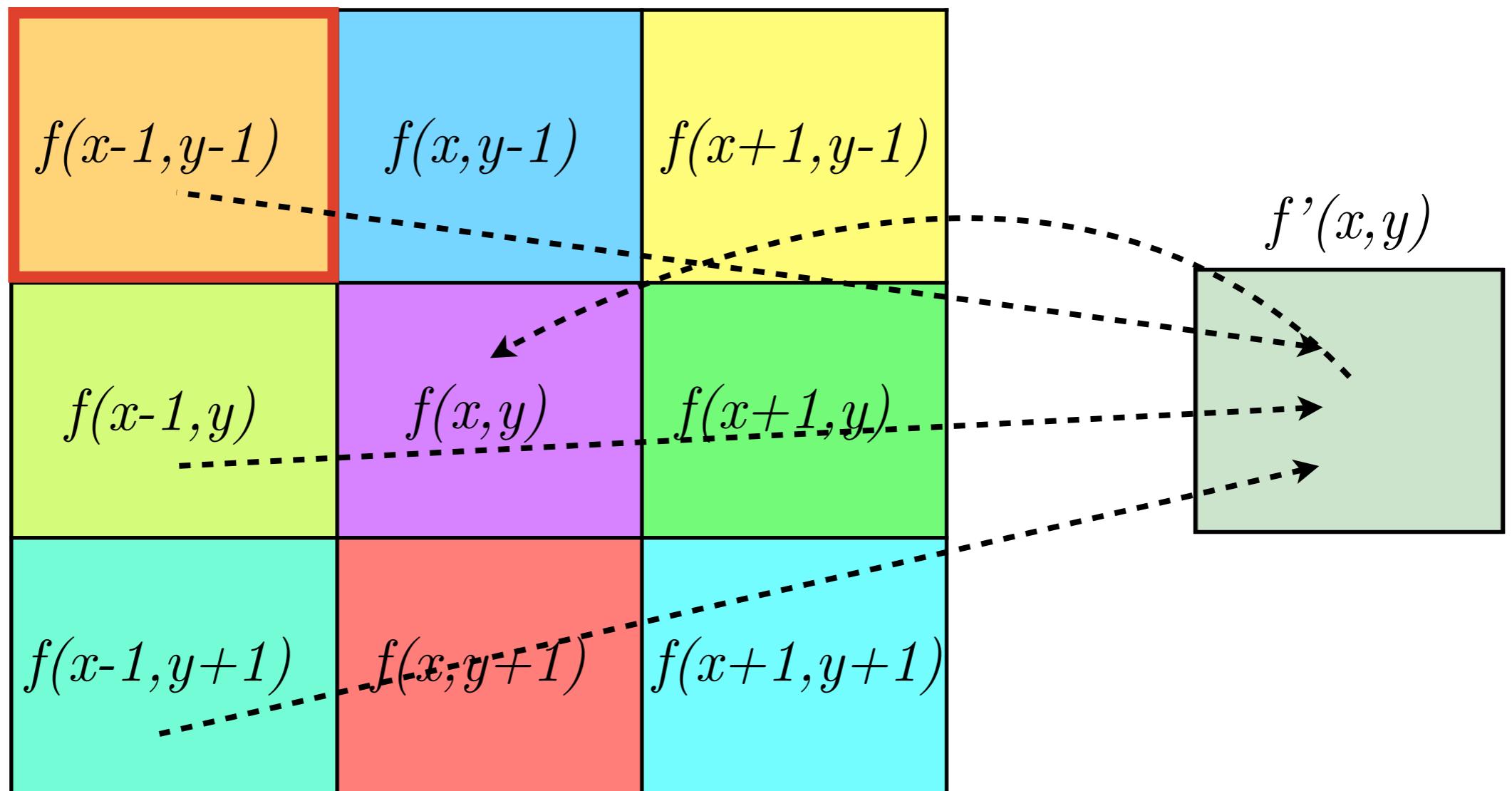
- **Weichzeichner** ist linearer Filter, bei dem **Filterkernel** mit bspw. 9 Pixel berücksichtigt wird

$$f'(x, y) = \frac{1}{9} \sum_{i=-1}^1 \sum_{j=-1}^1 f(x + i, y + j)$$

- *Bemerkung: Weichzeichner berechnet Mittelwert aus umliegenden Pixel*

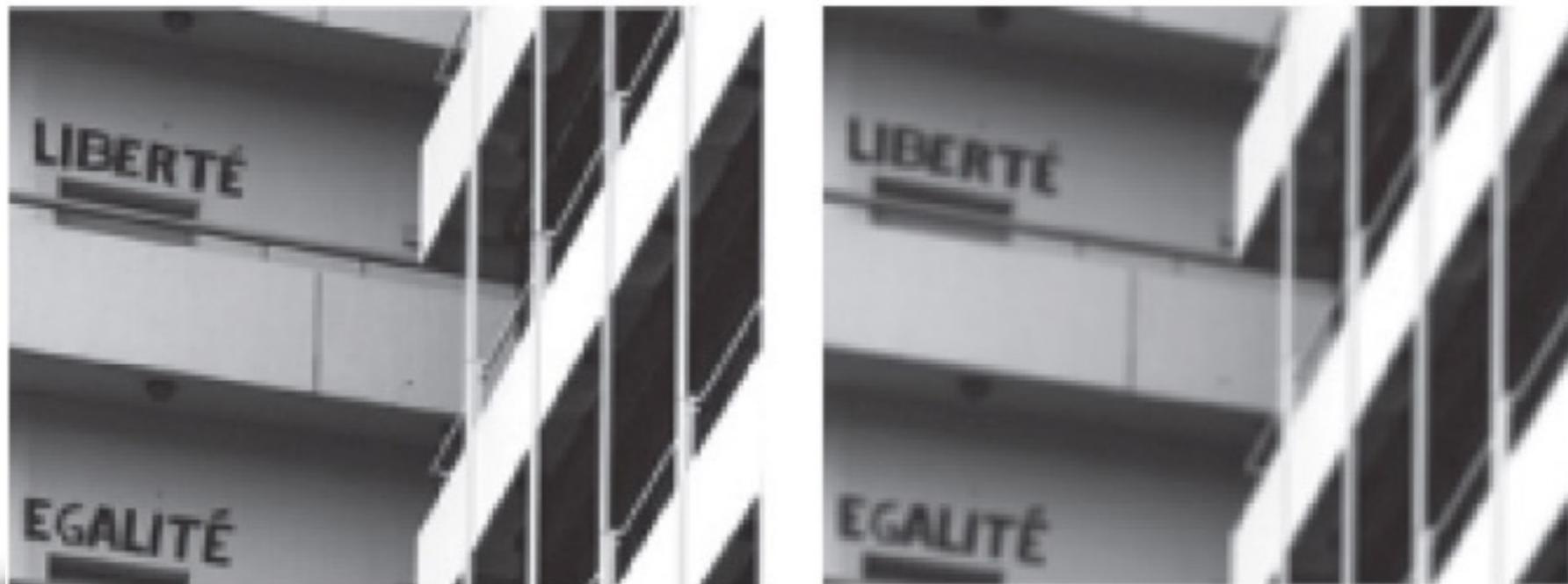
# Filter

## Bsp: Glättungs-/Weichzeichner



# Filter

Bsp: Weichzeichner

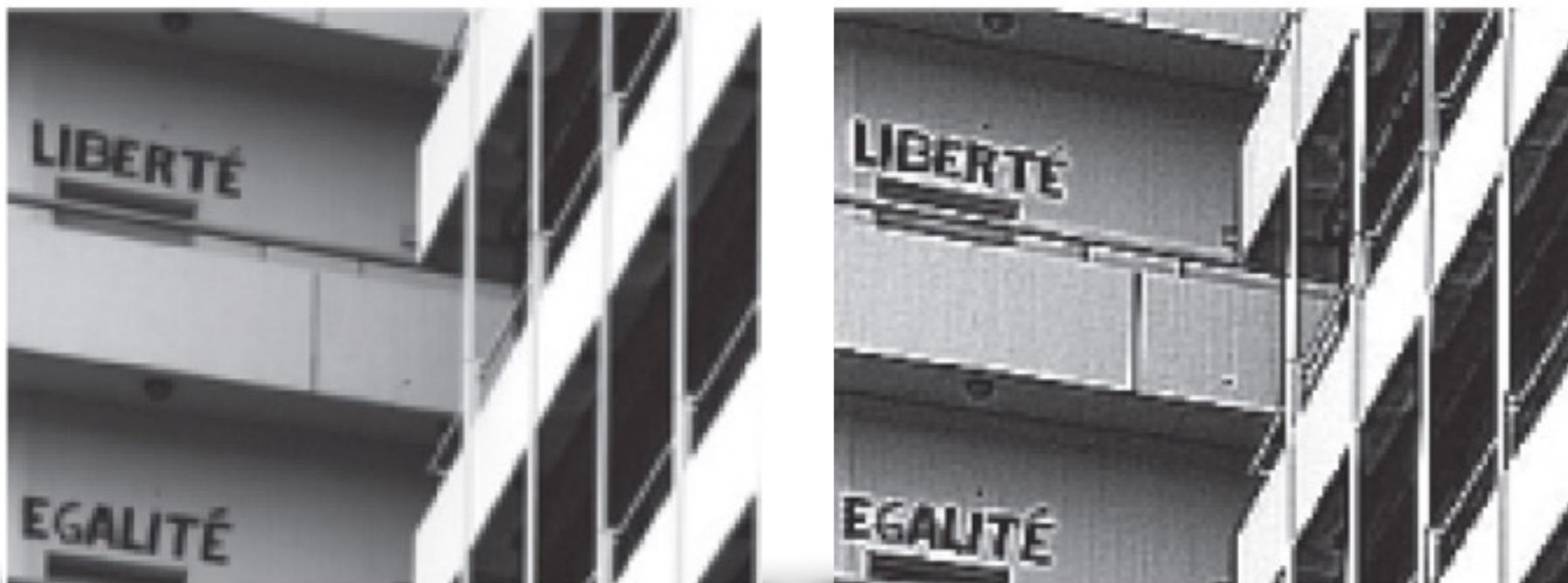


$$f'(x,y) = \frac{1}{9} \begin{array}{|ccc|} \hline & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 \\ \hline & 1 & 1 & 1 \\ \hline \end{array} \times \begin{pmatrix} f(x-1,y-1) & f(x,y-1) & f(x+1,y-1) \\ f(x-1,y) & f(x,y) & f(x+1,y) \\ f(x-1,y+1) & f(x,y+1) & f(x+1,y+1) \end{pmatrix}$$

↑  
Filterkernel

# Filter

## Bsp: Scharfzeichner



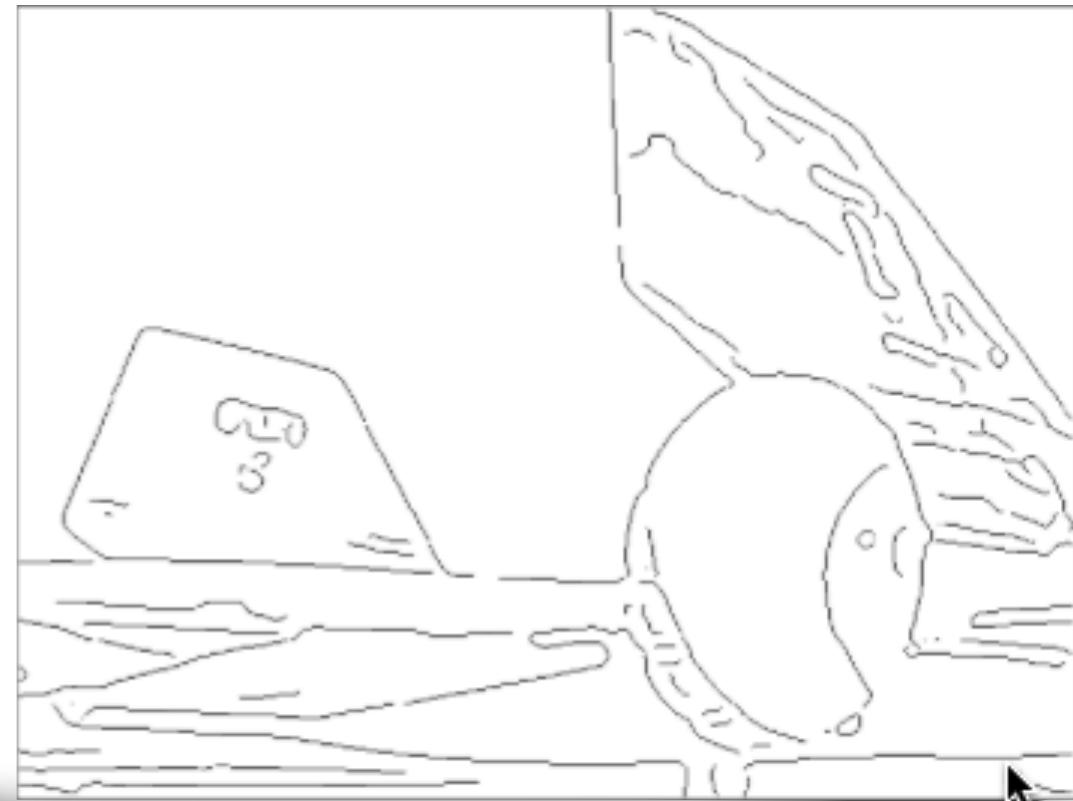
$$f'(x,y) = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{pmatrix} \times \begin{pmatrix} f(x-1,y-1) & f(x,y-1) & f(x+1,y-1) \\ f(x-1,y) & f(x,y) & f(x+1,y) \\ f(x-1,y+1) & f(x,y+1) & f(x+1,y+1) \end{pmatrix}$$

↑  
Filterkernel

# Fokus Mediendesign

## Kantendetektion

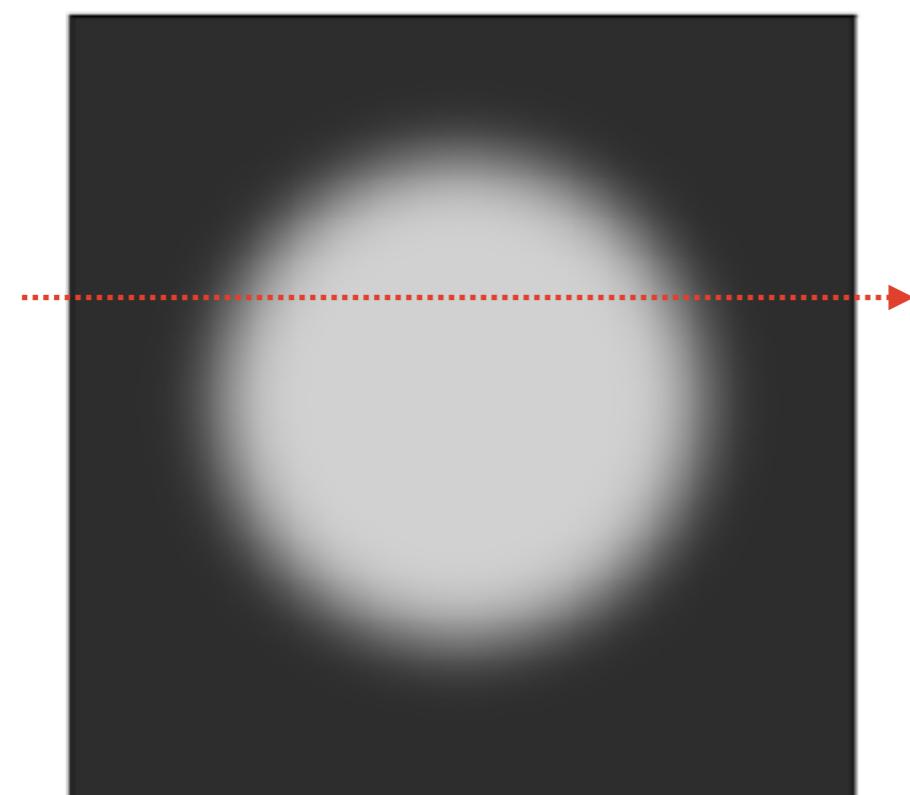
- Auffinden von **Kanten**, d.h. starken Intensitätsübergängen, in Bildern



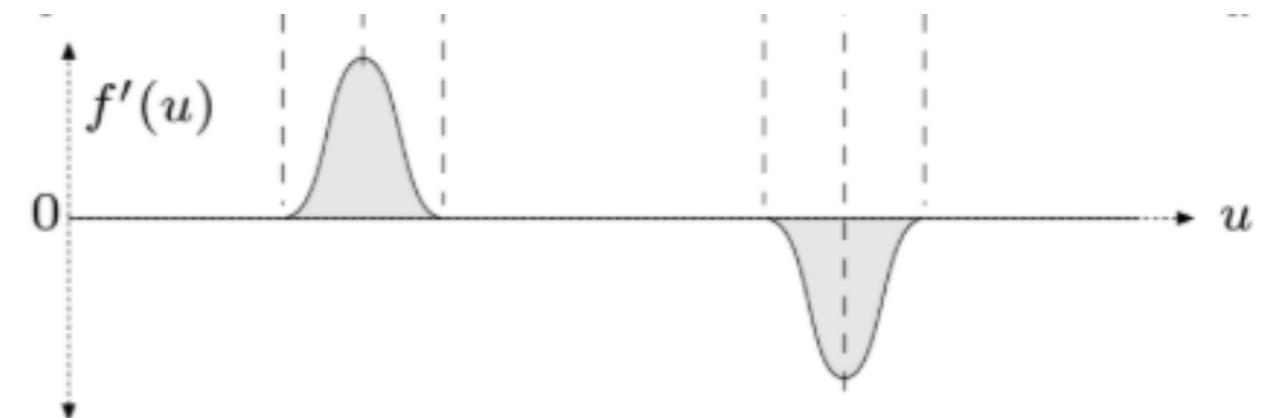
# Fokus Mediendesign

## Kantendetektion

- **Kanten** sind Bildorte, an denen sich Intensität auf kleinem Raum stark verändert



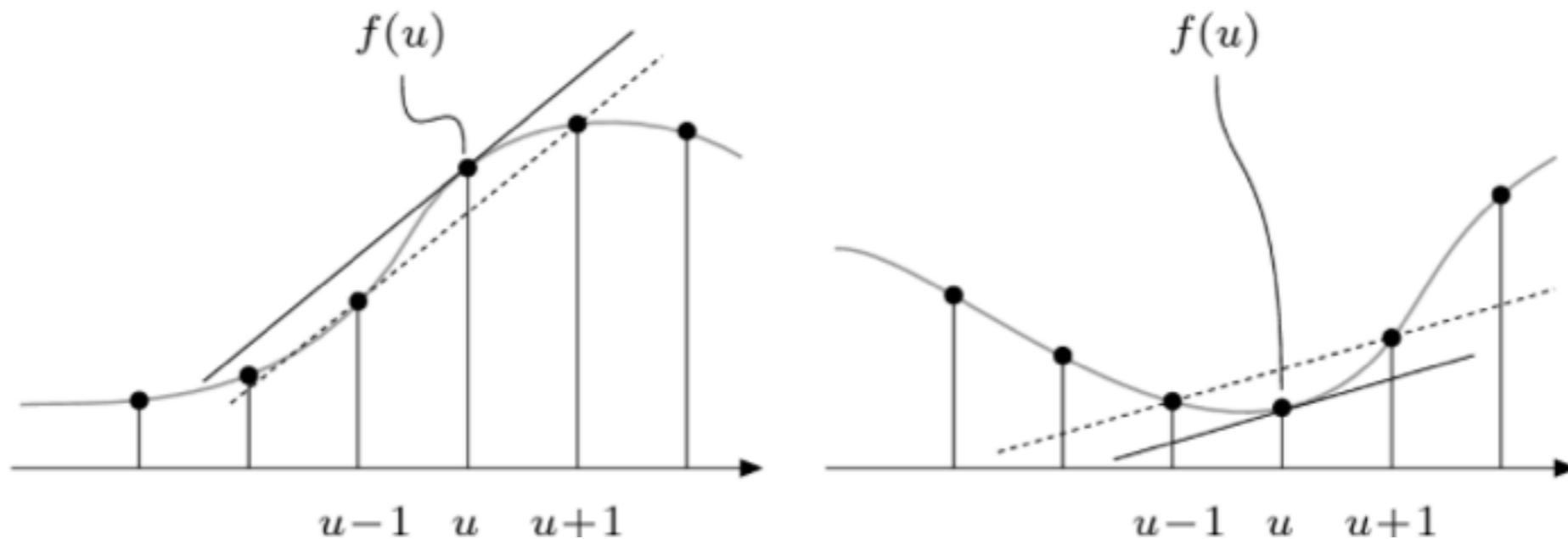
$$f'(u) = \frac{df(u)}{du}$$



# Fokus Mediendesign

## Kantendetektion

- Approximation der Ableitung auf diskretem Gitter durch endliche Differenzen



$$\frac{df(u)}{du} \approx \frac{1}{2} (f(u+1) - f(u-1))$$

# Fokus Mediendesign

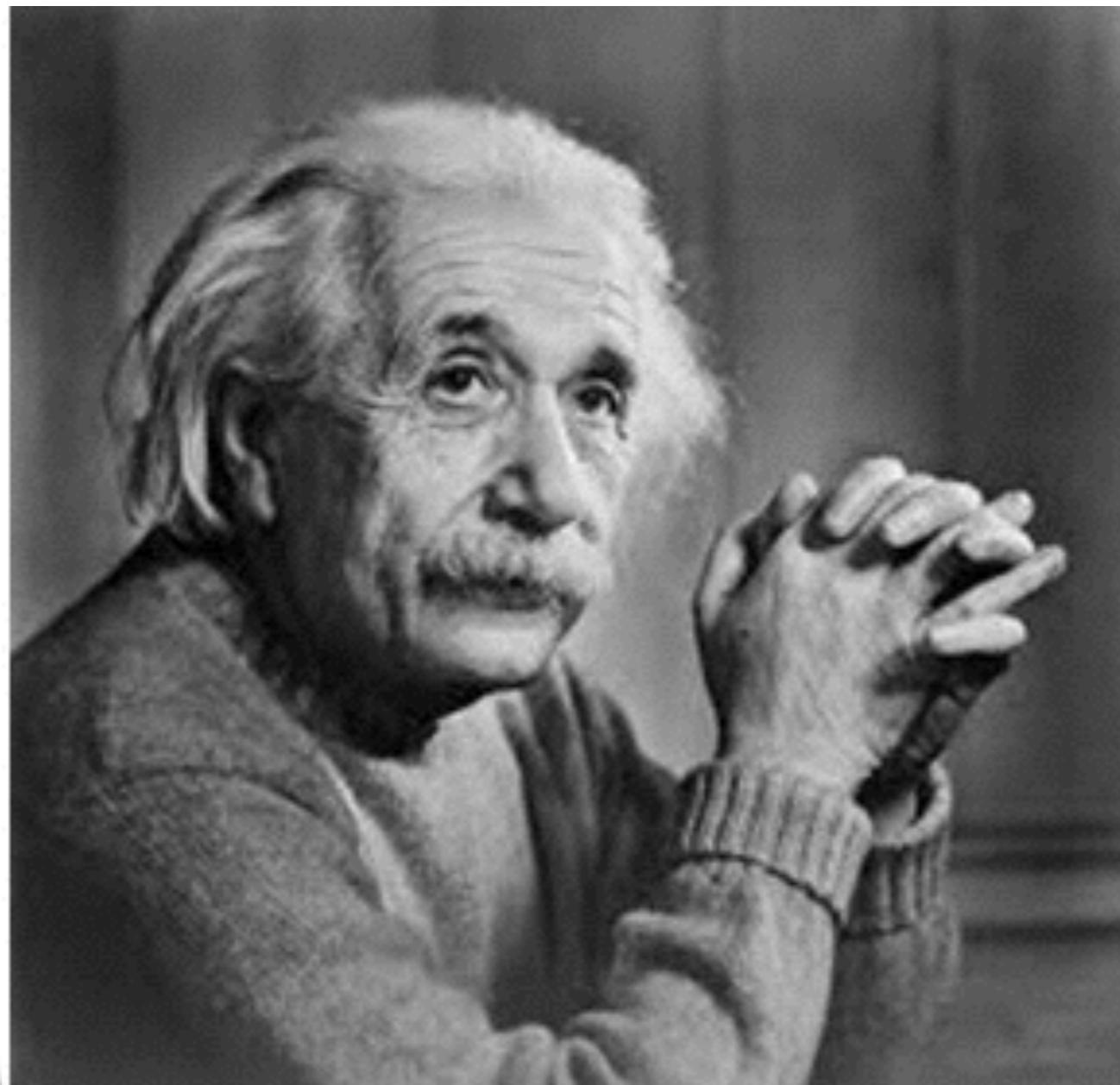
## Kantendetektion

- **Sobel-Operator** ist einfacher Kantendetektions-Filter
- **Sobel-Operator** berechnet erste Ableitung der Bildpunkt-Helligkeitswerte orthogonal zur Ableitungsrichtung

$$f'(x,y) = \frac{1}{8} \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} f(x-1,y-1) & f(x,y-1) & f(x+1,y-1) \\ f(x-1,y) & f(x,y) & f(x+1,y) \\ f(x-1,y+1) & f(x,y+1) & f(x+1,y+1) \end{pmatrix}$$

# Kantenerkennung

## Beispiel: Albert



# Kantenerkennung

## Beispiel: Sobelfilter

horizontaler  
Sobelfilter

$$\frac{1}{8} \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

vertikaler  
Sobelfilter

$$\frac{1}{8} \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$$

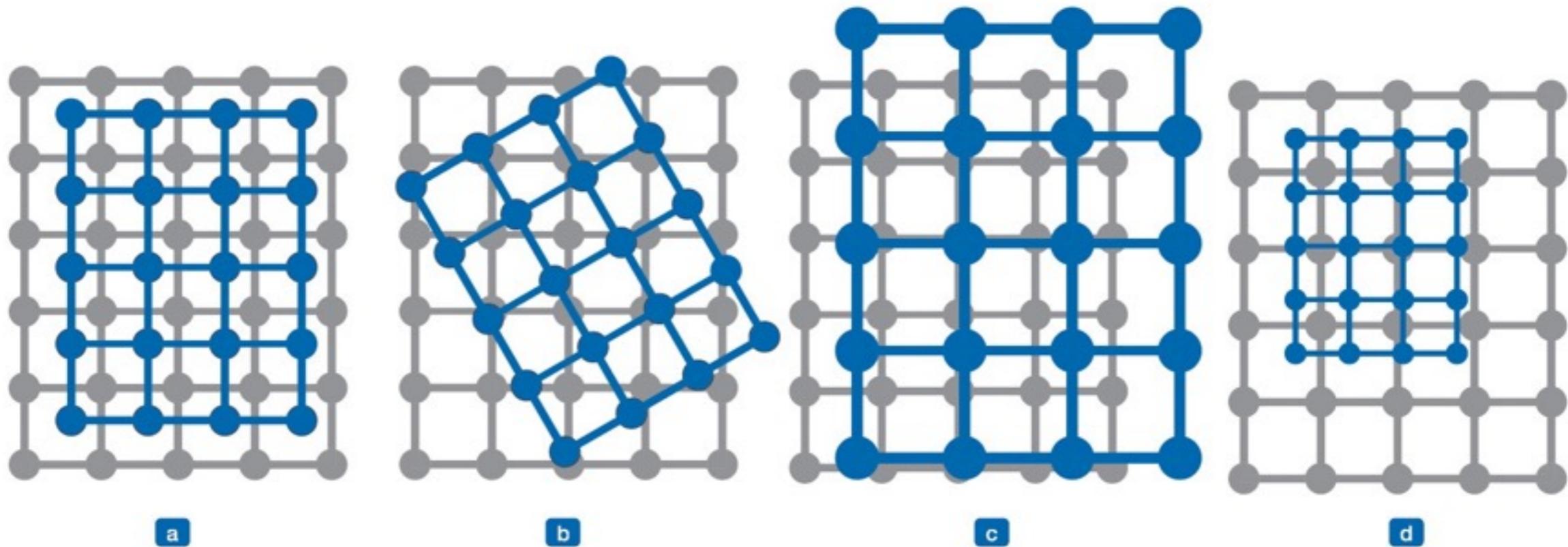


# Bildtransformation

- **Geometrische Bildtransformation** ist Operation bei der Pixel an andere Position verschoben wird
- Beispiele:
  - Verschiebung, Streckung (Vergrößern, Verkleinern), Drehung, Scherung
- nicht jedem **Ausgangspixel** kann genau ein **Zielpixel** zugeordnet werden

# Bildtransformation

## Beispiele: Geometrische



Probleme bei (a) Verschiebung, (b) Drehung,  
(c) Vergrößerung und (d) Verkleinerung

# Bildtransformation

## Down-Scaling

- Bei **Verkleinerung** (*engl. Down-Scaling*) eines Bildes werden einem **Zielpixel** evtl. mehrere **Ausgangspixel** zugeordnet
- bei solcher **Unterabtastung** (*engl. Subsampling*) kann es zu störenden Effekten (**Moiré-Effekt**) kommen
- **Moiré-Effekt** lässt sich durch Weichzeichner vor Skalierung reduzieren

# Bildtransformation

## Beispiel: Down-Scaling



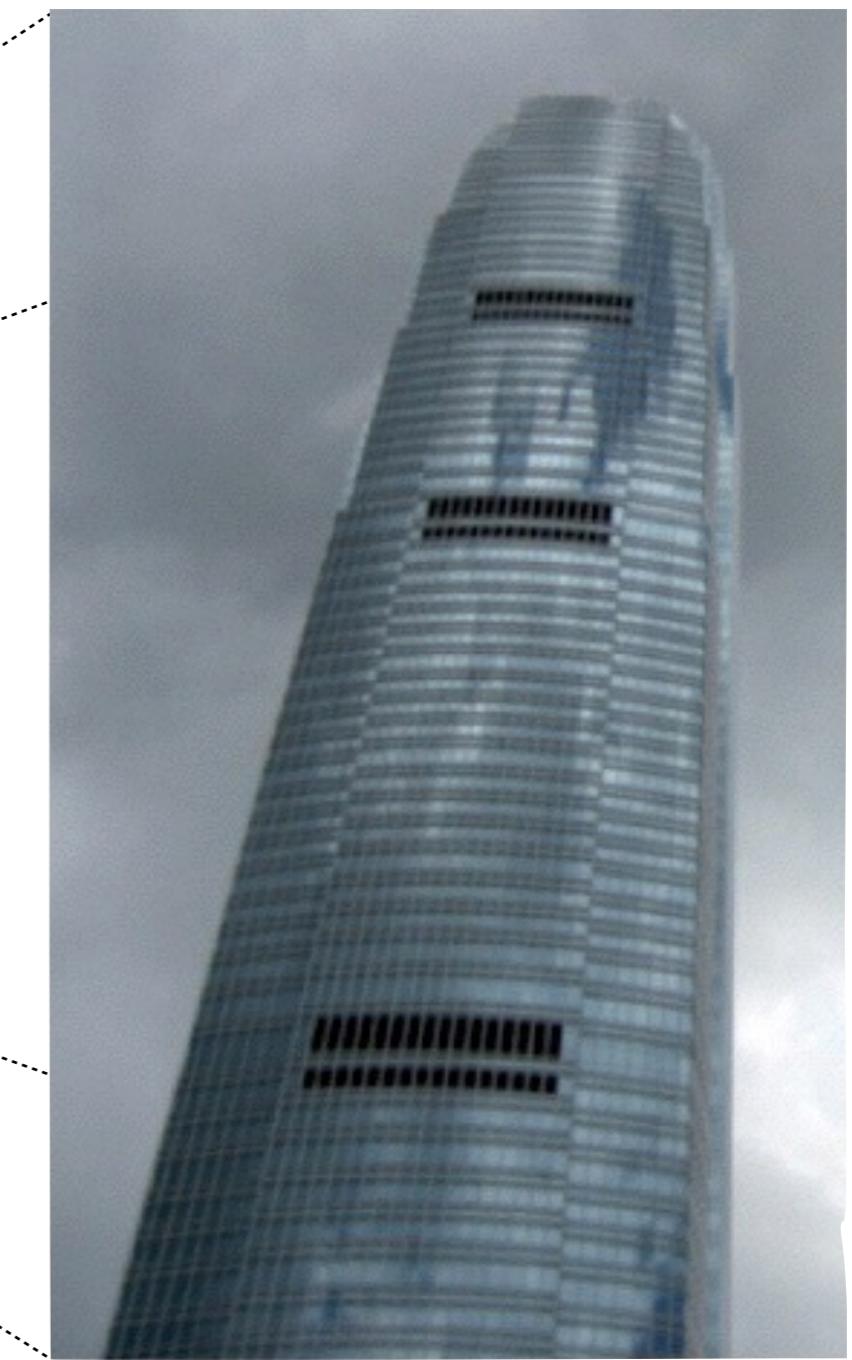
# Bildtransformation

## Beispiel: Down-Scaling

Weichzeichner



Skalierung



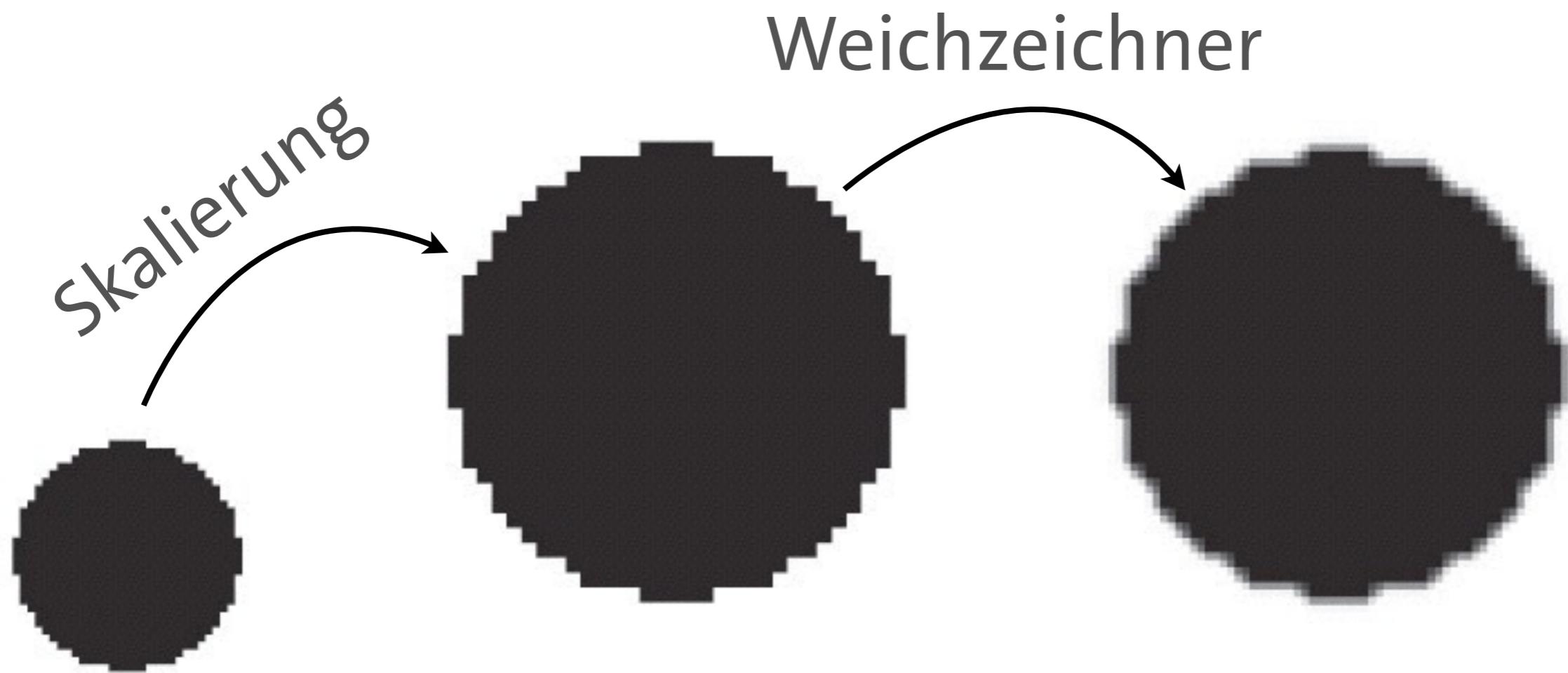
# Bildtransformation

## Up-Scaling

- bei **Vergrößerung** (*engl. Up-Scaling*) wird **Ausgangspixel** auf mehrere **Zielpixel** abgebildet
- *Bemerkung: endliche Bildauflösung des Grafikgerätes kann i.A. zu störenden Effekten führen, z.B. Treppeneffekt (Aliasing)*

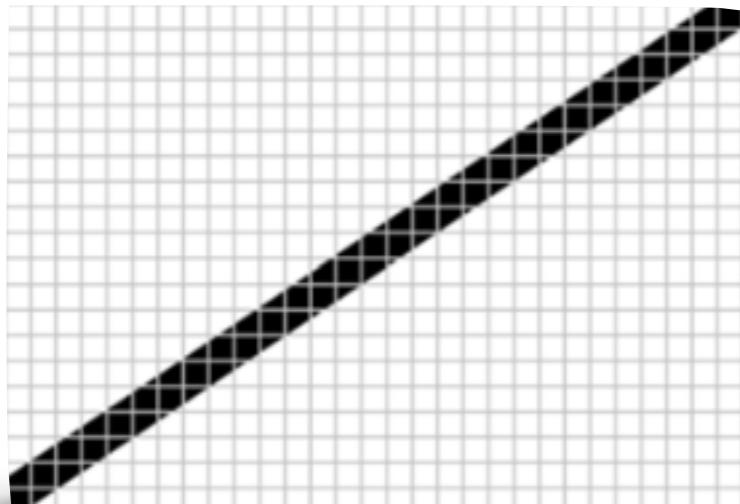
# Bildtransformation

Beispiel: Treppeneffekt

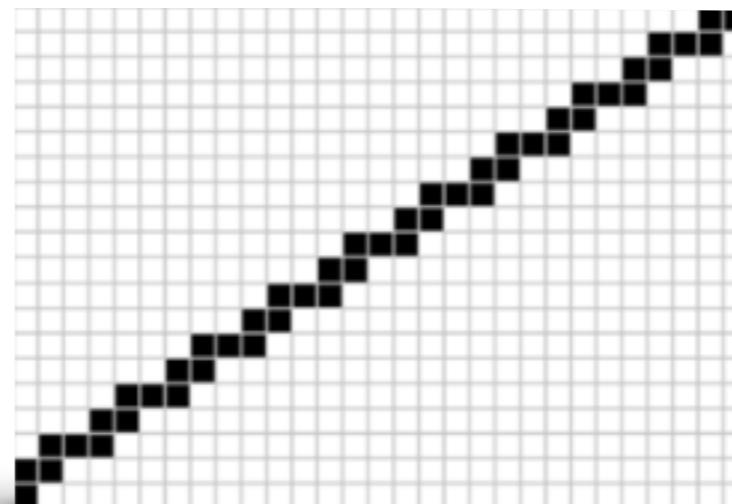


# Bildtransformation

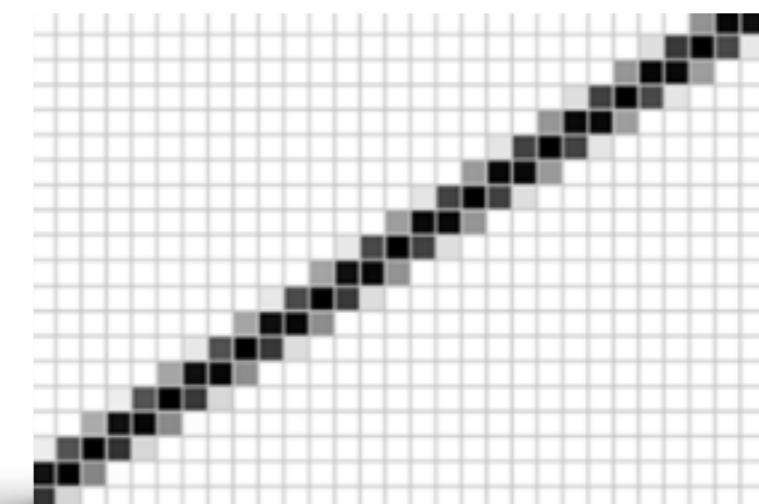
## Beispiel: Antialiasing



**Ideales Bild**  
wie auf Vektordisplay  
dargestellt



**Objektdarstellung  
(ohne Antialiasing)**  
Pixel wird gesetzt,  
wenn Bedeckung > 50%



**Objektdarstellung  
(mit Antialiasing)**  
Grauwerte der Randpixel  
gemäß  
Flächenbedeckungsgrad

