# Bilder

Jörg Cassens

Medieninformatik
WS 2019/2020

Universität Hildesheim
medieninformatik
IMAI – Institut für Mathematik und Angewandte Informatik

**Lernziele**

- Grundlagen der visuellen Wahrnehmung
  - Farbwahrnehmung
  - Darstellung in Grundfarben Rot, Grün und Blau
- Digitalisierung und Codierung digitaler, pixelbasierter Bilder
- Kompressionsverfahren für Rastergraphiken
  - Beispiel JPEG
- Grundlagen Bildbearbeitung

## 1 Licht

**Bilder**

- Oft als wichtigster menschlicher Sinn dargestellt
- Visuelle Eindrücke zweifelsohne wichtig
  - auch wenn es Tiere gibt, die besser sehen können und
  - unsere Sehleistung im Alter abnimmt
- Unsere Umwelt ist voll von Bildern
- Wandel von Produktion und Verarbeitung hin zu digitalen Technologien
- Ob DSLR oder Smartphone-Kamera, Prinzipien bei Digitalisierung und Speicherung die gleichen
  - Messung physikalischer Signale und Abbildung in einem digitalen Raster
  - Daher: geeignete Diskretisierung und Quantisierung
- Dazu: kompakte Speicherung
- Für die Be- und Verarbeitung: digitale Technologien

**Es werde Licht**

- Fotografieren: Mit Licht schreiben
- Licht kann heller oder dunkler sein, unterschiedliche Farben haben
- Vereinfachte Annahme: Keine Photonen, keine Relativität
- Licht ist eine **elektromagnetische Welle**
- Charakteristische Eigenschaften
  - Ausbreitungsrichtung, Frequenz/Wellenlänge, Amplitude, Polarisation

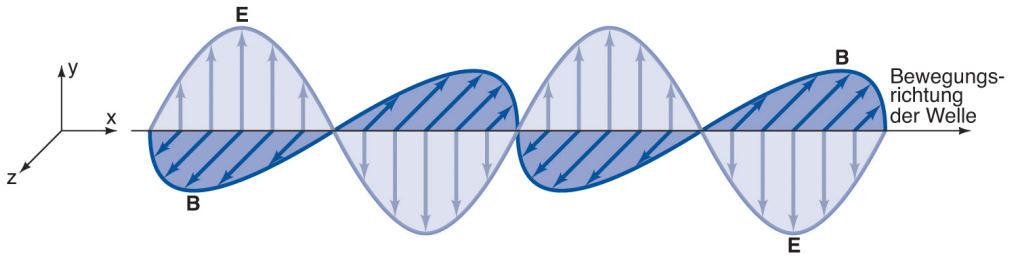



**Abbildung 3.1:** Zwei Wellen E und B, die sich in x-Richtung ausbreiten

**Grundlegende Größen**

- **Amplitude**: Maximaler Wert in einer Schwingung
- **Phase**: Ein Durchlauf
- **Phasenverschiebung**: Verschiebung des Nulldurchgangs (0-360 Grad)
- **Frequenz**: Anzahl der Schwingungen pro Zeiteinheit
- **Wellenlänge**: Der Weg, den das Signal bei gegebener Ausbreitungsgeschwindigkeit zurücklegt
- **Polarisation**: Ebene, in der die Schwingung verläuft

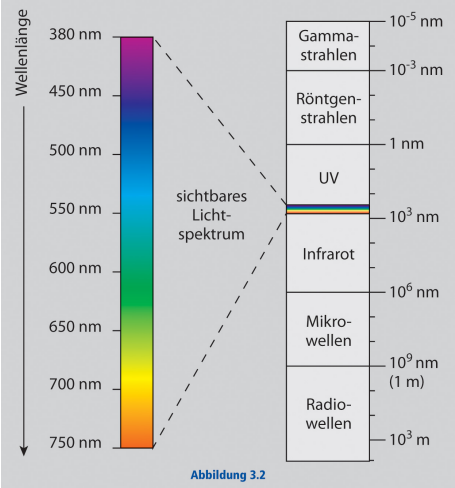
**Ausbreitungsrichtung**

- Elektromagnetische Wellen breiten sich von einer Quelle linear und mit konstanter Geschwindigkeit aus
- Abhängig von Wellenlänge und Beschaffenheit eines Hindernisses kann die Welle
  - abgelenkt, reflektiert werden
  - absorbiert werden
  - das Hindernis durchdringen
- Ungerichtete Quellen (Glühlampe)
- Gerichtete Quellen (Laser)
- Ausreichende Menge für die Wahrnehmung

**Frequenz**

- Entscheidet, ob und wie die Welle wahrgenommen wird
- Frequenz in Hz
- Sichtbarer Teil 385 bis 789 THz
- Idealisierte Ausbreitungsgeschwindigkeit 300.000.000 m/s
- Daher: Eine Schwingung ca. 380 bis 780 nm (Wellenlänge)
- Was ist sichtbar: unterschiedlich je nach Lebewesen
- Wellenlänge entscheidet über Farbwahrnehmung

**Frequenz (contd.)**




**Amplitude, Polarisation**

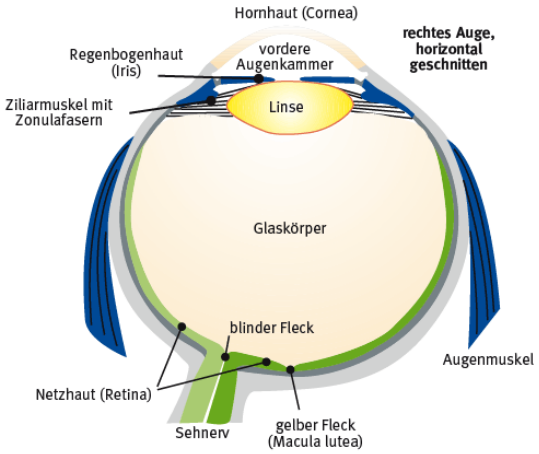
- **Amplitude**
  - Stärke einer Welle, höhere Amplitude = helleres Licht
- **Polarisation**
  - Horizontal/Vertikal oder Links/Rechts rotierend
  - Spielt im Alltag kaum eine Rolle
    * Sonnenlicht kann in der Atmosphäre polarisiert werden
    * einige Tiere können das wahrnehmen und zur Navigation nutzen
    * Polfilter Kameras gegen Spiegelungen
    * 3D-Darstellung (Stereoskopie)
    * Ein Auge bekommt vertikal polarisiertes Licht, das andere horizontal polarisiertes (bzw. unterschiedliche zirkulare Polarisation)

# 2 Wahrnehmen

**Das menschliche Auge**

- Komplexes Sinnesorgan
- Über die Linse wird das Licht auf die innen liegende Netzhaut projiziert
- Etwa 120 Millionen Sehzellen
  - ca. 113 Millionen Stäbchen, die nur Helligkeit wahrnehmen
  - ca. 7 Millionen Zapfen, die jeweils auf blaue, grüne, rote Farbtöne reagieren
- In der Tierwelt: Arten mit 2-4 Arten von Zapfen
- Bei Frauen tetrachromatisches Sehen möglich, allerdings nur ein Fall bekannt
- Ca. 8% der Männer, 1% der Frauen sind farbenblind
  - Verschiedene Formen
  - Meist können weniger Farben unterschieden werden
  - Häufigste Form: Rot-Grün-Schwäche

### Schema



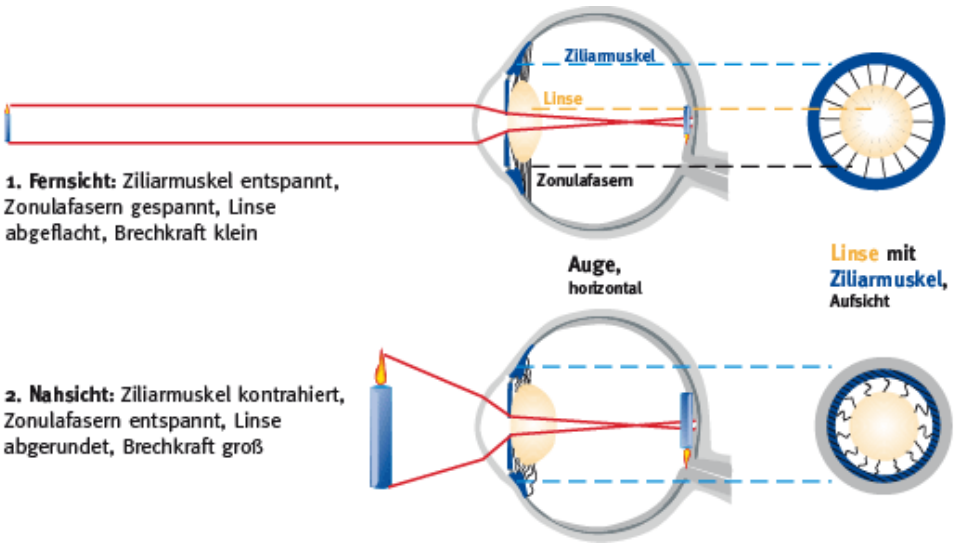

Bildquelle: ☞ www.tg8.eu

### Wichtige Bestandteile

- **Iris**: Reguliert die Menge des einfallenden Lichts
- **Linse**: Fokussiert ein kleines Abbild der Realität auf die Netzhaut
- **Netzhaut**: ist mit Lichtsinneszellen bedeckt die optische Reize (Stäbchen: Helligkeit, Zapfen: Farbe) wahrnehmen.
- **Gelber Fleck**: größte Dichte an Farbrezeptoren (17°des Sichtfeldes)
- **Fovea Centralis**: Bereich des schärfsten Sehens in der Mitte des Gelben Flecks (1.5-2°des Sichtfeldes)
- **Blinder Fleck**: hier verlässt der Sehnerv das Auge

### Fern- und Nahsehen

- Die Linse muß durch Muskeln abhängig von der Entfernung des Objektes eingestellt werden
- Der Prozeß heißt Akkomodation




Bildquelle: www.tg8.eu

**Helligkeit**

- Adaptation an unterschiedliche Helligkeit sehr wichtig
  - Von Nacht zu Sonnenschein etwa Faktor $10^{10}$
- Hell-Dunkel Adaptation: langsam (30-45 min) (abgedunkelte Kabine bei Nachtlandung von Flugzeugen)
  - Physiologische Änderung der Größe der Pupille
    * Etwa 1 sec, nur etwa Faktor 16-30
  - Neuronale Umschaltung auf Stäbchen
    * Bei Tageslicht aktive Zapfen hemmen die retinale Weiterverarbeitung der empfindlicheren Stäbchen
    * Vergrößerung der rezeptiven Felder der Ganglienzellen
    * Reduktion der Auflösung durch Verschaltung von Zellen
    * "Nachts sind alle Katzen grau"
  - Chemische Änderungen in den Sehzellen
    * Verstärkte Produktion von Sehpigmenten (photosensitiven Substanzen in den Sehzellen)
    * Nach etwa 2 Stunden ist die Empfindlichkeitsschwelle der Stäbchen etwa 15.000-mal geringer als die der Zapfen (1 Photon/500 sec gegenüber 3 Photonen /sec)
- Dunkel-Hell Adaptation: wesentlich schneller
  - Physiologisch: Pupille verkleinert sich
    * Zusammenkneifen der Augen
  - Neuronal: Aktive Zapfen übernehmen von Stäbchen
  - Chemisch: Reduktion der Sehpigmente
    * Änderung der Produktion der Pigmente
    * Änderungen in der Zellmembrane verringern die Empfindlichkeit

**Gesichtsfeld**

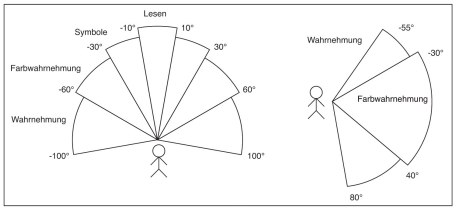
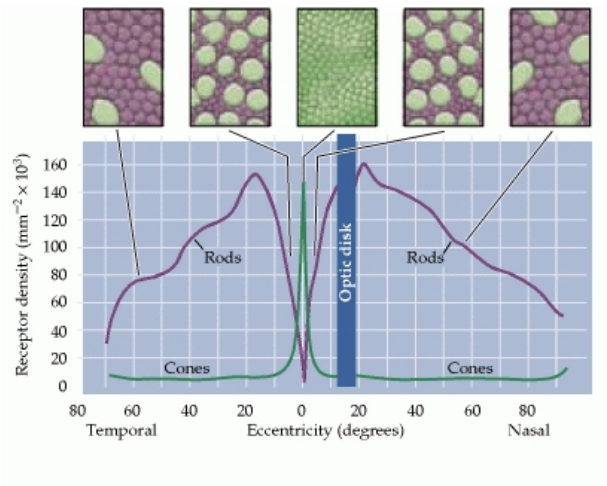



**Abbildung 1.1:** Sehfeld beim Menschen. Links horizontal, rechts vertikal (nach Herczeg, 1994)

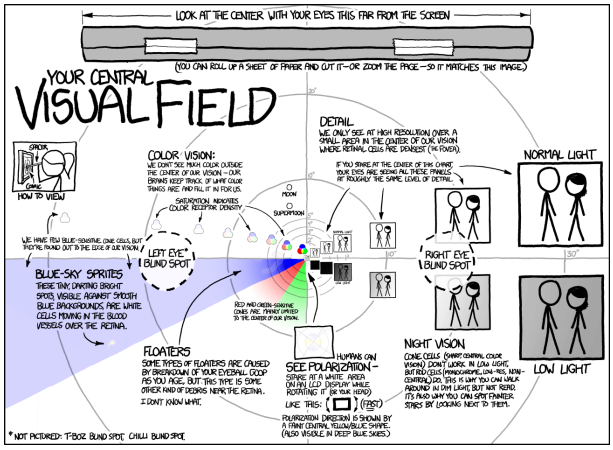
- Höchste Auflösung in der Mitte des Sehfeldes (ca 2 Grad)
  - Etwa 1 Bogenminute (1/60 Grad)
  - Ein Objekt von 3 mm Größe in ca 10 m Entfernung
- Dort finden sich viele Zapfen, aber kaum Stäbchen
- Nachts sind wir im Zentrum des Sehfeldes faktisch blind
- In der Peripherie ist das Sehen stark eingeschränkt, bis 40 Grad Augenbewegung, sonst Kopf drehen

**Verteilung Zapfen & Stäbchen**




Purves et al. [2001]

### Gesichtsfeld (contd.)




☞ xkcd 1080: Visual Field

### Augenbewegung

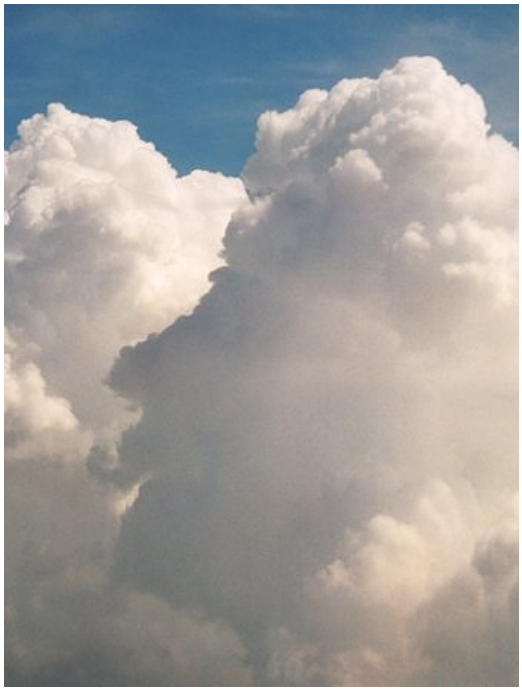
- Texte, die wir lesen, oder Objekte, die wir betrachten, liegen häufig außerhalb der 2 Grad scharfen Sehens
- Augenbewegungen bringen den schärfsten Punkt dahin, wo die Aufmerksamkeit liegt
- Ruckartige Bewegungen (Sakkaden)
  - Sprung von 50 ms zum neuen Fixationspunkt
  - Ruhe für 250 ms bis 2 s
  - Unbewußt, wir denken, daß es fließend geschieht

### Zeitliche Auflösung

- Visuelle Reize werden ab einer Dauer von 15 bis 50 ms wahrgenommen
- Sobald genug Einzelbilder nacheinander erscheinen, nehmen wir sie als Bewegung wahr
- Flimmern kann auftreten bei unter 20 bis 70 Hz
- Fließende Bewegung: 20 bis 25 Bilder pro Sekunde
  - Aber: schon ab 3 bis 5 Bildern pro Sekunde werden Bewegungen wahrgenommen
- Interpretation der Reize des Auges aufwendige kognitive Leistung

**Wissensbasiertes Sehen**

- Vertraute Konturen, wichtige Dinge besser erkennen
- Menschen, Gesichter
- Daher sehen wir teilweise Gesichter, wo keine sind



Bildquelle: Annegret Richter

**Tiefenwahrnehmung**

- Zum einen stereoskopisch
- Unterschiedliche Bilder des rechten und linken Auges
- Verschiebungen geben Rückschlüsse auf die Tiefe
- Zum anderen: Wissensbasierte Verarbeitung
  - Verdeckungen
  - Schatten
  - Graduelle Unterschiede in der Textur
  - Wissen z.B. über die Größe von Objekten

**Täuschung**

- "Optische Täuschungen" sind weniger ein Beispiel für Fehler als für neuronale Verschaltung und deren Effektivität und Effizienz

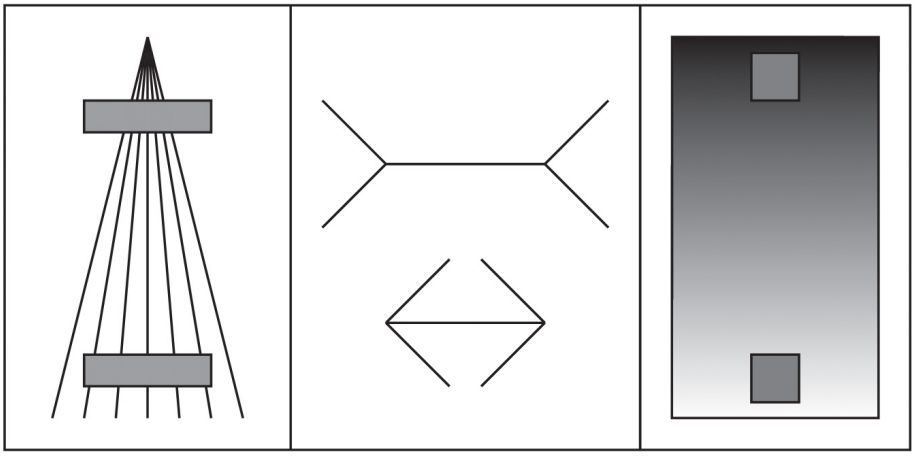
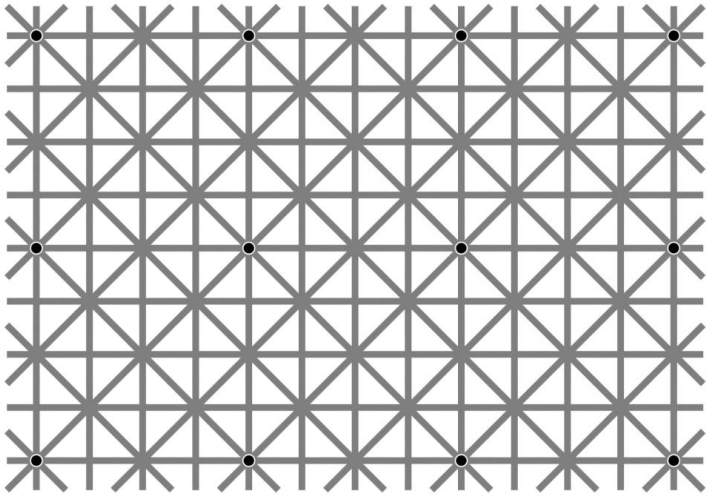


**Abbildung 1.4:** Optische Täuschungen. Links: Bei der Ponzo-Illusion scheint der obere Balken größer als der untere, beide sind aber gleich groß. Mitte: Muller-Lyer-Illusion: Auch hier scheint die obere waagerechte Linie größer. Rechts: Abhängig von der umgebenen Helligkeit scheint das graue Quadrat einmal heller und einmal dunkler zu sein.

### Extinction Grid

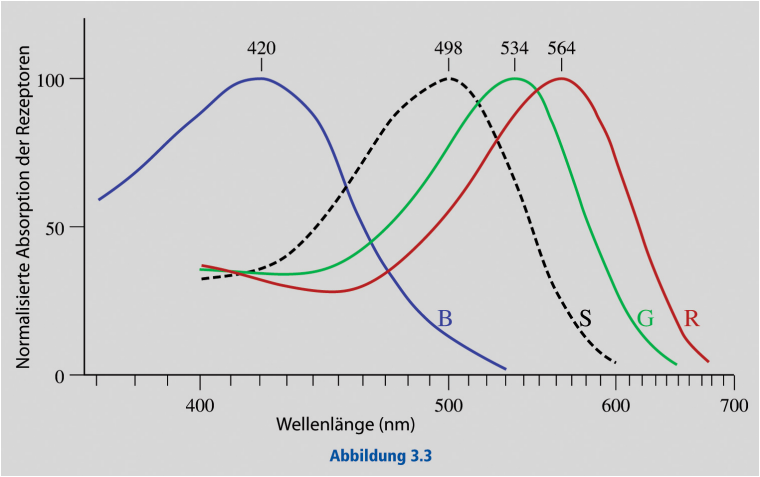


Ninio and Stevens [2000]

### Farben im Kopf

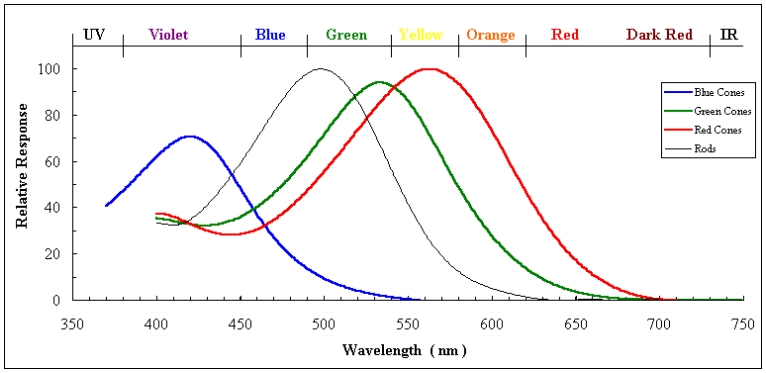
- Viele Eigenheiten entstehen durch Art und Weise der Wahrnehmung
- Nervenzellen in den Augen verwandeln Lichtwellen in neuronale Aktivitäten
- Rekapitulation: Netzhaut
- Stäbchen (Helligkeit), drei Arten von Zapfen (Farben)
- Rot/Grün Wahrnehmung recht ähnlich

### Empfindlichkeit Rezeptoren I




Achtung, normalisierte Werte

### Empfindlichkeit Rezeptoren II

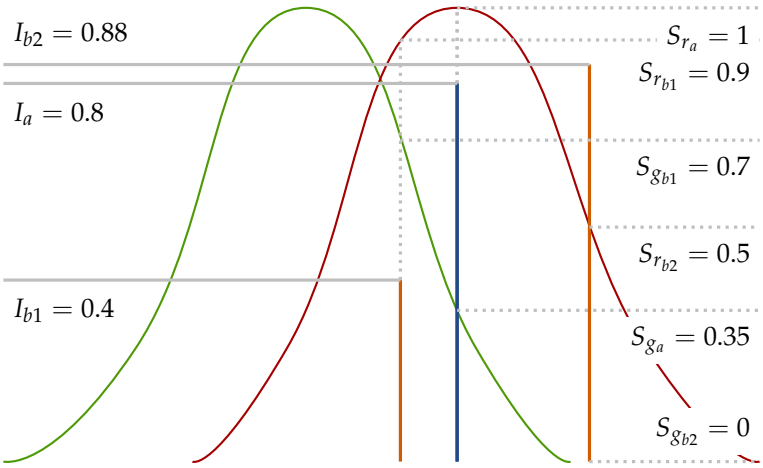



☞ www.unm.edu/~toolson/human_cone_response.htm

### Auflösung nach der Frequenz

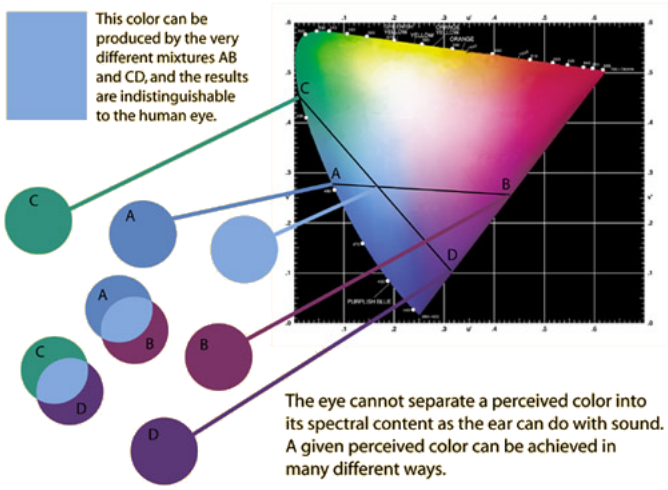
- Können wir also alle möglichen Wellenlängen unterscheiden?
- Nein, Wahrnehmung von Farben dramatisch reduziert
- Farben entstehen im Kopf, nicht in den Augen
- Problem: Überlagerung von Wellen
- Überlagerung schafft komplexes Signal mit den Ausgangsfrequenzen präsent
- Töne nehmen wir in ihrer Komplexheit wahr (Akkorde)
  - In der Fourier-Transformierten Anteile auch sichtbar
- Farben werden über die Sinneszellen integriert

### Integration über verschiedene Wellenlängen




Erregungspotential der Sehzellen: $P_c(x) = I(x)S_c(F(x))$
$P_r(a)$=0.8, $P_r(b)$=0.36+0.44, $P_g(a)$=0.28, $P_g(b)$=0.28+0

**Erzeugung von Farbeindrücken**



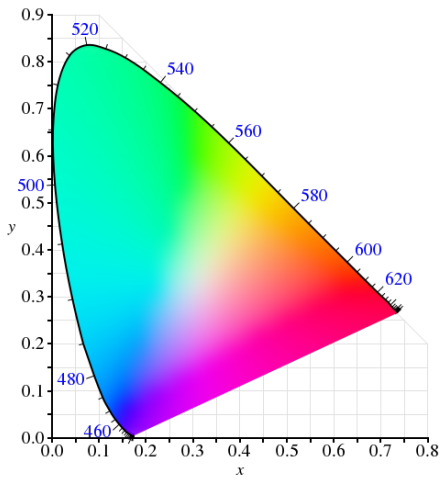

☞ hyperphysics.phy-astr.gsu.edu/hbase/vision/colper.html

# 3 Farbräume

**CIE und die drei Grundfarben**

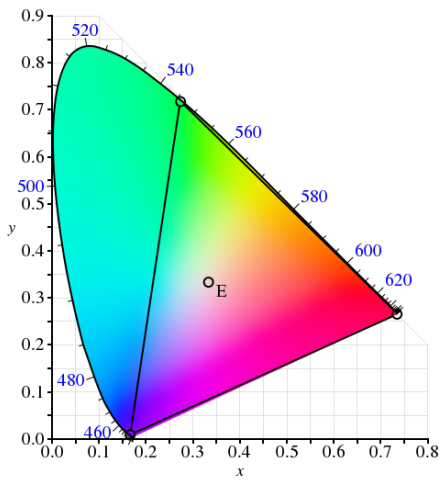
- Kann "jeder" Farbeindruck aus maximal drei Grundfarben erzeugt werden?
- Allgemeiner: Welche Farben kann der Mensch standardisiert sehen?
- Eine der Fragen der CIE
  - Commission internationale de l'éclairage (Internationale Beleuchtungskommission)
- Internationale, nichtstaatliche Non-Profit-Organisation
- Unter anderem: CIE-Normfarbsystem der wahrnehmbaren Farben

**CIE Farbraum**



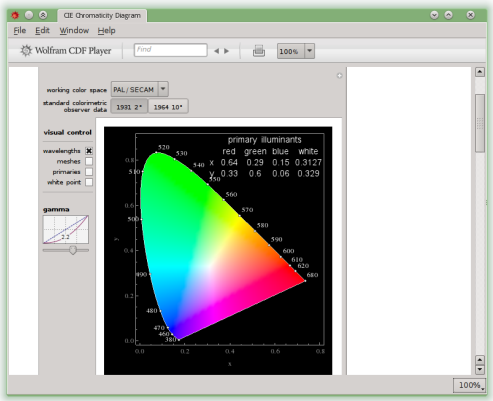

☞ Wikipedia-User BenRG

### CIE Farbraum RGB



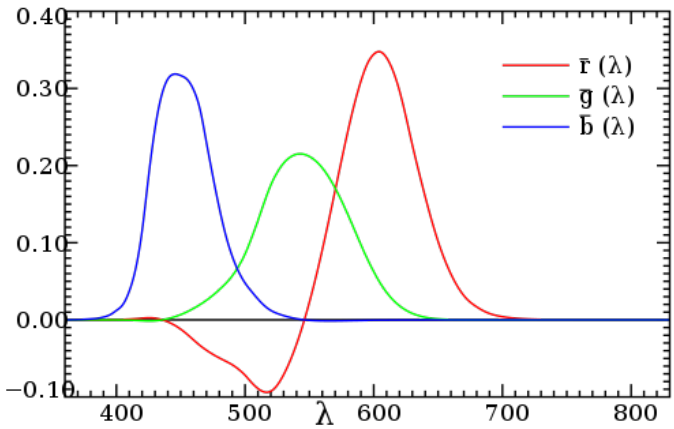

☞ Wikipedia-User BenRG

### CIE: Weitere Beispiele




☞ demonstrations.wolfram.com/CIEChromaticityDiagram/

### CIE Tristimuluskurven




☞ Wikipedia-User Marco_Polo

## 3.1 RGB/CMY(K)

**Additive Farbmischung**

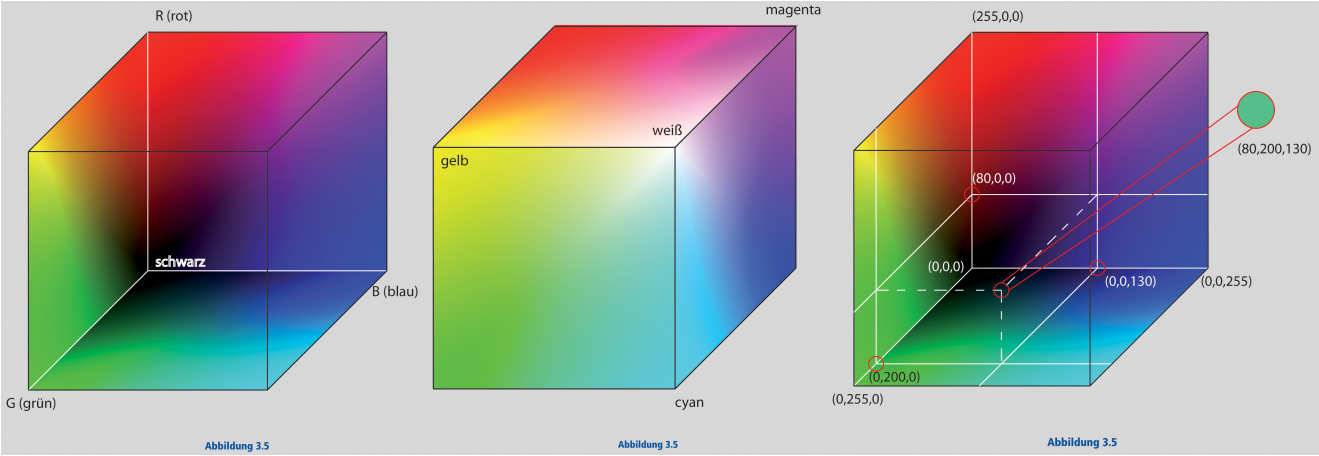
- Wir können die wahrnehmbaren Farben als Mischung von Rot, Grün und Blau herstellen
- Unser Weiß ist Rot + Grün + Blau
- **Additive** Farbmischung:
- Durch Hinzufügen von Farben zu Schwarz werden neue Farben erzeugt
  - Rot + Grün = Gelb
  - Rot + Blau = Magenta
  - Grün + Blau = Cyan
- **Farbraum**: Mischbare Farben

Original

RGB

RGB

**RGB-Würfel**




**Subtraktive Farbmischung**

- Farbeindruck auf Papier entsteht durch Filterung von Farbanteilen
- Grün ist die Abwesenheit von Blau und Rot
  - Blau und Rot werden absorbiert
- Mischung auf Papier heißt somit: mehr Filterung
- **Subtraktive** Farbmischung
- Im RGB-Farbwürfel werden jeweils die gegenüberliegenden Ecken gefiltert
  - Cyan: Rot wird gefiltert
  - Gelb: Blau wird gefiltert
- CMY-Modell: Cyan Magenta Yellow

Original

RGB

CMY

Original

RGB

CMY

**Umwandlung**

- Umrechnung des additiven und des subtraktiven Farbmodells
- Additiv: RGB-Modell
- Subtraktiv: CMY-Modell
- Einfache Betrachtung von der gegenüberliegenden Seite des Farbwürfels

$$f_{CMY} = \begin{pmatrix} f_C \\ f_M \\ f_Y \end{pmatrix} = weiß_{RGB} - \begin{pmatrix} f_R \\ f_G \\ f_B \end{pmatrix} = weiß_{RGB} - f_{RGB}$$

$$f_{RGB} = \begin{pmatrix} f_{max} - f_C \\ f_{max} - f_M \\ f_{max} - f_Y \end{pmatrix} = \begin{pmatrix} f_{max} \\ f_{max} \\ f_{max} \end{pmatrix} - \begin{pmatrix} f_C \\ f_M \\ f_Y \end{pmatrix} = weiß_{RGB} - \begin{pmatrix} f_C \\ f_M \\ f_Y \end{pmatrix}$$

**Umwandlung: Beispiel**

- Annahme: Farbwerte pro Farbkanal mit 8 Bit
- Türkis in RGB: (80, 200, 130)
- Dann Umrechnung in CMY:

$$f_{CMY} = \begin{pmatrix} f_C \\ f_M \\ f_Y \end{pmatrix} = weiß_{RGB} - \begin{pmatrix} f_R \\ f_G \\ f_B \end{pmatrix} = \begin{pmatrix} 255 \\ 255 \\ 255 \end{pmatrix} - \begin{pmatrix} 80 \\ 200 \\ 130 \end{pmatrix} = \begin{pmatrix} 175 \\ 55 \\ 125 \end{pmatrix}$$

**Varianten**

- Eine Variante von CMY ist das CMYK-Modell
  - Hinzufügen von Schwarz (Black)
  - Bessere Farbdarstellung (Mischung aus CMY eher braun)
  - Kostenfaktor
- Jedes Gerät hat andere Grundfarben
- Können abhängig von Umgebung verschieden wirken
- Daher: weitere Farbprofile
- Farben können auch auf unterschiedlichem Papier unterschiedlich aussehen
- Daher muß der Drucker die (CMY-) Komponenten umrechnen und an seine Grundfarben anpassen
- Dafür muß der Drucker "kalibriert" werden
  - Messen, welche Farbwiedergabe tatsächlich erzielt wird
  - Das Ergebnis der Messung ist ein sogenanntes Farbprofil
- Entsprechendes gilt für Monitore etc.

**Kalibrierung, Profile**

☞ Beschreibung ColorHug

Original

CMY

CMYK

Original

CMY

CMYK

## 3.2 Andere

**Weitere Modelle**

- Im wesentlichen zwei Gruppen:
  - Varianten von RGB/CMY(K) (Grundfarbenorientierte Modelle)
  - Luminanz-Chrominanz (Wahrnehmungsorientierte Modelle)
- Letztere arbeiten nicht mit Grundfarben, sondern trennen **Farbinformation** von **Helligkeitsinformation**

- Analog menschliche Wahrnehmung: Integration über die Antwort der Zapfen
- Vorteil: Anpassung an unterschiedliche Helligkeiten
- Bei RGB und CMY durch Änderung eines Farbwertes auch Änderung von Helligkeit und Sättigung
- Auch: Schwarzweißbild + Farbinformationen
- Umrechnung zum Teil aufwendig und nicht linear

**YCbCr**

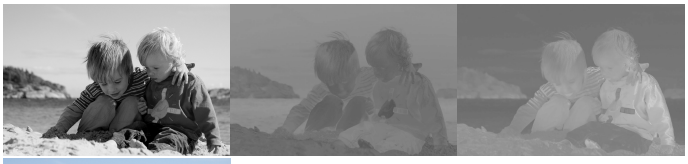
- Einfaches lineares Modell mit einem Helligkeitskanal: YCbCr
  - Y: Helligkeit, hoher Anteil von Grün, geringer Rot und Blau
  - Cb: Blauanteil
  - Cr: Rotanteil

$$f_{YCbCr} = \begin{pmatrix} f_Y \\ f_{Cb} \\ f_{Cr} \end{pmatrix} =$$

$$\begin{pmatrix} 0 \\ 128 \\ 128 \end{pmatrix} + \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.168736 & -0.331264 & 0.5 \\ 0.5 & -0.418688 & -0.081312 \end{pmatrix} \begin{pmatrix} f_R \\ f_G \\ f_B \end{pmatrix}$$

Original

RGB


YCbCr

Original

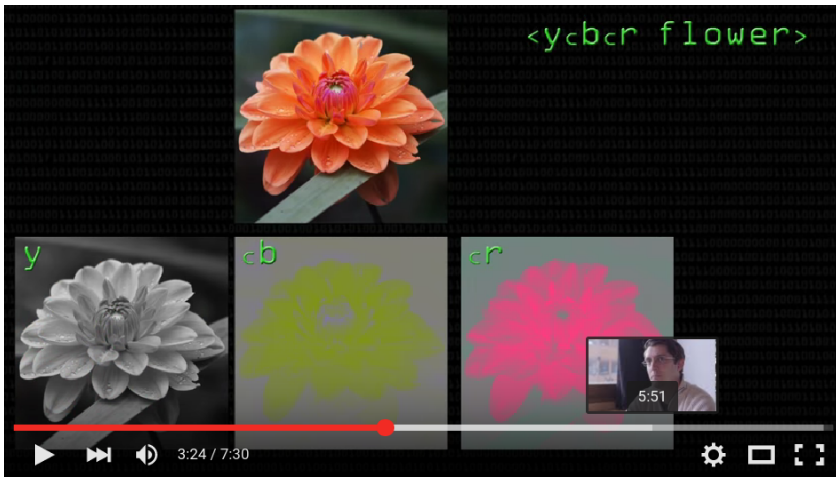
RGB

YCbCr

**Eigenschaften YCbCr**

- Cr und Cb viel "flächiger"
- benachbarte Bildpunkte haben oft den gleichen Wert
- zudem sind diese oft im mittleren Wertebereich
- Das Phänomen, das mehr Information in den Helligkeitswerten spricht, tritt bei vielen natürlichen Bildern auf – Farben sind gleichmäßiger verteilt als Helligkeiten
  - einheitliche Färbung bei dreidimensionaler Struktur
  - Licht und Schatten liefern Kontrastunterschiede
- Unser Sehsystem hat sich mit darauf eingestellt, es verarbeitet Helligkeitsinformationen besser
- Besonders deutlich nachts, wenn wir ohne Farben sehen
- Farbkanäle haben weniger Informationen, unser Sehsystem ist daran angepaßt

**Video 7.1: JPEG – Farbräume**



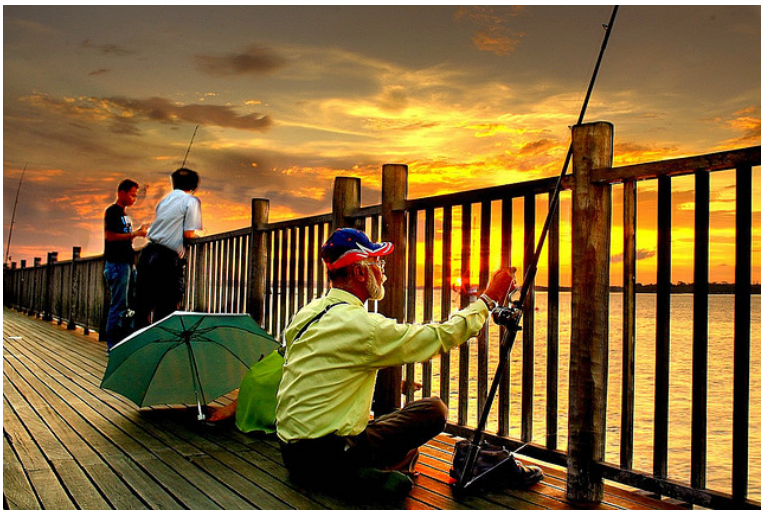

☞ Colourspaces – Computerphile (7:30)

# 4 Digitalisierung

**Digitalisierung und Kodierung**

- Digitale Bilder setzen sich aus Bildpunkten zusammen, den **Pixeln**
- Jedes Pixel hat Helligkeitsinformationen und Farbinformationen
- Typisch:
  - 2 Farben – 1 Bit – Schwarz-Weiß
  - 16 Farben – 4 Bit
  - 256 Graustufen – 8 Bit
  - 256 Farben – 8 Bit
  - 16,7 Millionen Farben – 24 Bit – "True Color"
- Weiterer Farbkanal: Alphakanal
- HDR (High Dynamic Range) je Farbkanal (mindestens) 32 Bit

**Beispiel HDR 1**



☞ cc by-nc-sa Kenny Teo

**Beispiel HDR 2**

☞ cc by-nc-sa Kenny Teo

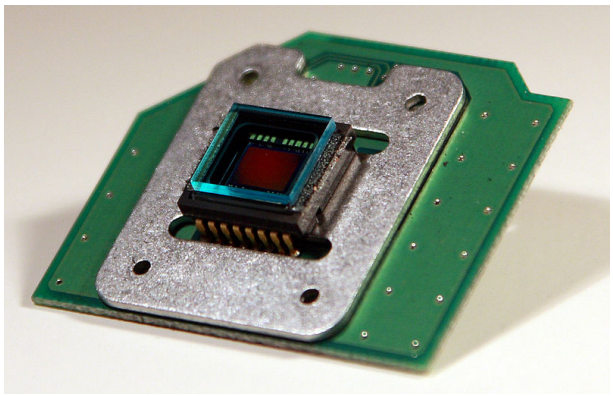
**Beispiel HDR 3**

☞ cc by-nc-nd kennytyy

**Typische Anwendungen**

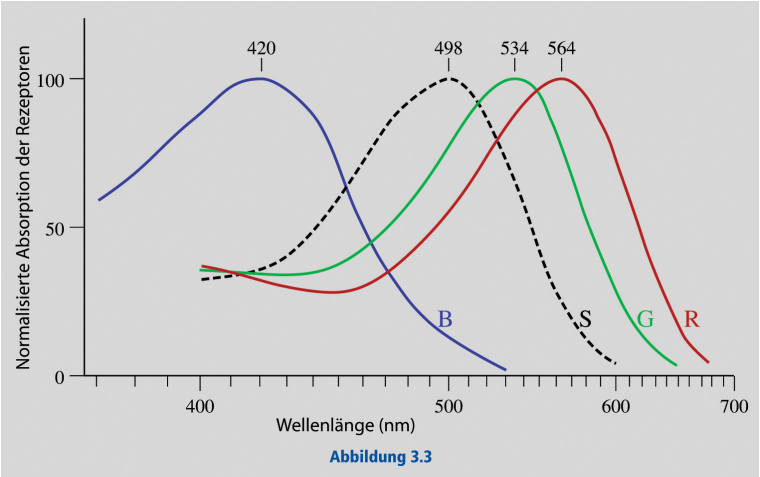
- Digitale Fotographie mit 24 Bit
- Im besonderen bei Graphiken mit reduzierter Farbzahl:
  - Farbpalette
  - Beispiel GIF mit Farbtabellen mit max. 256 Farben
  - Dazu eine Farbe als "Transparenzfarbe"
- Alternative: PNG
  - Echter Alphakanal
  - Farben als Paletten oder Farbkanal mit bis zu 24 Bit
- Fotos können je nach Auflösung und Farbtiefe erhebliche Größe erreichen
  - 10 Megapixel, 48 Bit pro Pixel = 480 Millionen Bit = 60 MByte
- Kompression sinnvoll
- Aber: Möglichkeiten des RAW-Formats

**Digitale Kamera**



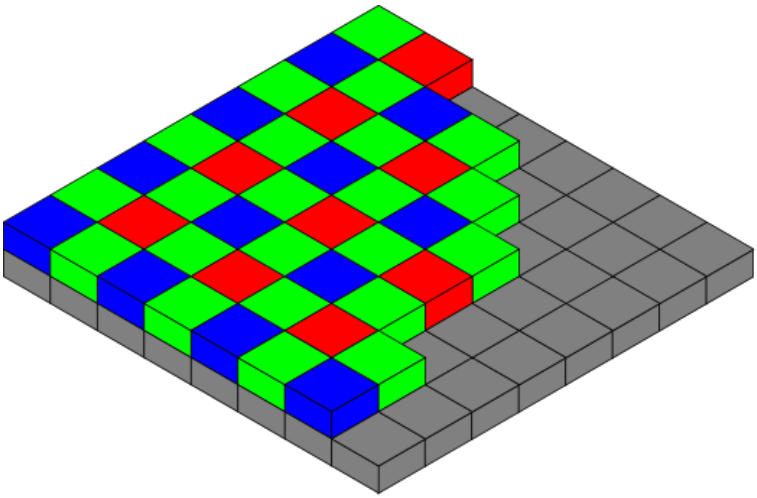
CCD-Sensor (Wikipedia-User Merzperson)

**Erinnerung: Farbwahrnehmung**




Achtung, normalisierte Werte

**Bayer-Filter**

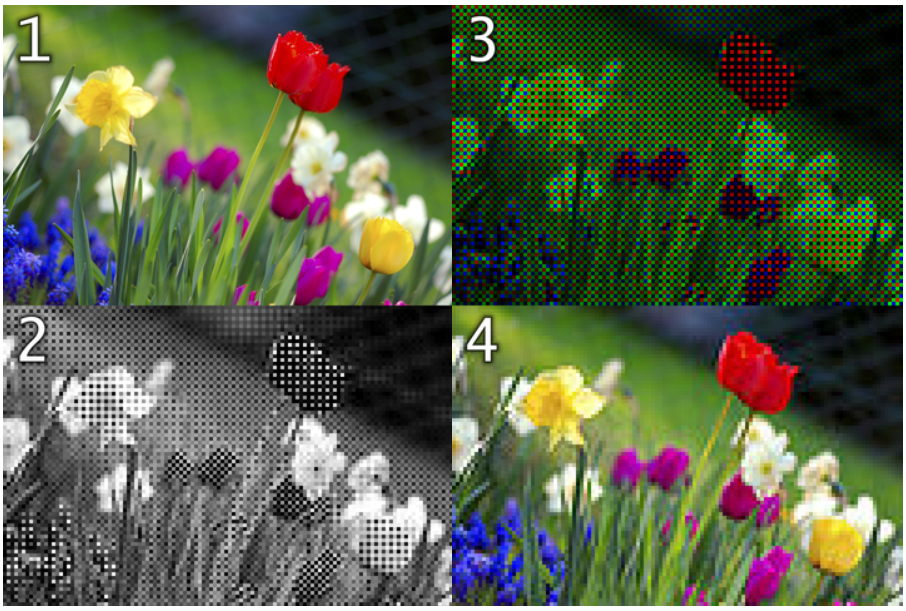


☞ Wikipedia-User Cburnett

**Demosaicing**

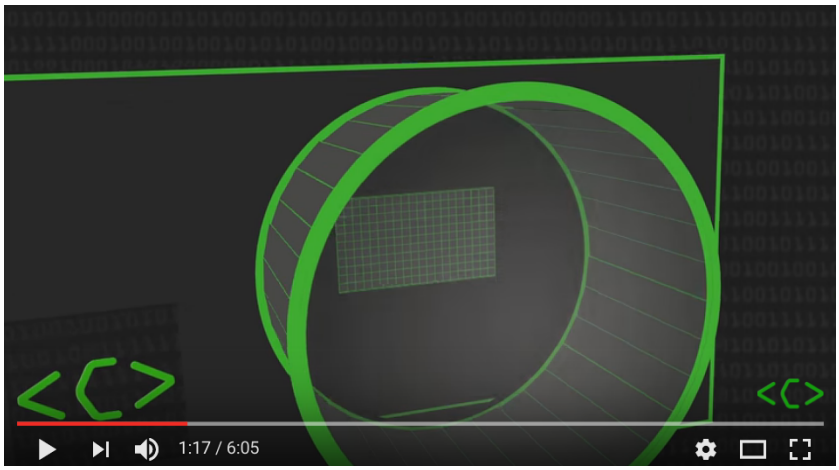
- Über den einzelnen lichtempfindlichen Sensoren (Pixel) liegen Farbfilter
  - Verschiedene Muster, Bayer-Filter das gebräuchlichste
- Die einzelnen Pixel enthalten jeweils nur Informationen für einen Farbkanal
- Das fertige Bild soll für jedes Pixel Informationen für alle Farben enthalten
- Lösung: Interpolation der nicht vorhandenen Farbwerte
  - Einfachster Ansatz: Interpolation aus umliegenden Pixeln
  - Komplexer: Annahmen über Bilder, z.B. Kantenerkennung
- Dieser Prozeß heißt Demosaicing und ist normalerweise Teil des Bildprozessors der Kamera, kann aber bei vorliegenden Rohdaten des Sensors (RAW-Format) im Nachhinein durchgeführt werden

**Bayer-Filter: Beispiel**



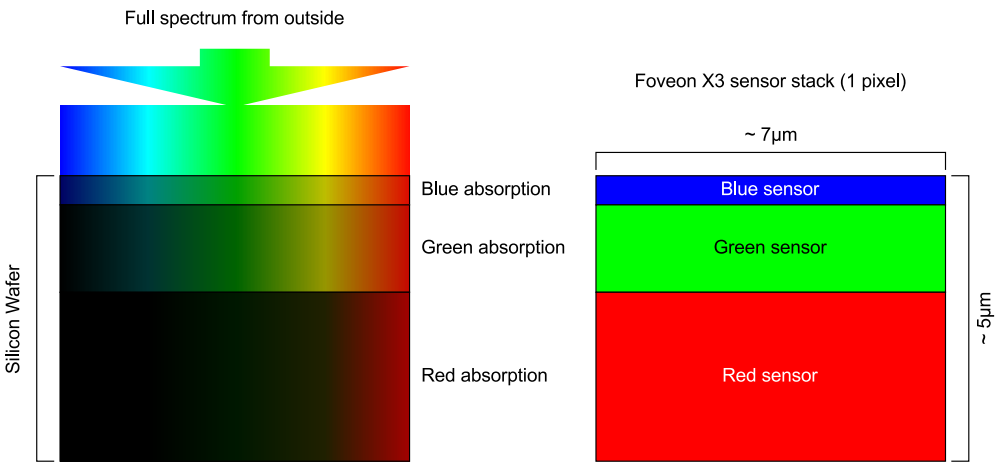

☞ Wikipedia-User Anita Martinz & Cmglee

### Video 7.2: Bayer-Filter



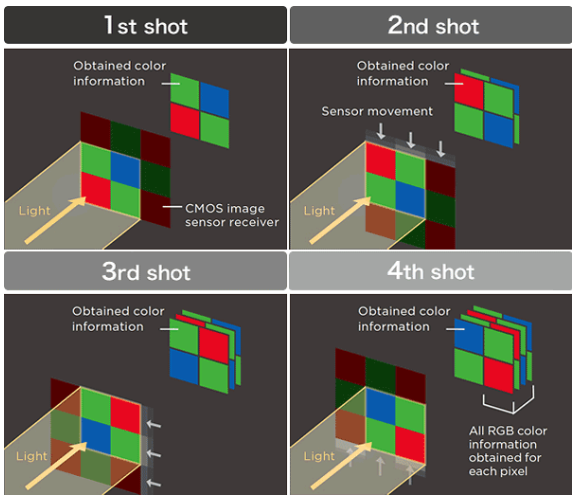

☞ Bayer Filter – Computerphile (6:05)

### Foveon-Sensor




☞ Wikipedia-User wdwd

### Pentax Pixelshift




☞ Imaging Resource

### Dithering

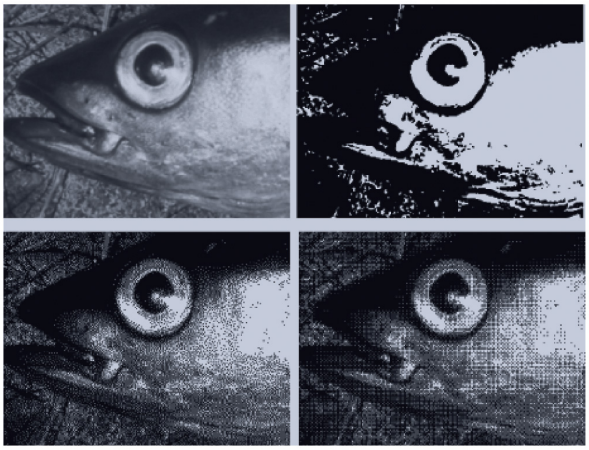
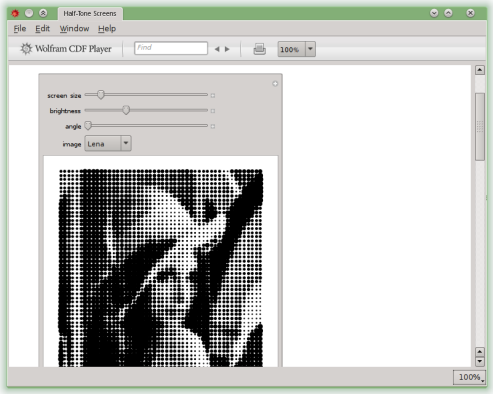


**Abbildung 3.7:** Dithering. Links oben: Originalbild mit 256 Grautönen. Rechts oben: Schwarzweißbild ohne Dithering. Link unten: Dithering mit Floyd-Steinberg-Verfahren. Rechts unten: Dithering mit festem Muster. Alle Bilder außer dem Originalbild haben 1 Bit Farbtiefe (Schwarz/Weiß).

**Dithering: Beispiel**




☞ demonstrations.wolfram.com/HalfToneScreens/

# 5 Kompression

**Kompression**

- Wirklich hohe Raten erst bei verlustbehafteten Verfahren
- GIF nicht verlustfrei, da reduzierte Farben
- Datenmenge abhängig von
  - Abtasthäufigkeit (Diskretisierung, Samplingrate, Auflösung)
  - Abtastgenauigkeit (Quantisierung, Farbtiefe)
- verlustbehaftete Verfahren setzen (eine) diese(r) beiden Größen herab, aber da, wo es weniger auffällt
- Ausnutzen lokaler Eigenschaften des Bildes
  - Globale Reduktion der Abtasthäufigkeit wäre eine Verkleinerung der "Auflösung" in Pixeln
  - Globale Reduktion der Abtastgenauigkeit wäre eine Reduzierung der möglichen Farben pro Pixel

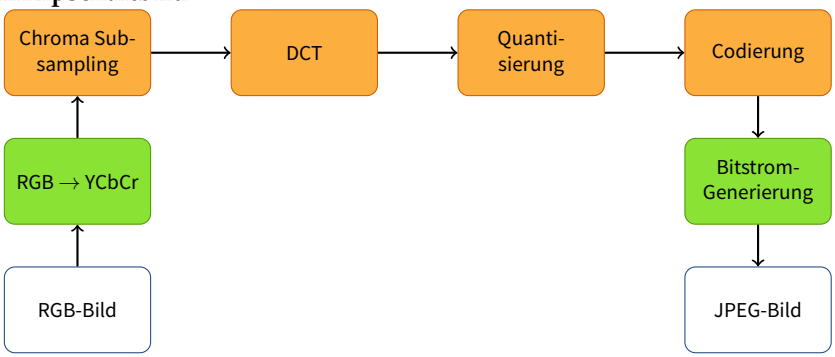
## 5.1 JPEG Verfahren

**JPEG**

- Joint Photographic Expert Group (ISO + ITU)
- Schritte:

- (Transformation RGB nach YCbCr)
- Chroma-Subsampling
- Umcodierung in den Frequenzraum
- Quantisierung
- Codierung mit weiterer Kompression
- (Bitstrom-Generierung)

**Prinzipschaltbild**




Vereinfachtes Prinzipschaltbild

**Transformation nach YCbCr**

$$f_{YCbCr} = \begin{pmatrix} f_Y \\ f_{Cb} \\ f_{Cr} \end{pmatrix} =$$

$$\begin{pmatrix} 0 \\ 128 \\ 128 \end{pmatrix} + \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.168736 & -0.331264 & 0.5 \\ 0.5 & -0.418688 & -0.081312 \end{pmatrix} \begin{pmatrix} f_R \\ f_G \\ f_B \end{pmatrix}$$

**Chroma-Subsampling**

- Verlustbehafteter Kompressionsschritt
- Abtastrate eines Teils des Bildes wird reduziert
- Idee: Flächigkeit der Farben im YCbCr
- Verschiedene Varianten
  - Block aus 4 Pixeln zusammenfassen
  - Zwei benachbarte Pixel zusammenfassen (horizontal oder vertikal)
- Reduziert Cr und Cb auf die Hälfte/ein Viertel
- Y-Kanal unverändert
- Subsampling nur auf den Farbkanälen
- Auswählbar: Gar nicht/Halbierung/Viertelung

### Detail Chroma-Subsampling

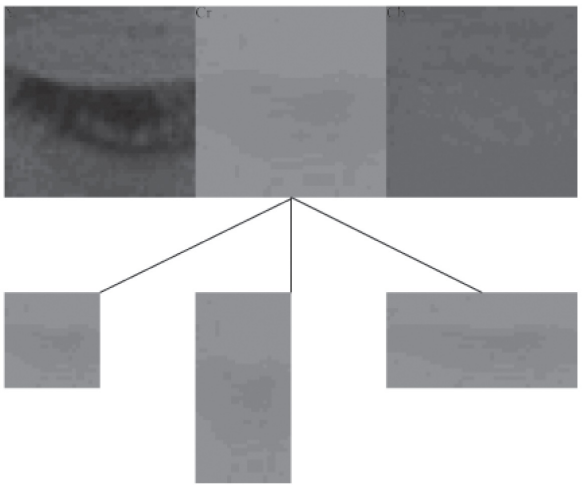
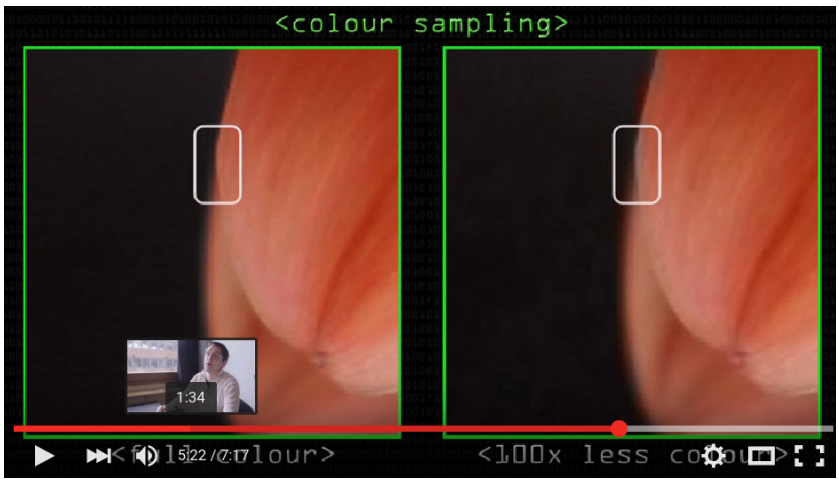



**Abbildung 3.8:** Vergrößerung aus ▶Abbildung 3.6, Y-, Cr-, Cb-Kanal, jeweils 48x48 Pixel und Chroma-Subsampling für den Cr-Kanal auf 24x48, 48x24 und 24x24 Pixel (zweite Zeile)

### Video 7.3: JPEG – Farben




☞ JPEG 'files' & Colour – Computerphile (7:17)

### Umcodierung in den Frequenzraum

- Folgende Schritte für die drei Kanäle getrennt
- Y, Cr, Cb unabhängig voneinander untersucht und komprimiert
- Grauwertbilder, jeweils ein Byte pro Pixel (255 Stufen)
- Umcodiert, um danach die Quantisierung (Genauigkeit der Abtastung) zu vergröbern
- Im Frequenzraum, um die in einem Bild vorkommenden Frequenzen zu betrachten

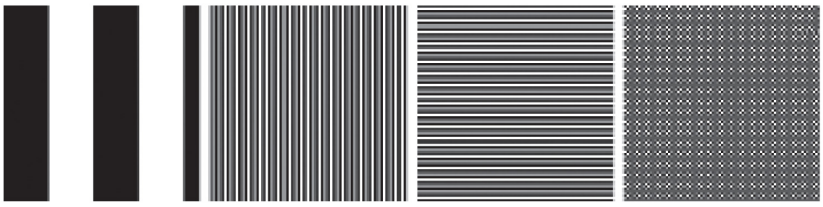


**Abbildung 3.9:** Bilder mit unterschiedlichen horizontalen und vertikalen Frequenzanteilen, von links nach rechts: (1) niedrige horizontale Frequenzen, (2) nur hohe horizontale Frequenzen, (3) nur hohe vertikale Frequenzen, (4) Mischung hoher horizontaler und vertikaler Frequenzen

**Frequenzraum bei Bildern**

- Ähnlich wie Audiosignale, die als Überlagerung einfacher Schwingungen betrachtet werden können, lassen sich auch Bilder aus einfachen Bildern (Mustern) zusammensetzen
- Insbesondere aus aus einer Grundmenge von nur horizontalen/vertikalen Frequenzen lassen sich alle möglichen Bilder zusammensetzen
- Jedes Pixel hat nur eine endliche Anzahl möglicher Grauwerte, und es gibt eine endliche Anzahl von Pixeln

**Umkodierung durch DCT**

- Transformation mit Hilfe der diskreten Cosinustransformation (DCT)
- Bei JPEG für 8x8 Pixel große Teilblöcke
  - Prinzipiell auch andere Blockgrößen möglich, JPEG beschränkt sich auf 8x8
- jeder 8x8 Block enthält 64 Werte
- Mit Hilfe der DCT werden diese in 64 neue Werte, die DCT-Koeffizienten $F(u,v)$, umgerechnet
  - Bildraum: $f(x,y)$ gibt für das Pixel an Stelle $(x,y)$ innerhalb des betrachteten Blocks den Grauwert an
  - Frequenzraum: Die Koeffizienten geben an, mit welchem Anteil das entsprechende Basismuster zum Block beiträgt

**Umkodierung: Basismuster**

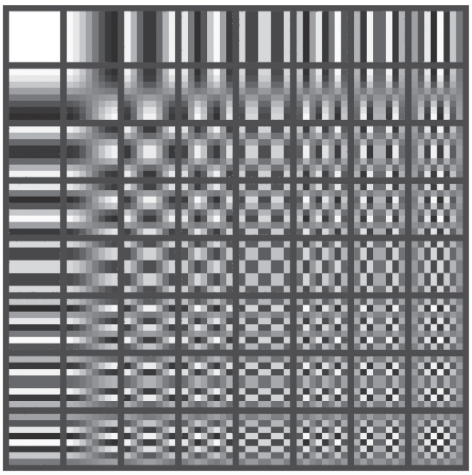


**Abbildung 3.10:** 64 Basismuster für die DCT, jedes Basismuster besteht aus 8x8 Werten, die unterschiedliche Frequenzanteile des Bildes repräsentieren. Ganz oben links ist die DC-Komponente. Die Muster weiter rechts entsprechen höheren horizontalen, die weiter unten höher werdenden vertikalen Frequenzen.

**Umkodierung: DCT AC/DC-Komponenten**

- DCT-Koeffizient $F(0,0)$ berechnet den Mittelwert aller Pixel innerhalb des Teilbildes
  - Bei einfarbigen Bildern ist dies der einzige Anteil, der nicht Null ist
- Dieser Koeffizient wird DC-Komponente genannt, die anderen AC-Komponenten
- DCT reduziert den Speicherbedarf noch nicht (64 Koeffizienten statt 64 Pixel)
- DCT plus inverse verlustfrei (modulo Rundungsfehler)
- Warum die DCT trotzdem bei der Komprimierung hilft läßt sich nachvollziehen, wenn man die typischen Werte der Koeffizienten betrachtet

**Umkodierung: DCT Koeffizienten**

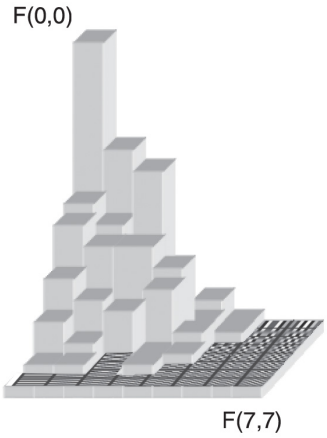



**Abbildung 3.11:** DCT-Koeffizienten. Links: typische Werteverteilung bei Bildern. Der DC-Anteil F(0,0) ist hoch. Hohe Frequenzanteile (vorn und rechts) sind niedrig und oft null. Rechts: Zick-Zack-Scan zum Auslesen der AC-Koeffizienten.

**Quantisierung: Kompression im Frequenzraum**

- Annahmen
  1. Bei vielen Bildern sind hohe Frequenzanteile seltener als niedrige
  2. Änderungen bei hohen Frequenzanteilen fallen bei der Wahrnehmung weniger ins Gewicht
- Vorgehen
  - Werte nahe Null auf Null setzen
  - Höhere Frequenzen mit weniger Bits kodieren (Auflösung reduzieren)
    - Quantisierung (Auflösung) wird vergröbert
    - Nur auf weniger gut wahrgenommenen (hohen) Frequenzanteilen
  - Daher Transformation in den Frequenzraum

**Re-Quantisierung**

- Die neue Quantisierung der DCT-Koeffizienten wird mit einer Quantisierungsmatrix durchgeführt
- Dabei gibt es für jeden der 8x8 DCT-Koeffizienten einen Wert $Q(u,v)$, durch den dieser geteilt wird

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

- Rundung reduziert Wertebereich für Koeffizient $F(u,v)$
- Wesentlicher Schritt, der auch die Güte bestimmt
  - Werte $Q(u,v)$ bestimmen, wie viele Informationen verloren gehen

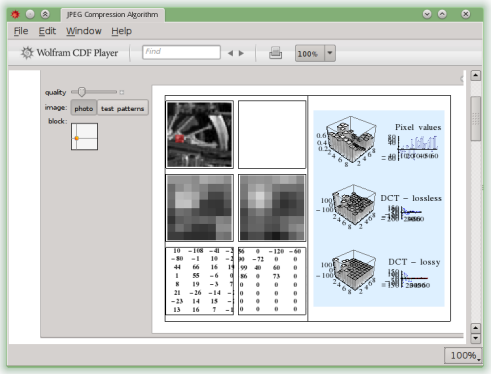
**Quantisierungsmatrix**

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

$$Q = \begin{pmatrix} 10 & 20 & 40 \\ 20 & 30 & 50 \\ 40 & 50 & 60 \end{pmatrix} F = \begin{pmatrix} 105 & 63 & 11 \\ 61 & 14 & 8 \\ 13 & 9 & 2 \end{pmatrix}$$

$$F' = \begin{pmatrix} 10 & 3 & 0 \\ 3 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

### JPEG: Beispiel




☞ demonstrations.wolfram.com/JPEGCompressionAlgorithm/

### Quantisierung: Rekonstruktion

- Für die Rekonstruktion muß die verwendete Quantisierungsmatrix bekannt sein:

$$\tilde{F}(u,v) = F'(u,v) \cdot Q(u,v)$$

- Es gibt vordefinierte Quantisierungsmatritzen oder selbst definierte
- Aufgrund der verlustbehafteten Rundung bei der Konstruktion von $F'(u,v)$ werden die rekonstruierten Werte nicht den originalen entsprechen
  - Erneuter Kompression mit gleicher Quantisierungsmatrix führt nicht zu weiteren Verlusten

### Bitstromgenerierung/Codierung

- Subsampling und re-Quantisierung sind die verlustbehafteten Schritte
- Bei Speicherung ergänzt um verlustfreie Verfahren, DC-Komponente anders als andere
- Kodierung DC
  - Da benachbarte Blöcke oft ähnliche Grundtöne haben kann prädiktive Kodierung helfen – von einem Startwert werden nur Differenzen gespeichert
  - weniger Bits für kleinere Differenzwerte nötig – Huffman, andere entropiebasierte Codierung
  - DC-Komponenten ergeben ein vergröbertes Vorschau-Bild
- Kodierung AC
  - Zick-Zack-Scan, zuerst niedrige, dann hohe Frequenzen
  - Am Ende oft viele Nullen – Lauflängenkodierung
  - Danach Huffman (oder arithmetisch)
  - Standardverteilungen für Huffman vorhanden

## 5.2 DCT Details

### DCT Details I

- Verschieben der Werte $f(x,y)$ von 0 bis 255 zu -128 bis 127

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

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

- $x, y$ Koordinaten im 8x8 Teilbild $u, v$ Indizes der 64 DCT Koeffizienten
- Leicht berechnet, da Cosinus-Werte und Werte $c_k$ nur einmal berechnet werden müssen

**DCT Details II**

- Man kann für die 64 möglichen Werte von $x, y$ und die 64 möglichen Werte von $u, v$ die 4096 Parameter $T(x, y, u, v)$ berechnen

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

- Dadurch wird die DCT zu:

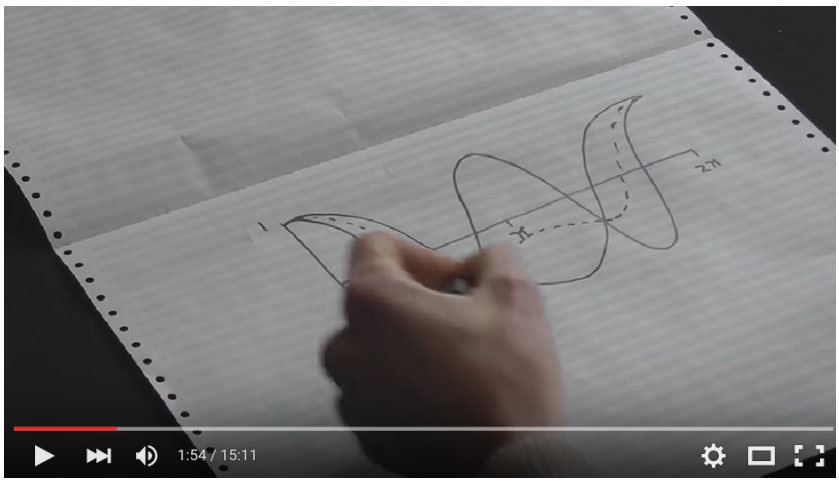
$$F(u, v) = \sum_{x=0}^{7} \sum_{y=0}^{7} T(x, y, u, v) f(x, y)$$

**DCT Details III**

- Ein Teilbild muß also immer nur mit einer bestimmten Matrix bzw. Basismuster multipliziert werden, um die jeweiligen DCT Koeffizienten zu berechnen
- $F(u, v)$ gibt an, wie stark das Muster aus horizontalen und vertikalen Frequenzen in diesem Teilbild vorkommt
- Zur Rekonstruktion genügt es, das Bild aus den (über die Koeffizienten gewichteten) Anteilen der Basismuster zusammenzusetzen

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

**Video 7.4JPEG – DCT**




☞ JPEG DCT – Computerphile (15:11)

## 5.3 JPEG Beispiel

**Beispiel JPEG-Kompression**

- Beispielbilder für JPEG-Kompression
- Alle Bildmanipulationen mit GIMP 2.8.2
- Skaliert von Originalgröße im Lanczos-Verfahren
- Gespeichert mit zwei Variablen:
  - "Qualitätsfaktor" von 0-100
  - Chroma-Subsampling

**Beispiel JPG**

Vergleich losless PNG, 403K

**Beispiel JPG**

JPEG, 100, kein Chroma-Subsampling, 348K

**Beispiel JPG**

JPEG, 85, kein Chroma-Subsampling, 78K

**Beispiel JPG**

JPEG, 85, Chroma geviertelt, 60K

**Beispiel JPG**

JPEG, 15, kein Chroma-Subsampling, 23K

**Beispiel JPG**

JPEG, 15, Chroma geviertelt, 21K

**Beispiel JPG**

JPEG, 10, kein Chroma-Subsampling, 19K

**Beispiel JPG**

JPEG, 10, Chroma geviertelt, 17K

**Beispiel JPG**

JPEG, 5, kein Chroma-Subsampling, 14K

**Beispiel JPG**

JPEG, 5, Chroma geviertelt, 12K

**Beispiel JPG**

JPEG, 0, kein Chroma-Subsampling, 11K

**Beispiel JPG**

JPEG, 0, Chroma geviertelt, 9K

**Video 7.5: JPEG – Probleme**

☞ JPEG Problems – Computerphile (5:36)

**JPEG 2000**

- JPEG hat diverse bekannte Nachteile
  - Block-Artefakte bei hoher Kompression
  - Artefakte, wenn hohe Frequenzanteile bedeutungstragend sind
- Nachfolgestandard: JPEG 2000
  - Bessere Leistung bei hoher Kompression (typisch weniger als 0.25 Bit pro Pixel)
  - wahlweise verlustbehaftet und verlustfrei
  - Progressiver Bildaufbau
  - Kodierung von "Regions of Interest" mit höherer Genauigkeit
  - Alphakanal
- Technisch: Diskrete Wavelet-Transformation statt DCT

## 5.4 Verlustfrei

**Verlustfrei**

- JPEG erreicht die hohen Kompressionsraten dadurch, daß verlustbehaftet komprimiert wird
- Die direkte Speicherung der Samples ist sehr speicherplatzintensiv
- Ist eine verlustfreie Speicherung möglich, die effektiver in Hinblick auf den Speicherplatz ist?

- Einige der betrachteten Eigenschaften von Bildern sind unabhängig von der Wahrnehmung
  - Flächigkeit von Farben
  - Getrennte Codierung von Kanälen
- Einige dieser Eigenschaften sind bei synthetischen Bildern noch stärker ausgeprägt
  - Liniengraphiken mit einfarbigen Flächen
- Idee: Optimale Codierung jedes Pixels je nach Umfeld

**PNG: Grundsätze**

- Zweistufiges Verfahren
  - Filter/Prediction
  - Kompression
- In der Regel hängt Farb-/Helligkeitswert eines Pixels von den Pixeln in der Nachbarschaft ab
- Differenz-/Deltakodierung: $[1, 2, 3, 4, 5, 6] \rightarrow [1, 1, 1, 1, 1, 1]$
- Bei PNG wird jedes Pixel in Relation zu den Pixeln links, oben und links oben bestimmt

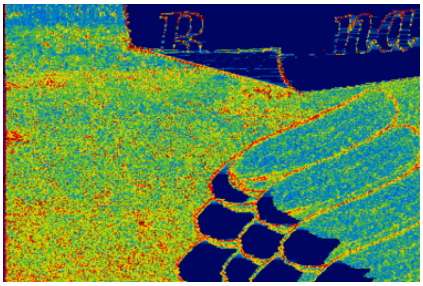
| | | | | |
|---|---|---|---|---|
| | | | | |
| | $C$ | $B$ | $D$ | |
| | $A$ | $X$ | | |
| | | | | |

| | | | | |
|---|---|---|---|---|
| 52 | 55 | 61 | 66 | 70 |
| 63 | 59 | 55 | 90 | 109 |

$\rightarrow$

| | | | | |
|---|---|---|---|---|
| 52 | 55 | 61 | 66 | 70 |
| 63 | $-4$ | $-4$ | 35 | 19 |

**Aufwand pro Bit**



dunkel/blau: geringe Bitkosten, grün: mittlere Kosten, rot: hohe Bitkosten

☞ en.wikipedia.org/wiki/Portable_Network_Graphics

**PNG Filter**

- Fünf unterschiedliche Filter zur Auswahl
  - None
    - Keine Vorfilterung
    - Es wird auf den originalen Pixeldaten gearbeitet
  - Sub
    - Differenzen zu dem jeweils links benachbarten Pixel
  - Up
    - Differenzen zu dem jeweils darüber liegenden Pixel
  - Average
    - Mittelwert aus darüber liegenden und dem links benachbarten Pixel
  - Paeth

    - Es wird aus dem links benachbarten, dem darüber liegenden und dem schräg links oben benachbarten Pixel ein sogenannter Paeth-Predictor-Wert berechnet, welcher das Pixel für die Differenzbildung bestimmt
    - $A, B$ oder $C$, was am nächsten $p = A + B - C$ ist
    - tie-breaker A - B - C

- Pro Zeile & Kanal kann der beste ausgewählt werden

**PNG: Filterbeispiele I**

| | | | | |
|---|---|---|---|---|
| 52 | 55 | 61 | 66 | 70 |
| 63 | 59 | 55 | 90 | 109 |

$\rightarrow$

| | | | | |
|---|---|---|---|---|
| 52 | 55 | 61 | 66 | 70 |
| 63 | 59 | 55 | 90 | 109 |

None

| | | | | |
|---|---|---|---|---|
| 52 | 55 | 61 | 66 | 70 |
| 63 | 59 | 55 | 90 | 109 |

$\rightarrow$

| | | | | |
|---|---|---|---|---|
| 52 | 55 | 61 | 66 | 70 |
| 63 | $-4$ | $-4$ | 35 | 19 |

Sub $(X - A)$

| | | | | |
|---|---|---|---|---|
| 52 | 55 | 61 | 66 | 70 |
| 63 | 59 | 55 | 90 | 109 |

$\rightarrow$

| | | | | |
|---|---|---|---|---|
| 52 | 55 | 61 | 66 | 70 |
| 63 | 4 | $-6$ | 24 | 39 |

Up $(X - B)$

**PNG: Filterbeispiele**

| | | | | |
|---|---|---|---|---|
| 52 | 55 | 61 | 66 | 70 |
| 63 | 59 | 55 | 90 | 109 |

$\rightarrow$

| | | | | |
|---|---|---|---|---|
| 52 | 55 | 61 | 66 | 70 |
| 63 | 0 | $-5$ | 30 | 29 |

Average $((A + B)/2)$

| | | | | |
|---|---|---|---|---|
| 52 | 55 | 61 | 66 | 70 |
| 63 | 59 | 55 | 90 | 109 |

$\rightarrow$

| | | | | |
|---|---|---|---|---|
| 52 | 55 | 61 | 66 | 70 |
| 63 | $-4$ | $-6$ | 29 | 19 |

Paeth

**PNG: Kompression & Eigenschaften**

- Nach der Filterung folgt die Kompression
- Theoretisch verschiedene Verfahren möglich, bisher spezifiziert nur DEFLATE
  - Substitution + Entropie
- Indexed/Paletten: 1 Kanal, mit 1, 2, 4 oder 8 bit pro Pixel pro Kanal (bpc)
- Grauwerte: 1 Kanal, 1, 2, 4, 8 oder 16 bpc
- Grauwerte & Transparenz/Alphakanal: = 2 Kanäle, 8 oder 16 bpc
- Truecolor (RGB): 3 Kanäle, 8 oder 16 bpc
- Truecolor & Transparenz (RGBA): 4 Kanäle, 8 oder 16 bpc

**Kompressionsverfahren**

- LZ77
  - Suche nach identischen Teilstrings
  - Sliding Window mit fester Größe
  - "Offset" (Distanz) und "Länge" des Teilstrings speichern
  - Effiziente Lauflängenkodierung: 111111 wird zu 1, (Offset: -1, Länge:5), 121212 zu 12, (-2, 4)
- LZ78
  - Statt "Offset" und "Länge" zu speichern wird Wörterbuch angelegt und darauf verwiesen
- LZW
  - Grundidee: Initiales statt leeres Wörterbuch
- DEFLATE
  - Erst analog LZ77/LZ78, danach Huffman
  - Neben PNG in gzip, zip etc.

# 6 Bearbeitung

**Bearbeitung von Bildern**

- Auswahl möglicher Operationen
  - Skalierung
  - Aufbereitung für Druck, Web, etc.
  - Drehung, Entzerrung, Filterung
  - Schnitt, Montage, Manipulation
- Neben der Bildbearbeitung auch die Bildverarbeitung
- Tendentielle Unterteilung
  - Automatische Verarbeitungsketten, z.B. in der Robotik: Bildverarbeitung
  - Interaktive Veränderung von Bildern: Bildbearbeitung

**Punkt-Operationen**

- Bei Punkt-Operationen werden die Werte einzelner Pixel verändert, ohne Nachbarpixel zu betrachten
- Beispiel: Änderung der Farbtiefe
- Grauwertbild mit 256 Werten pro Pixel in ein Schwarzweißbild

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

- $w$ beliebiger Grenzwert (Schwellwert)

### Histogramm

- Häufig spielen Histogramme eine Rolle
- Geben an, welche Werte wie oft in den Pixeln vorkommen
- Bei einem Grauwertbild mit 256 möglichen Werten haben wir einen Vektor $h(w), w = 0, ..., 255$ wobei $h(w)$ die Anzahl der Pixel mit dem entsprechenden Wert angibt
- Typischerweise soll ein Bild den gesamten Bereich eines Histogramms nutzen, sonst wirkt es flach und besitzt wenig Kontrast
- Nutzt ein Grauwertbild mit 8 Bit pro Pixel nur die Grauwerte 50 bis 150, kann man die Werte so spreizen daß der niedrigste bei 0 und der höchste bei 255 liegt:

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

### Beispiel

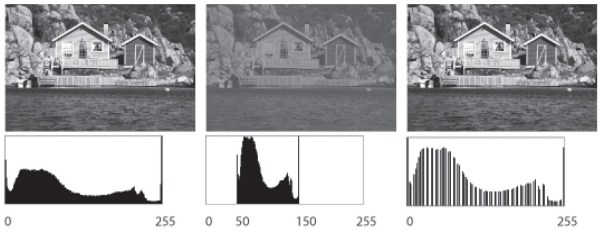



**Abbildung 3.12:** Grauwertbilder mit Histogrammen. Linkes Bild: Alle Grauwerte kommen vor. Mittleres Bild: Nur Grauwerte zwischen 50 und 150 kommen vor. Rechtes Bild: Spreizung der Grauwerte auf den ganzen Bereich, wobei das Ausgangsbild das mittlere Bild ist.

- Lücken im gespreizten Histogramm entstehen dadurch, daß Zwischenwerte nicht vorhanden waren
- Auch bei Farbbildern prinzipiell möglich
  - Einfach pro Kanal im RGB?
- auch nichtlineare Varianten denkbar

### Zonensystem

| Zone | Description |
|---|---|
| 0 | Pure black |
| I | Near black, with slight tonality but no texture |
| II | Textured black; the darkest part of the image in which slight detail is recorded |
| III | Average dark materials and low values showing adequate texture |
| IV | Average dark foliage, dark stone, or landscape shadows |
| V | Middle gray: clear north sky; dark skin, average weathered wood |
| VI | Average Caucasian skin; light stone; shadows on snow in sunlit landscapes |
| VII | Very light skin; shadows in snow with acute side lighting |
| VIII | Lightest tone with texture: textured snow |
| IX | Slight tone without texture; glaring snow |
| X | Pure white: light sources and specular reflections |

☞ Zonensystem nach Ansel Adams und Fred Archer (Wikipedia)

**Filter**

- Im Gegensatz zu Punkt-Operationen werden bei Filtern neue Pixel berechnet, indem Informationen mehrerer Pixel des Originals betrachtet werden
- Meist in der Nachbarschaft
- Einfach: Glättung, Weichzeichner bei dem jedes Pixel aus dem Mittelwert der Nachbarschaft besteht

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

**Filter (contd.)**

- Beispiel hier mit neun Pixeln
- Dieser Filter ist linear darstellbar als Faltung der Umgebung mit einer Matrix

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

- Sollen Kontraste nicht weichgezeichnet, sondern verstärkt werden, würde ein einfacher Boost-Filter dabei helfen:

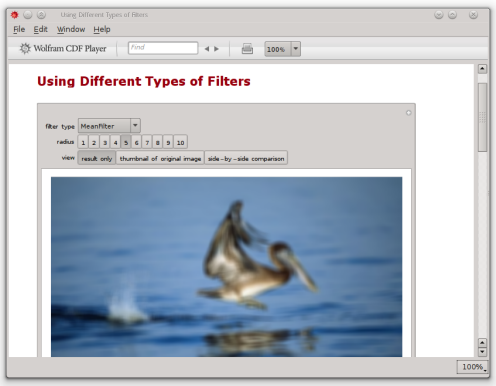
$$f'(x,y) = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{pmatrix} \begin{pmatrix} f(x-1,y-1) & f(x,y-1) & f(x+1,y-1) \\ f(x-1,y) & f(x,y) & f(x+1,y) \\ f(x-1,y+1) & f(x,y+1) & f(x+1,y+1) \end{pmatrix}$$

**Beispiele**

**Abbildung 3.13:** Linkes Bild: Original, Mitte: einfacher Weichzeichner, rechts: einfacher Scharfzeichner (Boost-Filter)

- Neben den gezeigten einfachen Filtern sind auch andere denkbar
  - Wahl der Funktion, mit der die Nachbarpixel einbezogen werden

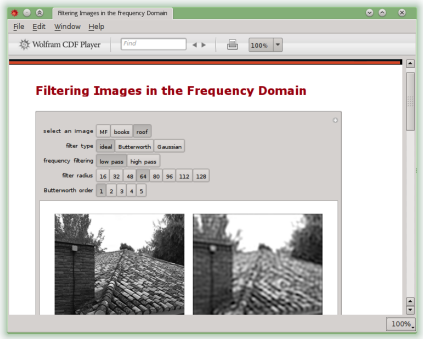
**Filter: Beispiel**




☞ demonstrations.wolfram.com/UsingDifferentTypesOfFilters/

### Filterung im Frequenzraum

- Filter können auch im Frequenzraum angewandt werden
  - Filtern hoher Frequenzen ergibt weichgezeichnete Bilder




☞ demonstrations.wolfram.com/FilteringImagesInTheFrequencyDomain/

### Filterung im Frequenzraum: Vor-und Nachteile

- Vorteile
  - Multiplikation statt Matritzenoperation (Faltung)
  - Weichzeichnung unabhängig von Größe des Filters
- Nachteile
  - Rechenaufwand der Transformation
  - Nicht alle Filter durchführbar (Beispiel: Medianfilter)
  - Einige Operationen aufwendiger (Beispiel: Punktoperatoren zur Helligkeitsänderung)
  - Ergebnis erst nach inverser Transformation sichtbar

### Geometrische Operationen

- Auswahl möglicher Operationen
  - Verschiebung
  - Streckung (Verkleinerung, Vergrößerung)
  - Drehung
  - Scherung
- Problem: Sowohl Ausgangsbild wie Zielbild mit festen Rastern

### Geometrische Operationen (contd)

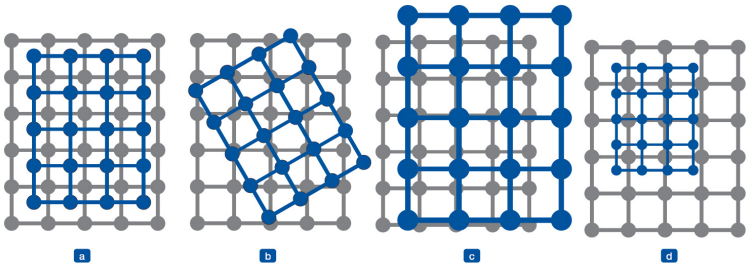



**Abbildung 3.14:** Probleme bei geometrischen Operationen. Blaues Raster: transformiertes Pixelraster des Originalbildes; graues Raster: Pixelraster des Zielbildes bei (a) Verschiebung, (b) Drehung, (c) Vergrößerung, (d) Verkleinerung

- Wenn man bei der Verkleinerung einem Zielpixel mehrere Ausgangsbild-Werte zuordnen muß kann dies durch Mittelwertbildung erreicht werden
- Problem: Unterabtastung, möglicher Moiré-Effekt

**Beispiel: Verkleinerung**

Abbildung 3.15

**Beispiel: Vergrößerung**

- Im umgekehrten Fall, ein Pixel auf mehrere, sollte der Wert nicht einfach kopiert werden
- Resultat wäre "verpixelt"
- Einfache Gegenmaßnahme: Weichzeichner als Anti-Aliasing

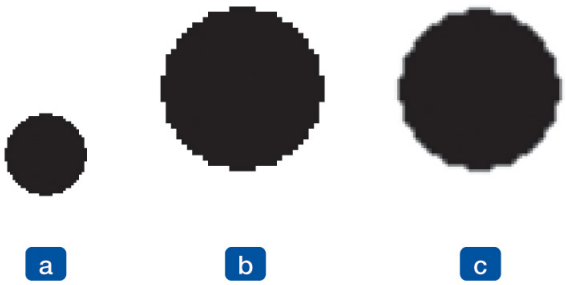
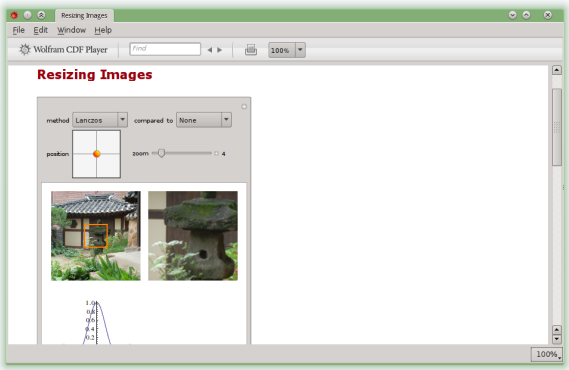



**Abbildung 3.16:** Alias-Effekt beim Vergrößern. (a) Originalbild, (b) Vergrößerung mit Zoom 200% ohne Antialiasing, (c) mit Antialiasing

**Größenänderung allgemein**




☞ demonstrations.wolfram.com/ResizingImages/

**Formate**

- Welches Format für welchen Zweck?
- Aufnahme

  - Maximale Auflösung
  - Maximale Farbtiefe
  - RAW; keine Kompression
- Archiv – RAW
- Web und Anwendungen
  - JPEG: digitale Fotos, für das Web stark komprimiert
  - GIF: deprecated, außer für Loops
  - PNG: Moderner und flexibler für Graphiken, mit echtem Alphakanal
- Bilder und Strichzeichnungen
  - Bilder: JPEG
  - Strichzeichnungen: PNG

**Video 7.6: Enhance**

☞ Enhance (1:40)

**Bildnachweis**

Alle Abbildungen, wenn nicht anders angegeben, aus Malaka et al. [2009].

**References**

# Literatur


Rainer Malaka, Andreas Butz, and Heinrich Hussmann. *Medieninformatik – Eine Einführung*. Pearson Studium, Munich, 2009.

Jacques Ninio and Kent A Stevens. Variations on the hermann grid: An extinction illusion. *Perception*, 29(10):1209–1217, 2000. doi: 10.1068/p2985. URL https://doi.org/10.1068/p2985. PMID: 11220212.

Dale Purves, George J Augustine, David Fitzpatrick, Lawrence C Katz, Anthony-Samuel LaMantia, James O McNamara, and S Mark Williams., editors. *Neuroscience, 2nd Edition*. Sinauer Associates, Sunderland (MA), 2001.