

Grundlagen Digitaler Medien

Farbe und Farträume

Dozent: Prof. Dr.-Ing. Sebastian Knorr

Email: knorr@htw-berlin.de

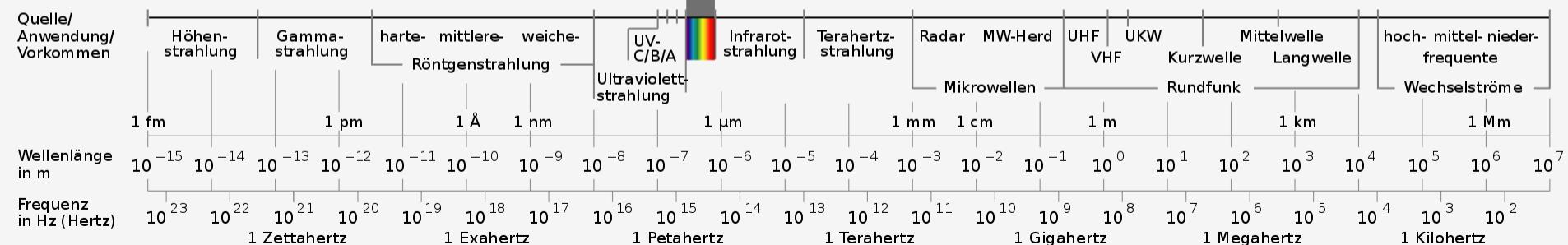
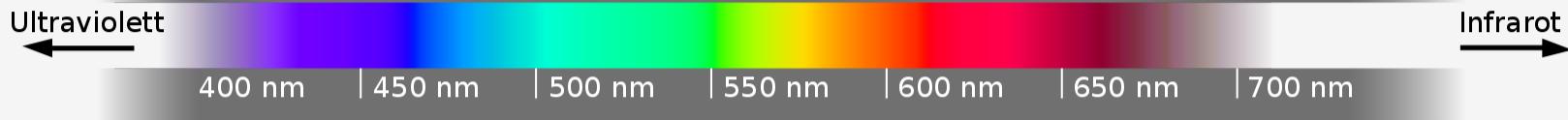


Lernziele

- Verständnis für die Darstellung von Farbe
- Verständnis für unterschiedliche Farbräume
- Verständnis für Farbquantisierung

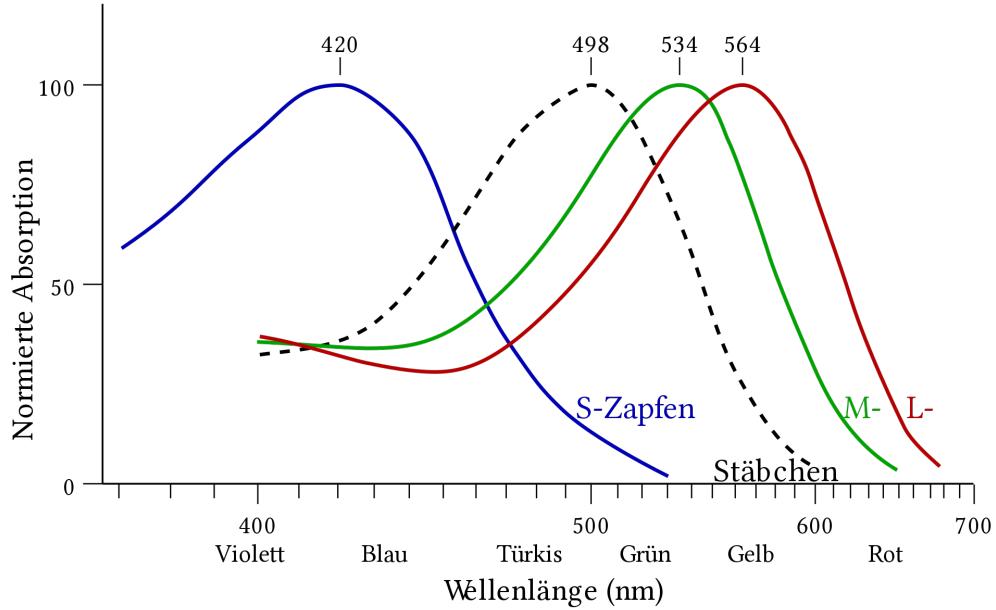
Farbwahrnehmung - Farbspektrum

Das für den Menschen sichtbare Spektrum (Licht)



Elektromagnetisches Spektrum

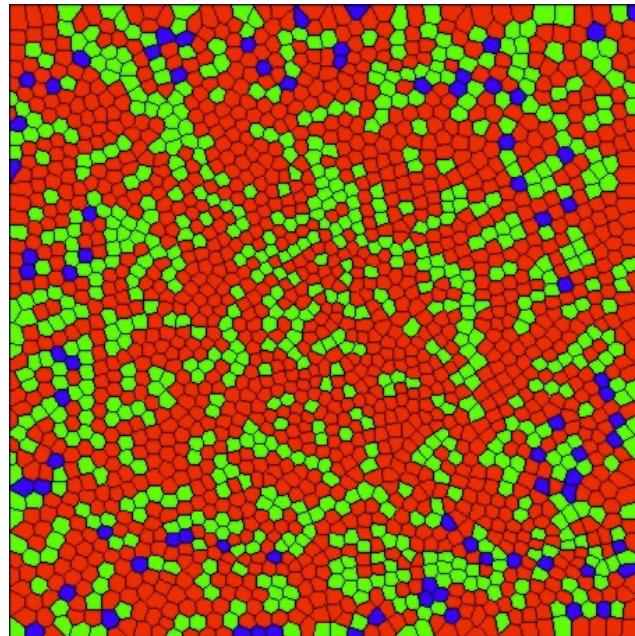
Farbwahrnehmung



S-Zapfen („Blauzapfen“): blau-violett

M-Zapfen („Grünzapfen“): smaragdgrün

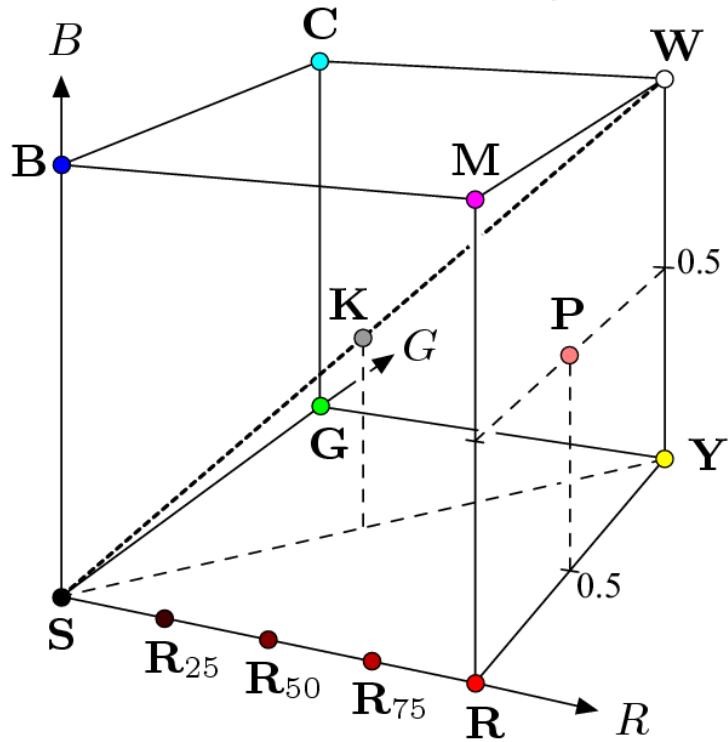
L-Zapfen („Rotzapfen“): grün-gelb



Simulation der Verteilung der Zäpfchen auf der Fovea

RGB – Farbraum / Farbwürfel

$$\mathbf{C}_i = (R_i, G_i, B_i) \quad \text{RGB-Pixel}$$



Pkt.	Farbe	RGB-Werte		
		R	G	B
S	Schwarz	0.00	0.00	0.00
R	Rot	1.00	0.00	0.00
Y	Gelb	1.00	1.00	0.00
G	Grün	0.00	1.00	0.00
C	Cyan	0.00	1.00	1.00
B	Blau	0.00	0.00	1.00
M	Magenta	1.00	0.00	1.00
W	Weiß	1.00	1.00	1.00
K	50% Grau	0.50	0.50	0.50
R₇₅	75% Rot	0.75	0.00	0.00
R₅₀	50% Rot	0.50	0.00	0.00
R₂₅	25% Rot	0.25	0.00	0.00
P	Pink	1.00	0.50	0.50

RGB-Komponenten



R



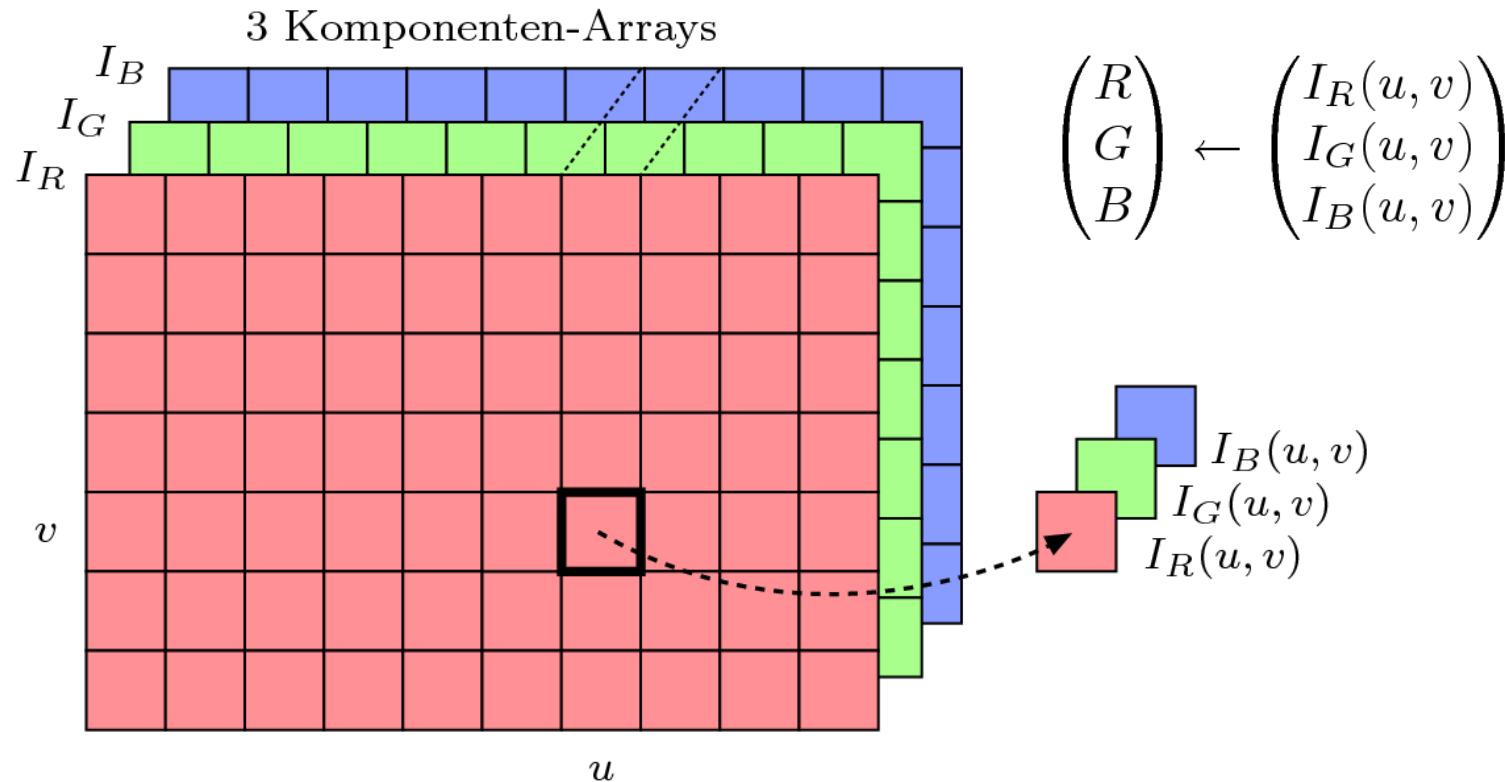
G



B

Bildaufbau: Komponentenanordnung (RGB)

“Planes”: Die 3 Farbkomponenten sind in getrennten 2D-Arrays gleicher Größe angelegt

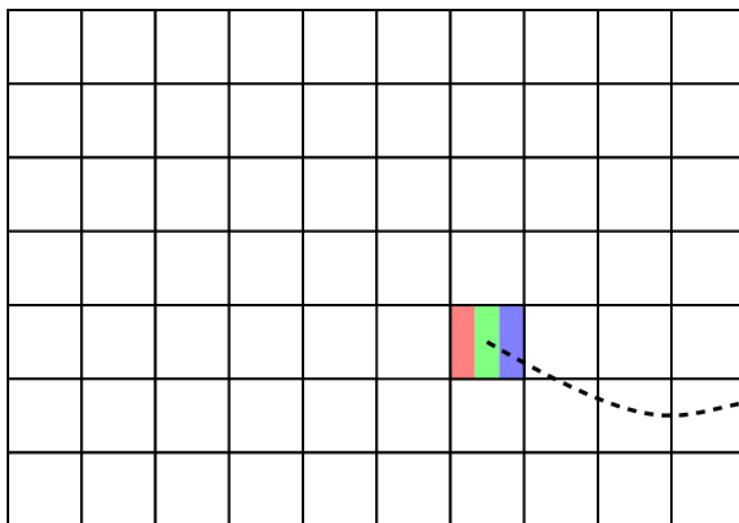


Bildaufbau: Gepackte Anordnung (RGB)

“Interleaved”:

Die 3 Farbkomponenten sind in einem 2D-Array-Element zusammengefügt.

RGB-Pixel-Array



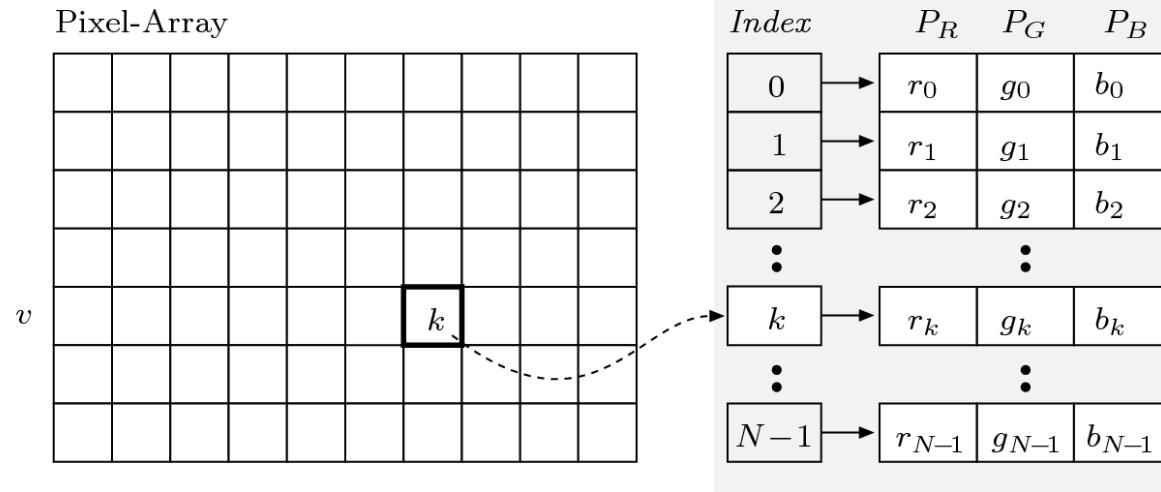
$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} \leftarrow \begin{pmatrix} \text{Red}(I(u, v)) \\ \text{Green}(I(u, v)) \\ \text{Blue}(I(u, v)) \end{pmatrix}$$

A horizontal vector consisting of three colored squares: red, green, and blue. To the left of the vector is a small black arrow pointing to the right, indicating the direction of the sequence. Below the vector, the label $I(u, v)$ is centered, identifying the entire row as a single 2D array element.

$$\begin{matrix} R & G & B \end{matrix}$$
$$I(u, v)$$

Sonderform: Index- oder Palettenbilder

Das Bildarray selbst enthält keine Farbwerte, sondern jeweils einen Index k in eine gesonderte Farbtabelle (bspw. bei GIF, PNG, TIFF)

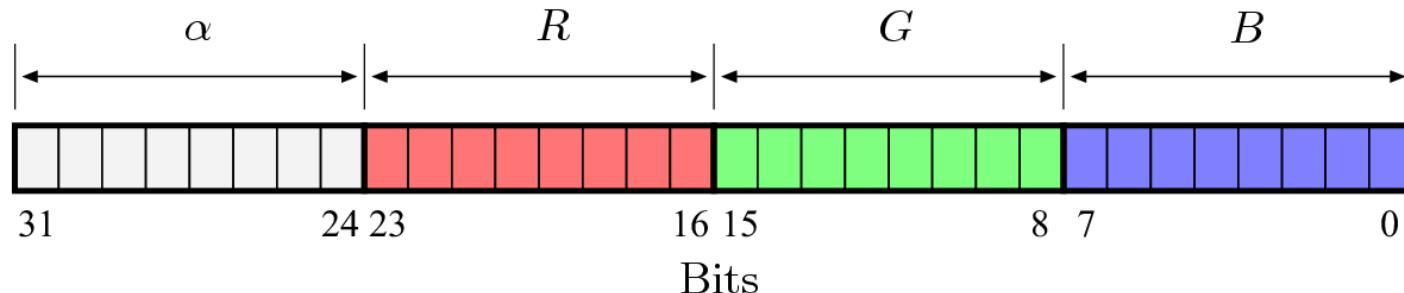


$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} \leftarrow \begin{pmatrix} P_R[k] \\ P_G[k] \\ P_B[k] \end{pmatrix}, \quad \text{wobei } k = I(u, v)$$

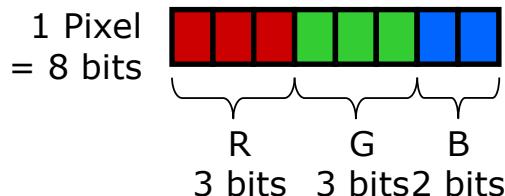
Vollfarbenbilder

Jedes Pixel kann eine beliebige Kombination von Primärfarben einnehmen (abhängig vom Farbraum: RGB, CMYK, etc.)

Typisch: 24-Bit RGB-Bilder (hier: 32-Bit RGBA-Bilder)



Extrembeispiel: 3-3-2-Bilder (8-Bit)



$3 \times 8 = 24$ Bit
$3 \times 10 = 30$ Bit
$3 \times 12 = 36$ Bit
$3 \times 16 = 48$ Bit

Umwandlung in Grauwerte

Einfacher Mittelwert aus allen 3 Farbkomponenten:

$$Y = \text{avg}(R, G, B) = \frac{R + G + B}{3}$$

unüblich, da nicht auf die menschliche Wahrnehmung angepasst (bspw. wird grün vom menschlichen Auge stärker wahrgenommen)

„Luminanz“ als gewichtete Summe der Farbkomponenten:

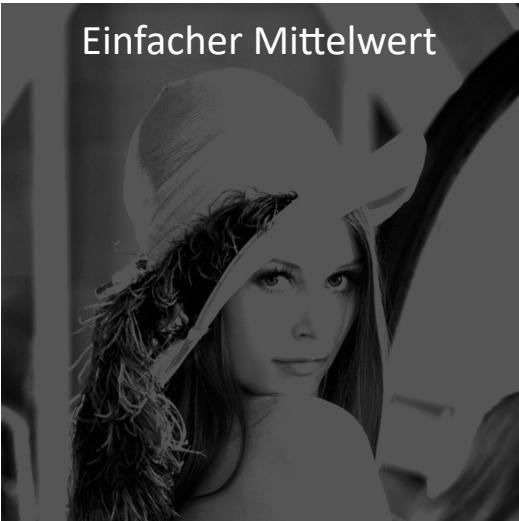
$$Y = \text{Lum}(R, G, B) = w_R \cdot R + w_G \cdot G + w_B \cdot B$$

Beispiele: Analog-TV: $w_R = 0.299$ $w_G = 0.587$ $w_B = 0.114$

Digital-TV: $w_R = 0.2125$ $w_G = 0.7154$ $w_B = 0.072$



Einfacher Mittelwert



Analog TV



Digital TV



Farbsättigung

Die *Sättigung* beschreibt „wie stark sich ein farbiger Reiz von einem achromatischen Reiz unabhängig von dessen Helligkeit unterscheidet“, also sein Abstand von der *Unbunt-Achse* (Schwarz-Weiß-Achse). Damit haben alle Farbtöne (Bunttöne) eine Sättigung bis zu 100 %, und Weiß, Grau und Schwarz eine Sättigung von 0.

Vom Menschen als grundlegend empfundene Eigenschaften einer Farbe:

- **Helligkeit**
- **Farbton**
- **Sättigung**



100%



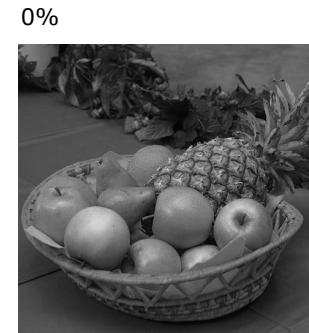
75%



50%



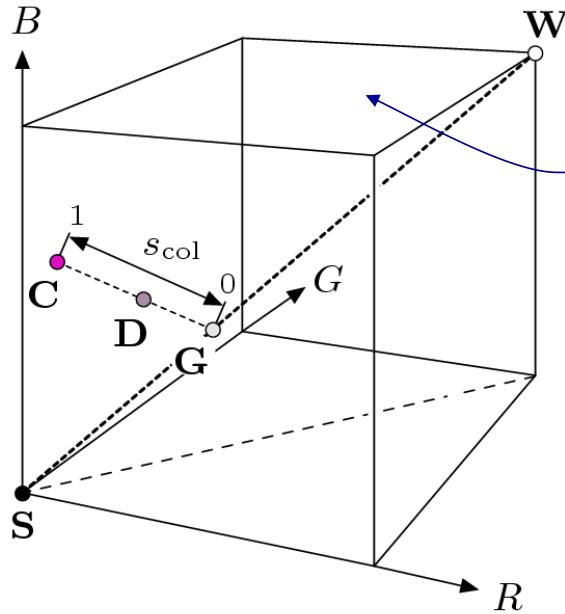
25%



0%

Desaturation

Graduelle Reduktion der Farbsättigung („desaturate“)



Graues Pixel:

$$R = G = B$$

$$\mathbf{G} = (Y, Y, Y)$$

Desaturiertes Pixel, $s_{\text{col}} \in [0,1]$
bestimmt verbleibende
Farbsättigung:

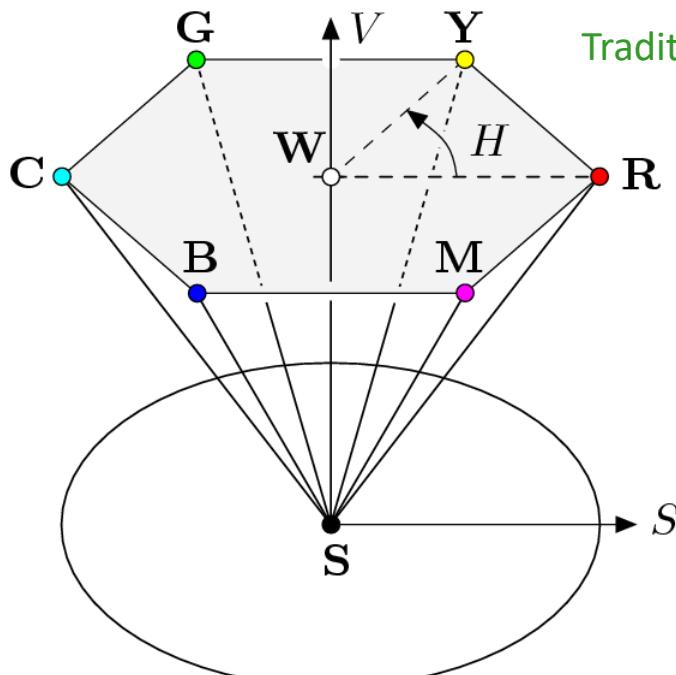
$$\mathbf{D} = \begin{pmatrix} R_D \\ G_D \\ B_D \end{pmatrix} = \begin{pmatrix} Y \\ Y \\ Y \end{pmatrix} + s_{\text{col}} \cdot \begin{pmatrix} R - Y \\ G - Y \\ B - Y \end{pmatrix}$$

RGB-Farbraum - Zusammenfassung

- Basiert auf Anzeigegeräten aus der Computertechnik – aus Programmiersicht besonders einfach.
- Verschiebungen im RGB-Farbraum werden sehr unterschiedlich wahrgenommen (RGB ist subjektiv "nichtlinear").
- Farbton, Farbsättigung und Helligkeit ändern sich bei jeder RGB-Koordinatenbewegung gleichzeitig.
- Eigenschaften von RGB sind nicht definiert, Erscheinung der Farben ist geräteabhängig.

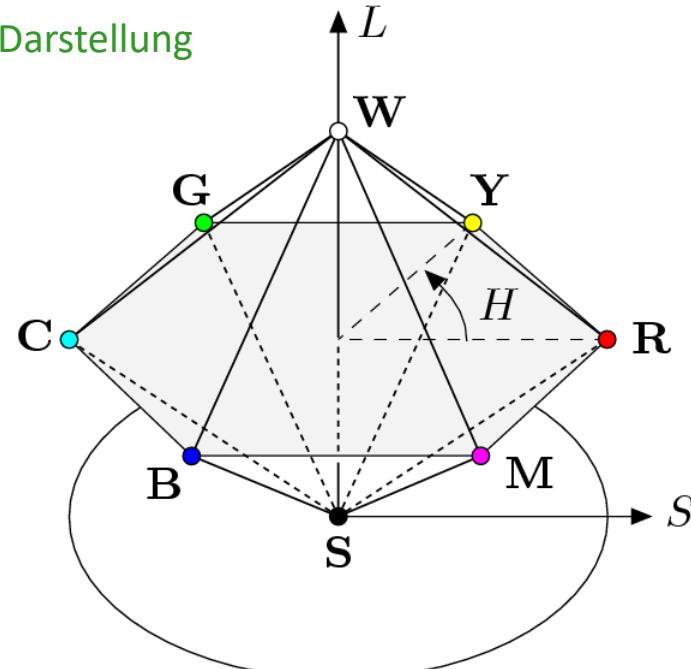
HSV- und HLS-Farträume

Hue (Farbton) • Saturation (Farbsättigung) • Value/Lightness (Helligkeit)



HSV - Farbraum

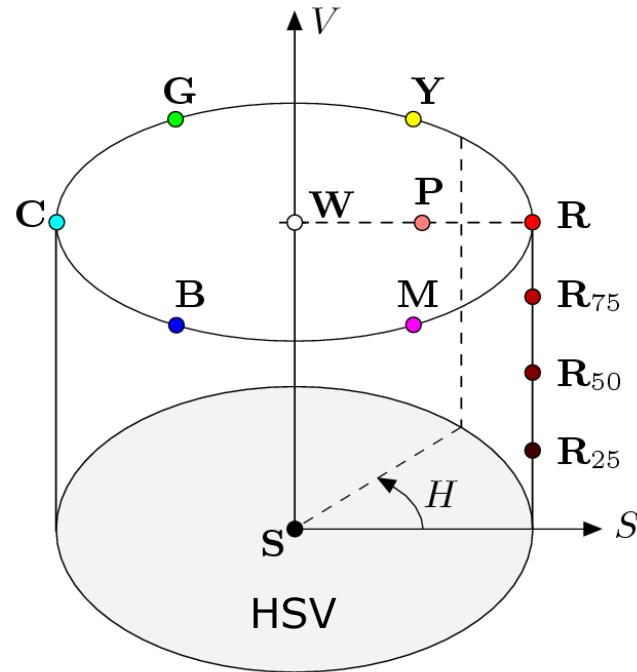
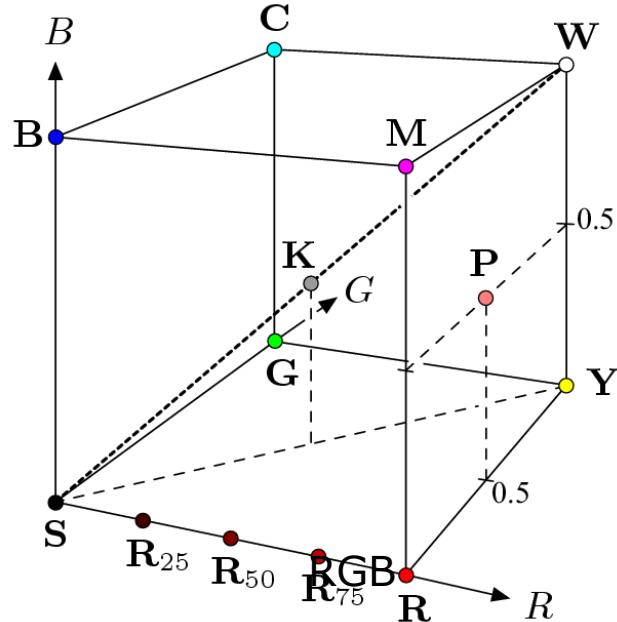
Traditionelle Darstellung



HLS - Farbraum

HSV-Farbraum (auch HSB, HIS)

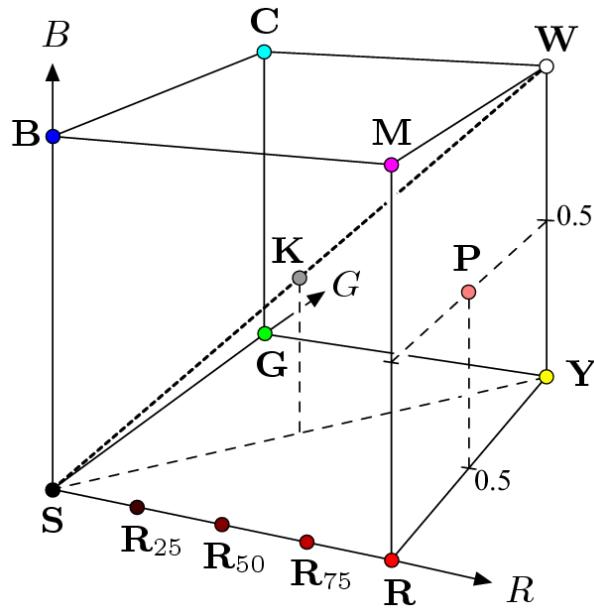
HSV: Hue/Saturation/Value, **HSB:** Hue/Saturation/Brightness, **HSV:** Hue/Saturation/Intensity



Realistische Darstellung

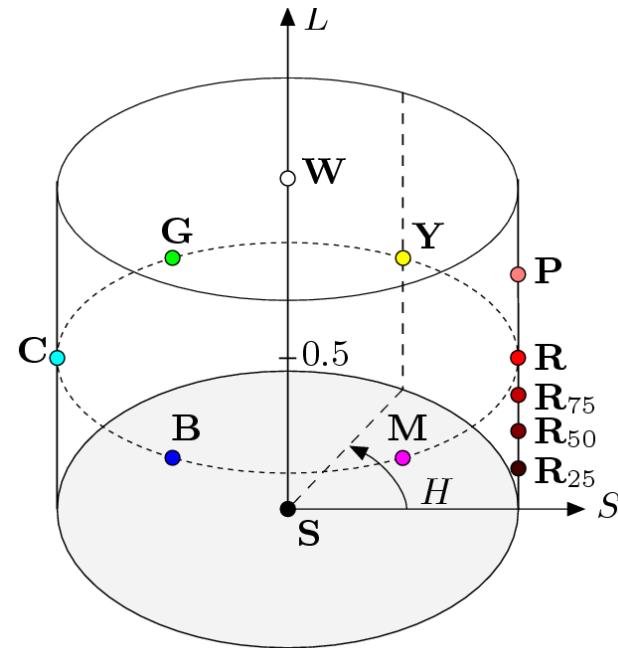
HLS-Farbraum (auch HSL)

HSV: Hue/Saturation/Value, HSL: Hue/Saturation/Lightness



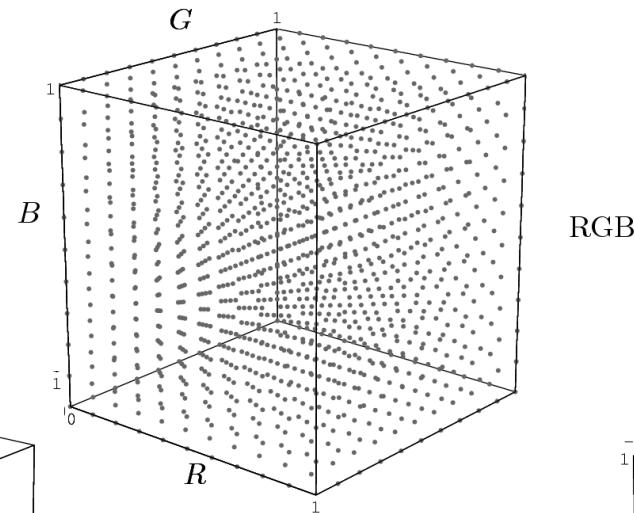
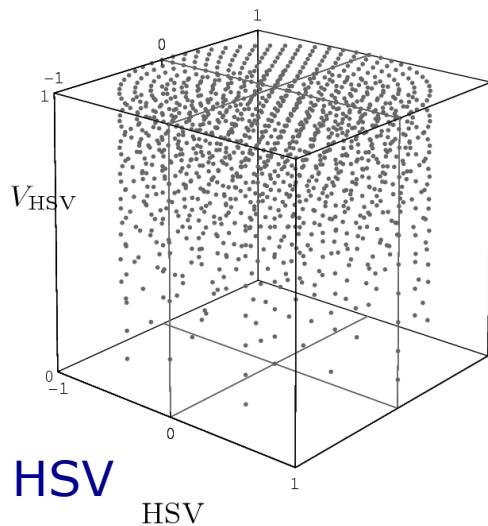
RGB

Realistische Darstellung

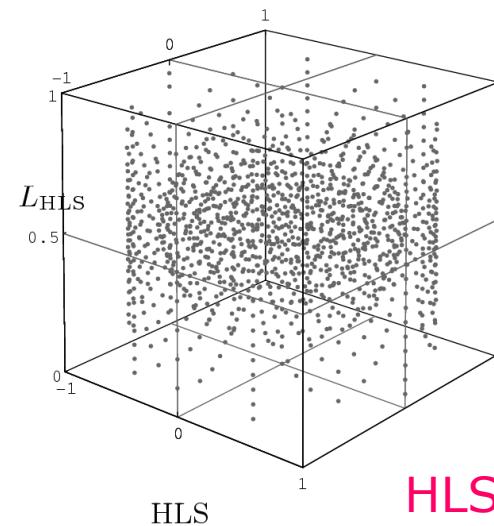


HLS

Farbverteilung in RGB / HSV / HLS



RGB



HLS

RGB → HSV Umrechnung (1)

Vorbedingung: $R, G, B \in [0, 1]$

$$MAX := \max(R, G, B), \ MIN := \min(R, G, B)$$

Saturation:

$$S_{HSV} := \begin{cases} 0, & \text{falls } MAX = 0 \Leftrightarrow R = G = B = 0 \\ \frac{MAX - MIN}{MAX}, & \text{sonst} \end{cases}$$

Value:

$$V := MAX$$

RGB → HSV Umrechnung (2)

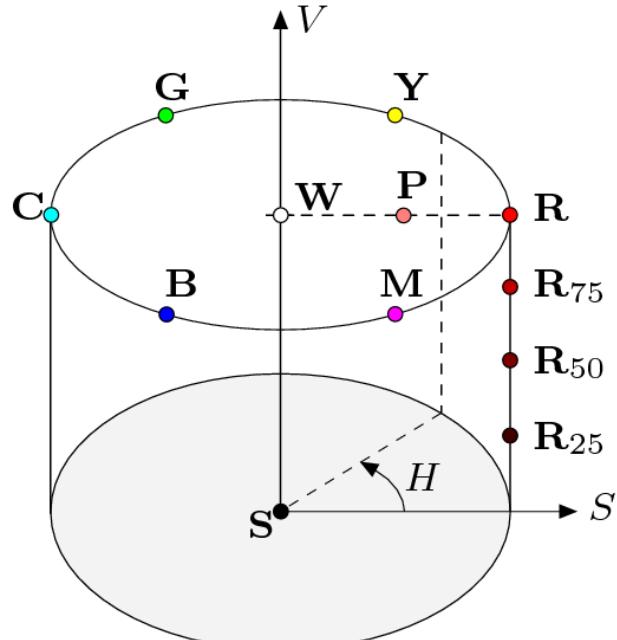
$$MAX := \max(R, G, B), \ MIN := \min(R, G, B)$$

Hue:

$$H := \begin{cases} 0, & \text{falls } MAX = MIN \Leftrightarrow R = G = B \\ 60^\circ \cdot \left(0 + \frac{G-B}{MAX-MIN}\right), & \text{falls } MAX = R \\ 60^\circ \cdot \left(2 + \frac{B-R}{MAX-MIN}\right), & \text{falls } MAX = G \\ 60^\circ \cdot \left(4 + \frac{R-G}{MAX-MIN}\right), & \text{falls } MAX = B \end{cases}$$

falls $H < 0^\circ$ dann $H := H + 360^\circ$

Konkrete HSV-Farbwerte



RGB-/HSV-Werte

Pkt.	Farbe	<i>R</i>	<i>G</i>	<i>B</i>	<i>H</i>	<i>S</i>	<i>V</i>
S	Schwarz	0.00	0.00	0.00	—	0.00	0.00
R	Rot	1.00	0.00	0.00	0	1.00	1.00
Y	Gelb	1.00	1.00	0.00	1/6	1.00	1.00
G	Grün	0.00	1.00	0.00	2/6	1.00	1.00
C	Cyan	0.00	1.00	1.00	3/6	1.00	1.00
B	Blau	0.00	0.00	1.00	4/6	1.00	1.00
M	Magenta	1.00	0.00	1.00	5/6	1.00	1.00
W	Weiß	1.00	1.00	1.00	—	0.00	1.00
R₇₅	75% Rot	0.75	0.00	0.00	0	1.00	0.75
R₅₀	50% Rot	0.50	0.00	0.00	0	1.00	0.50
R₂₅	25% Rot	0.25	0.00	0.00	0	1.00	0.25
P	Pink	1.00	0.50	0.50	0	0.5	1.00

RGB → HLS Umrechnung

Vorbedingung: $R, G, B \in [0, 1]$

$$MAX := \max(R, G, B), \ MIN := \min(R, G, B)$$

Hue:

$$H_{\text{HLS}} = H_{\text{HSV}}$$

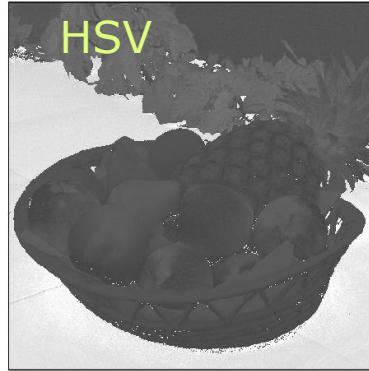
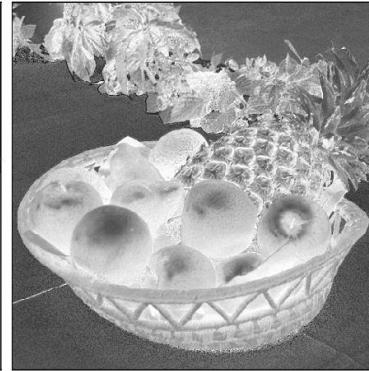
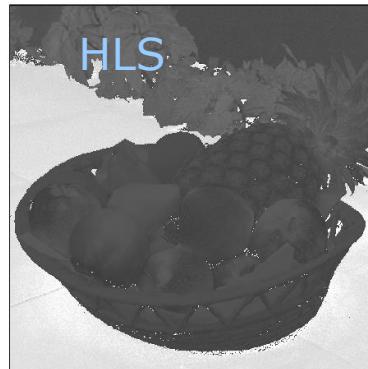
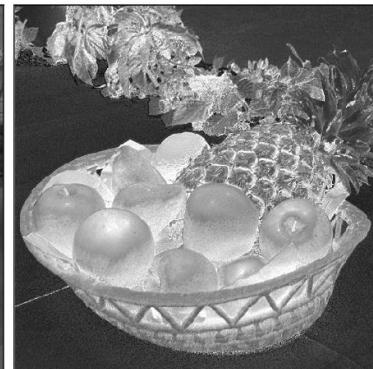
Saturation:

$$S_{\text{HSL}} := \begin{cases} 0, & \text{falls } MAX = 0 \Leftrightarrow R = G = B = 0 \\ 0, & \text{falls } MIN = 1 \Leftrightarrow R = G = B = 1 \\ \frac{MAX - MIN}{1 - |MAX + MIN - 1|}, & \text{sonst} \end{cases}$$

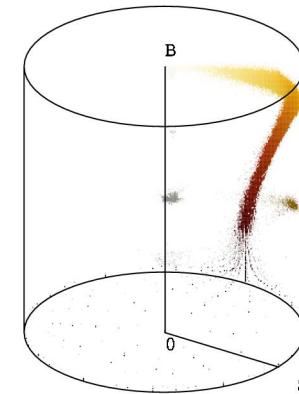
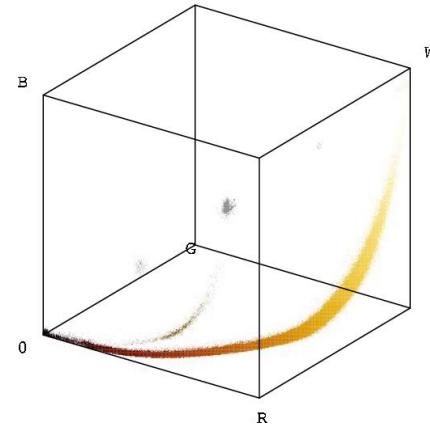
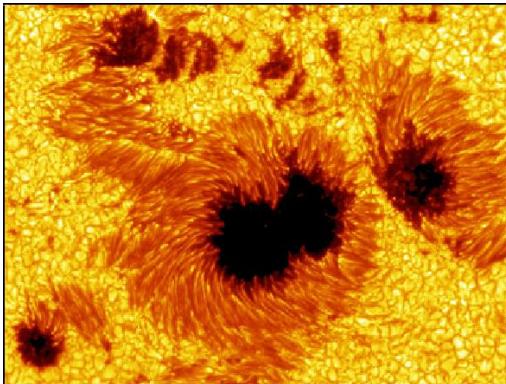
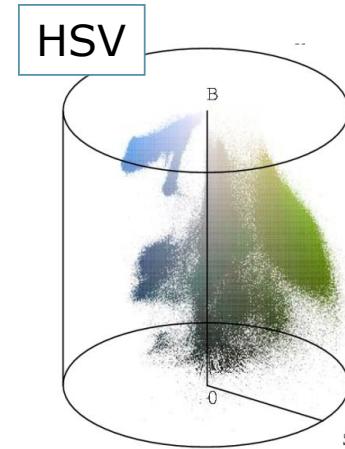
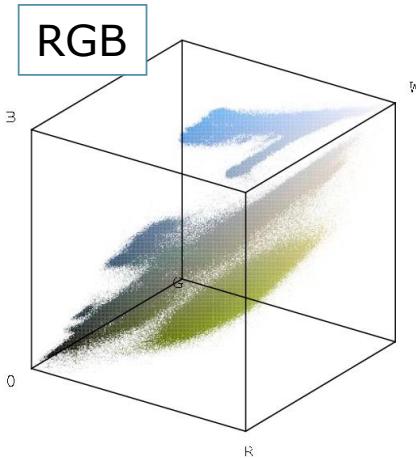
Lightness:

$$L := \frac{MAX + MIN}{2}$$

HSV, HLS Beispiele

 h_{HSV}  s_{HSV}  v_{HSV}  H_{HLS}  L_{HLS}  S_{HLS}

Farbverteilung in RGB / HSV



TV-Komponentenfarträume

- Farträume zur **standardisierten** Aufnahme, Speicherung, Übertragung, Wiedergabe von Farb-TV-Bildern
- Trennung in
 - Luminanzkomponente (Y) und
 - 2 Farbkomponenten (U, V)
- YUV (PAL/NTSC analog-TV), YCbCr (digital-TV)

YUV-Komponentenfarbraum

Umrechnung aus RGB:

$$Y = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B$$

$$U = 0.492 \cdot (B - Y)$$

$$V = 0.877 \cdot (R - Y)$$

In Matrixform (lineare Abbildung):

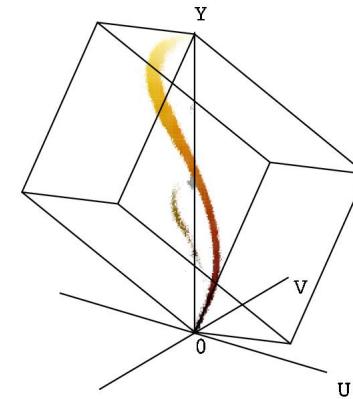
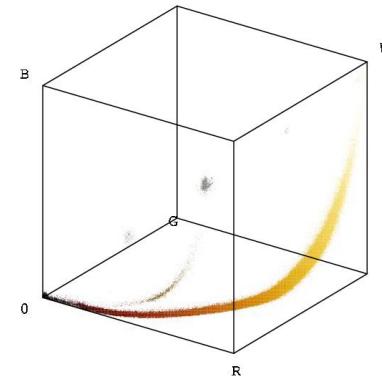
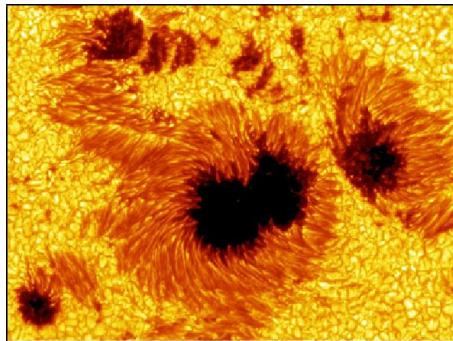
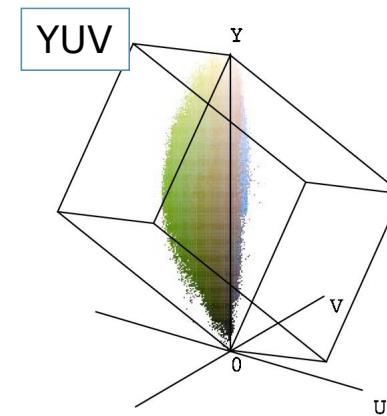
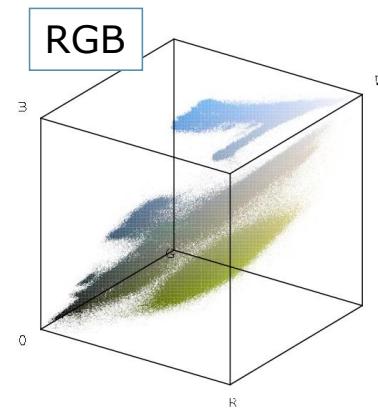
RGB → YUV

$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{pmatrix} \cdot \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

YUV → RGB

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 1.000 & 0.000 & 1.140 \\ 1.000 & -0.395 & -0.581 \\ 1.000 & 2.032 & 0.000 \end{pmatrix} \cdot \begin{pmatrix} Y \\ U \\ V \end{pmatrix}$$

Farbverteilung in RGB / YUV



YC_bC_r-Komponentenfarbraum

- Variante (Weiterentwicklung) von YUV
- Internationaler Standard für digital-TV
- Auch bei JPEG (JFIF, EXIF) üblich
- Koeffizienten für Chroma- und Luminanzwerte sind gekoppelt

RGB → YC_bC_r

$$\begin{pmatrix} Y \\ C_b \\ C_r \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{pmatrix} \cdot \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

YC_bC_r → RGB

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 1.000 & 0.000 & 1.403 \\ 1.000 & -0.344 & -0.714 \\ 1.000 & 1.773 & 0.000 \end{pmatrix} \cdot \begin{pmatrix} Y \\ C_b \\ C_r \end{pmatrix}$$

Nach ITU-BT.601 (4:3 und 16:9 Digital-TV): $w_R=0.299$, $w_B=0.114$

Nach ITU-BT.709 (Digital-HDTV): $w_R=0.2125$, $w_B=0.0721$

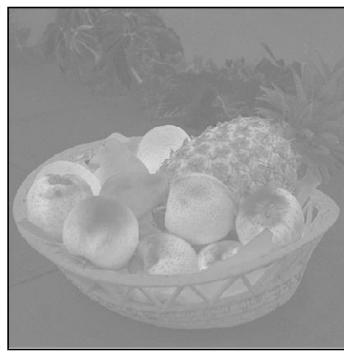
YUV und YC_bC_r Komponenten



Y



U



V

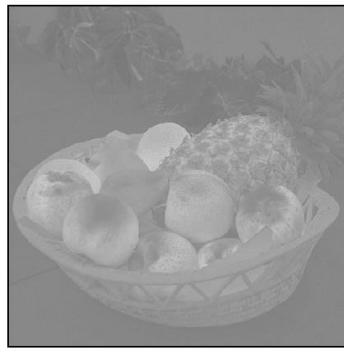
YUV



Y



C_b



C_r

YC_bC_r

Umwandlung True-Color/Indexed Color: Farbquantisierung

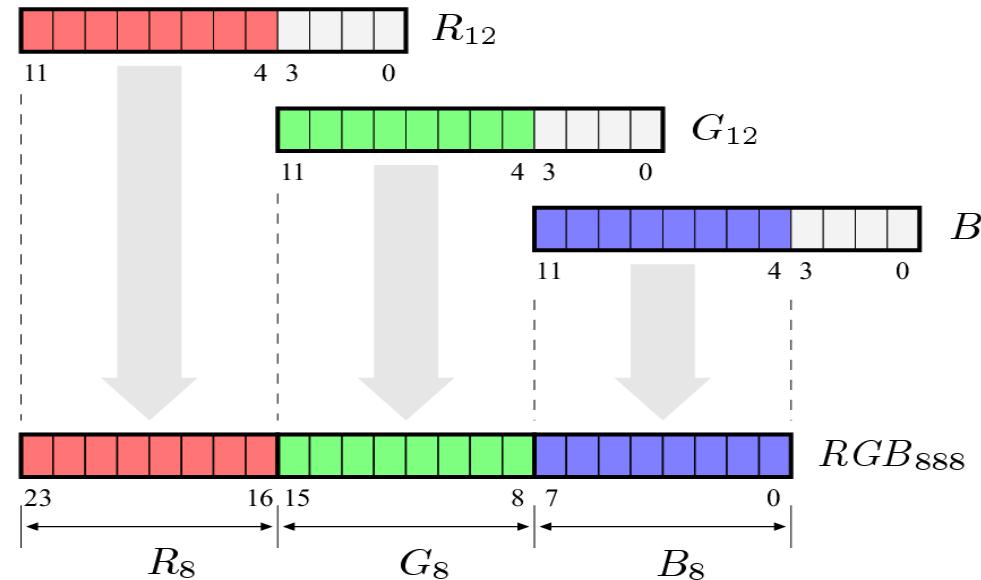
- Indexed-Color → True-Color
 - Problemlos: ersetze alle Pixel mit Wert (Index) k durch $(R_k, G_k, B_k) = \text{Colortable}[k]$.
- TrueColor → Indexed-Color
 - 24 bits → 8 bits (typ.)
 $2^{24} = 16.777.216$ Farben → 256 Farben
 - FARBQUANTISIERUNG – welche Farben sind wichtig/repräsentativ?

Skalare Farbquantisierung

- Wertebereich der einzelnen Komponenten wird gleichmäßig reduziert

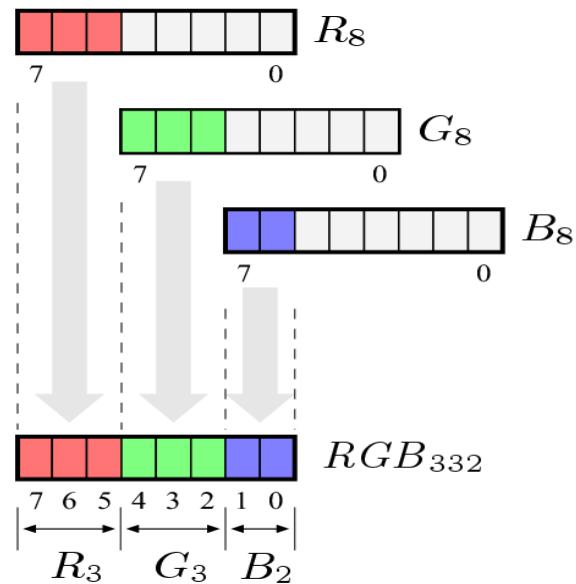
Beispiel:

36 Bit \rightarrow 24 Bit



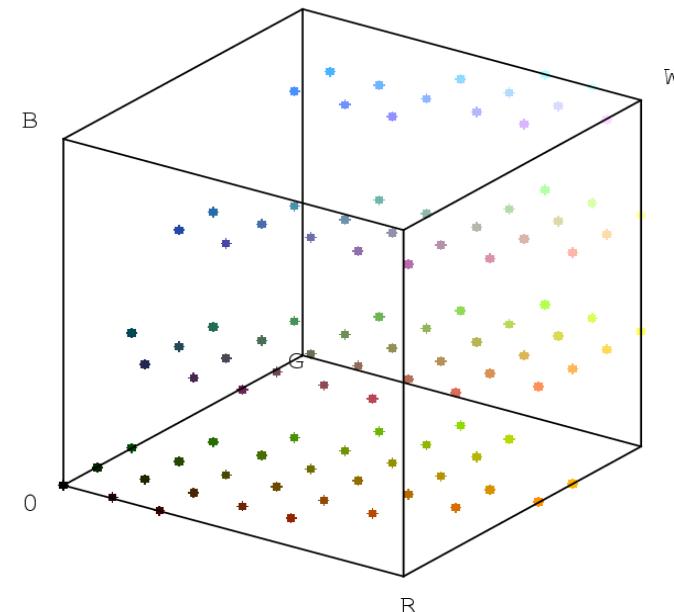
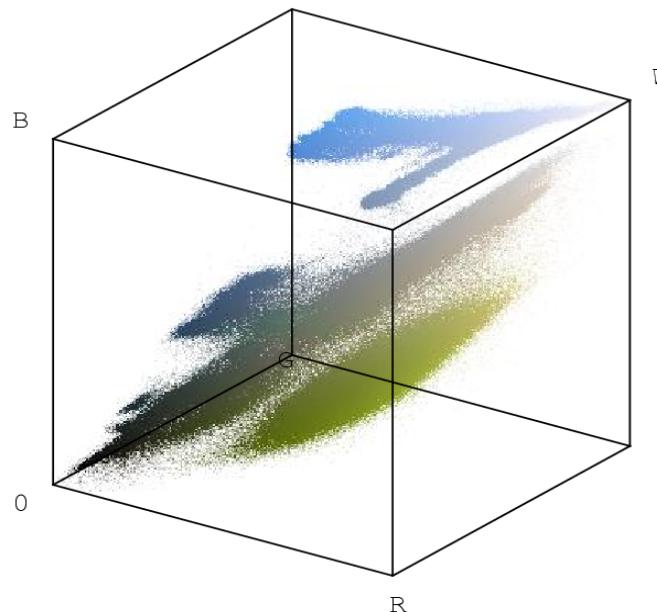
Skalare Farbquantisierung (2)

Beispiel:
24 Bit → 8 Bit



Farbverteilung nach skalarer Quantisierung

- Nachteil: Auch nicht vorkommende Farbwerte sind repräsentiert!



Beispiel (Original)



Graphik: Stephen Chenney, Univ. of Wisconsin

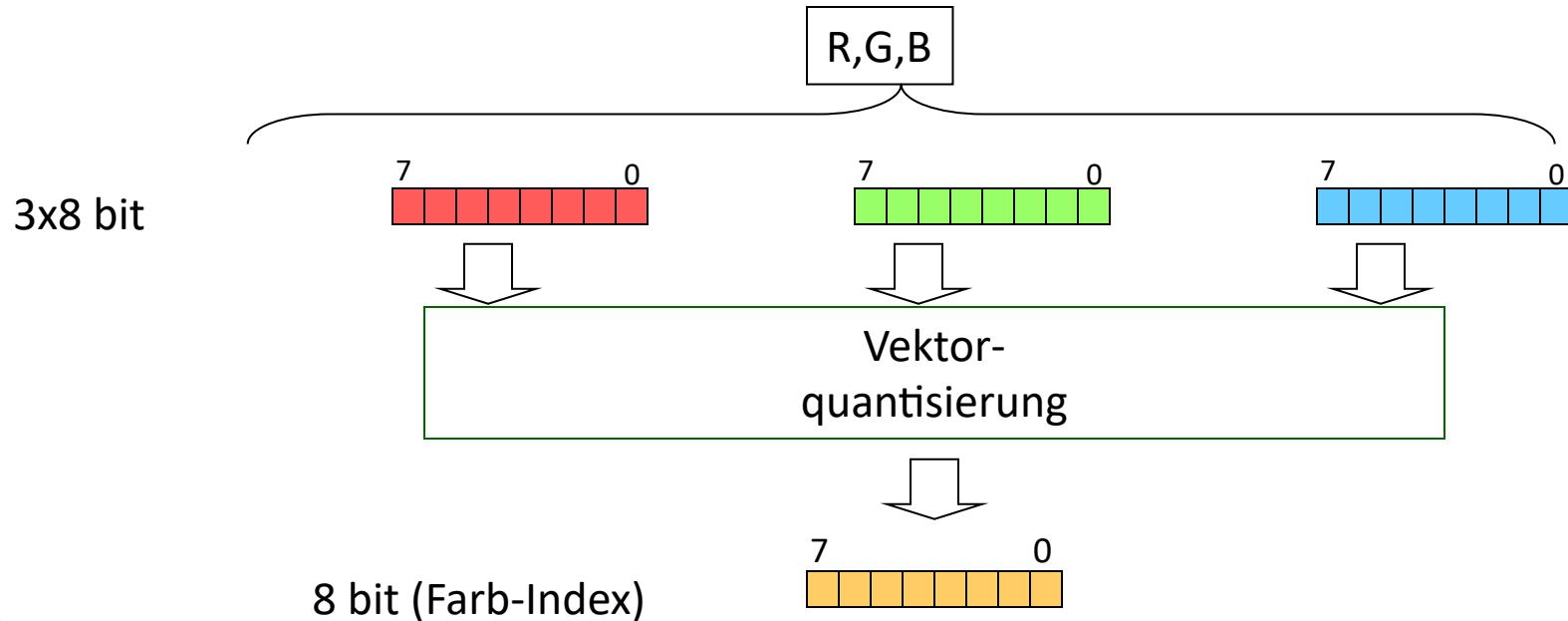
Gleichförmige Quantisierung



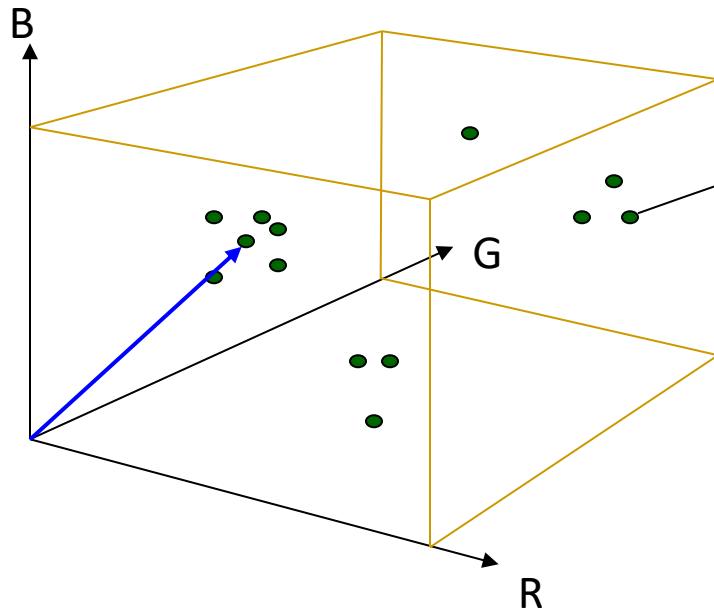
Graphik: Stephen Chenney, Univ. of Wisconsin

Vektor-Quantisierung

- Alle Komponenten werden zusammen als Vektor quantisiert.
- Abhängig von der tatsächlichen Farbverteilung im Bild



Vektor-Quantisierung



Farbwert
eines Pixels

- **Aufgabe:** Finde repräsentative Farbvektoren, die eine ganze Gruppe von Farbwerten ersetzen können.
- **Problem:**
wieviele Repräsentanten?
welche Repräsentanten?
welche Gruppen?

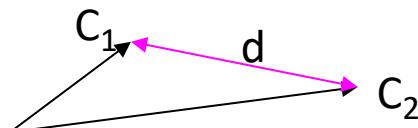
Popularity Alorithmus

1. Berechne, wie oft jeder Farbwert im Bild vorkommt
(meist vorher zu kleinen Zellen vereint).
2. Ermittle die n häufigsten Farbwerte (als Repräsentanten)
3. Ordne alle Farbwerte im Bild dem jeweils „nächstgelegenen“ der n Repräsentanten zu.

Anzahl der verschiedenen Farbwerte im Bild?

- Pixel sortieren und Übergänge abzählen.

Abstand zwischen 2 Farbwerten (Vektoren in r,g,b) C_1, C_2 :



$$d = |C_1 - C_2|$$

Beispiel (Original)



Graphik: Stephen Chenney, Univ. of Wisconsin

Popularity-Algorithmus

Problem: Selten vorkommende aber wichtige Farben werden evtl. völlig ignoriert (blau?).



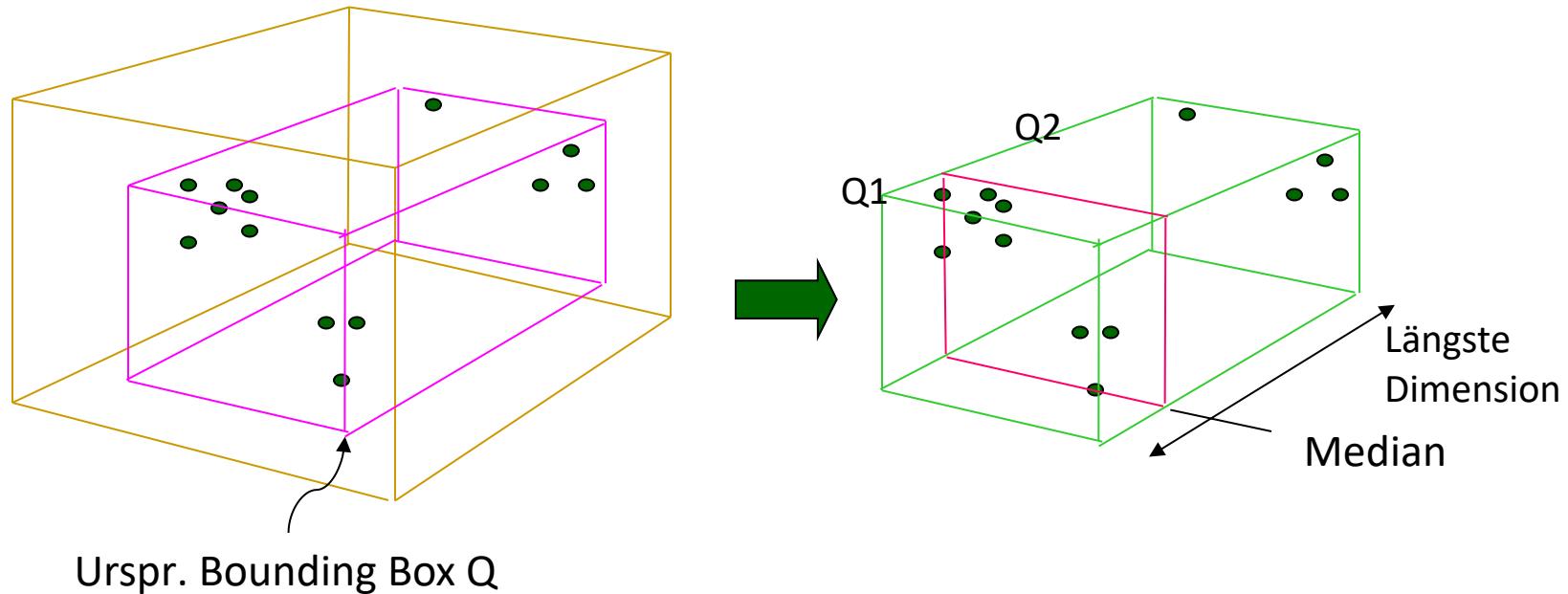
Graphik: Stephen Chenney, Univ. of Wisconsin

Median Cut Algorithmus (Heckbert 1982)

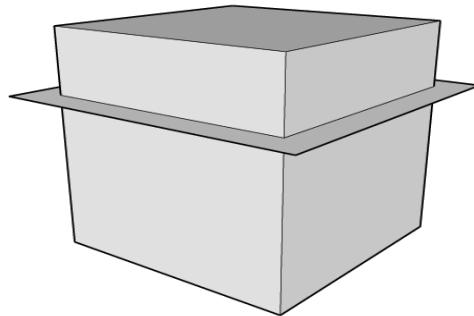
Idee: jeder Farb-Repräsentant soll gleich viele Pixel im Bild repräsentieren.

1. Berechne die „Bounding Box“ (Quader) Q der Farbverteilung im Farbraum (min/max in jeder Dimension).
2. Wiederhole bis N Quader erzeugt wurden:
 - a) Bestimme die längste Dimension des Quaders Q
 - b) Teile Quader Q an der Stelle des Median in Quader Q1 und Q2
 - c) Wende Schritt 2 rekursiv auf Q1 und Q2 an.
3. Ordne allen Farbpixeln den durchschnittlichen Farbwert des umhüllenden Quaders zu.

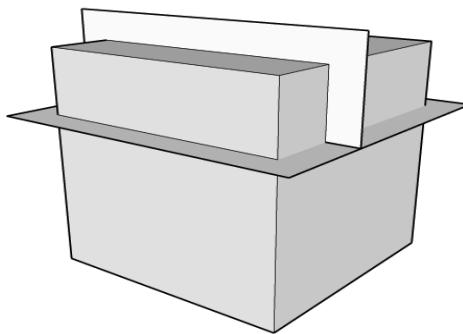
Median Cut Algorithmus



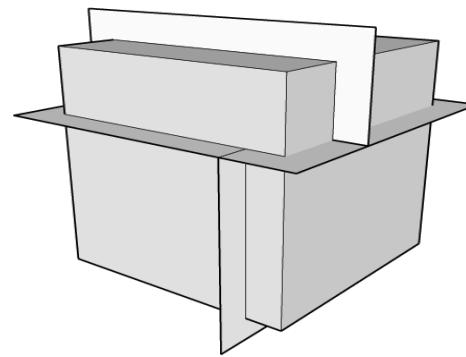
Median Cut



1. Schnitt



2. Schnitt



3. Schnitt

Beispiel (Original)



Graphik: Stephen Chenney, Univ. of Wisconsin

Median Cut

- Gute Ergebnisse in der Praxis.
- Teilt Farbraum in geeigneter Weise auf.
- Schneller Algorithmus, der häufig eingesetzt wird.

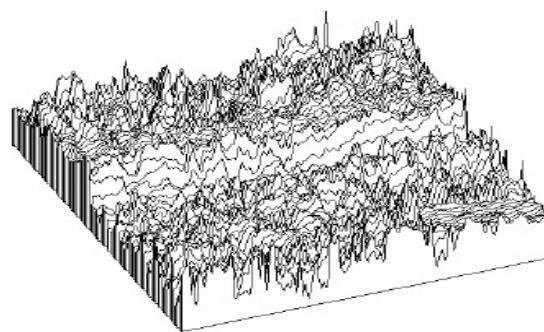


Graphik: Stephen Chenney, Univ. of Wisconsin

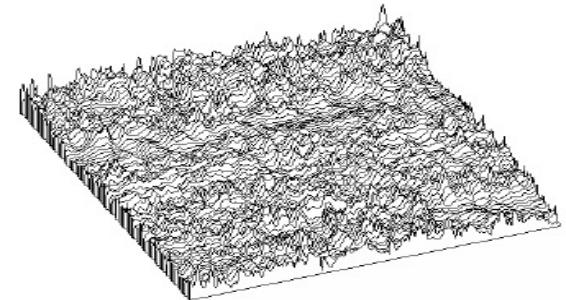
Quantisierungsfehler



(a) Detail



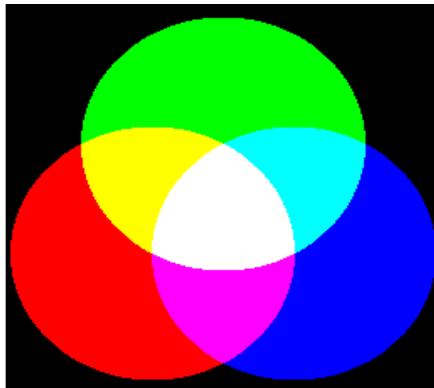
(b) 3:3:2



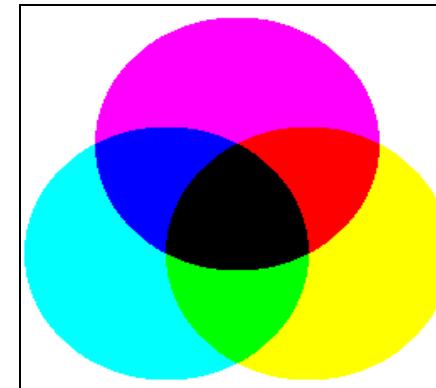
(c) Median-Cut

Farträume für den Druck – CMY und CMYK

- Subtraktive Farbmischung
- Primärfarben **Cyan (C)**, **Magenta (M)**, **Yellow (Y)**



RGB
(additiv)



CMY
(subtraktiv)

CMY und CMYK

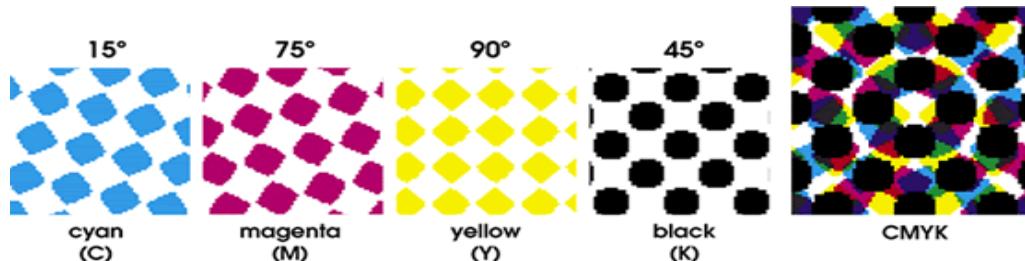
- Subtraktive Farbmischung:

- keine Druckfarbe (weiß): **C=M=Y=0**
- volle Sättigung (schwarz): **C=M=Y=1**

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

CMY
Basis-
modell

- CMY wird zur Vergrößerung des Farbbereichs in der Praxis durch **Schwarz (K)** ergänzt: → CMYK
 - Wieviel Schwarzanteil...?



Berechnung des Schwarzanteils: CMY -> CMYK

- Einfache Konversion:
Schwarzanteil K

$$K = \min(C, M, Y)$$

- Version 1

- Farbkomponenten werden einfach durch den Schwarzanteil (K) reduziert.

$$\begin{pmatrix} C' \\ M' \\ Y' \\ K' \end{pmatrix} \leftarrow \begin{pmatrix} C - K \\ M - K \\ Y - K \\ K \end{pmatrix}$$

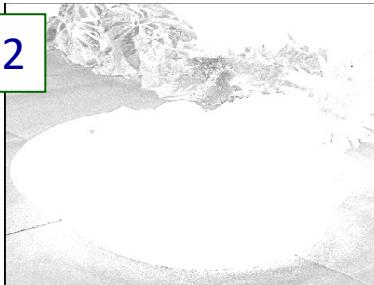
- Version 2

- Geringere Farbreduktion (kräftigere Farben) in dunklen Bereichen.

$$\begin{pmatrix} C' \\ M' \\ Y' \\ K' \end{pmatrix} \leftarrow \begin{pmatrix} C - K \\ M - K \\ Y - K \\ K \end{pmatrix} \cdot \begin{cases} \frac{1}{1-K} & \text{für } K < 1 \\ 1 & \text{sonst} \end{cases}$$

RGB -> CMYK: Einfache Konvertierung vs. Photoshop

Cyan (C)



Magenta (M)



Yellow (Y)



Black (K)

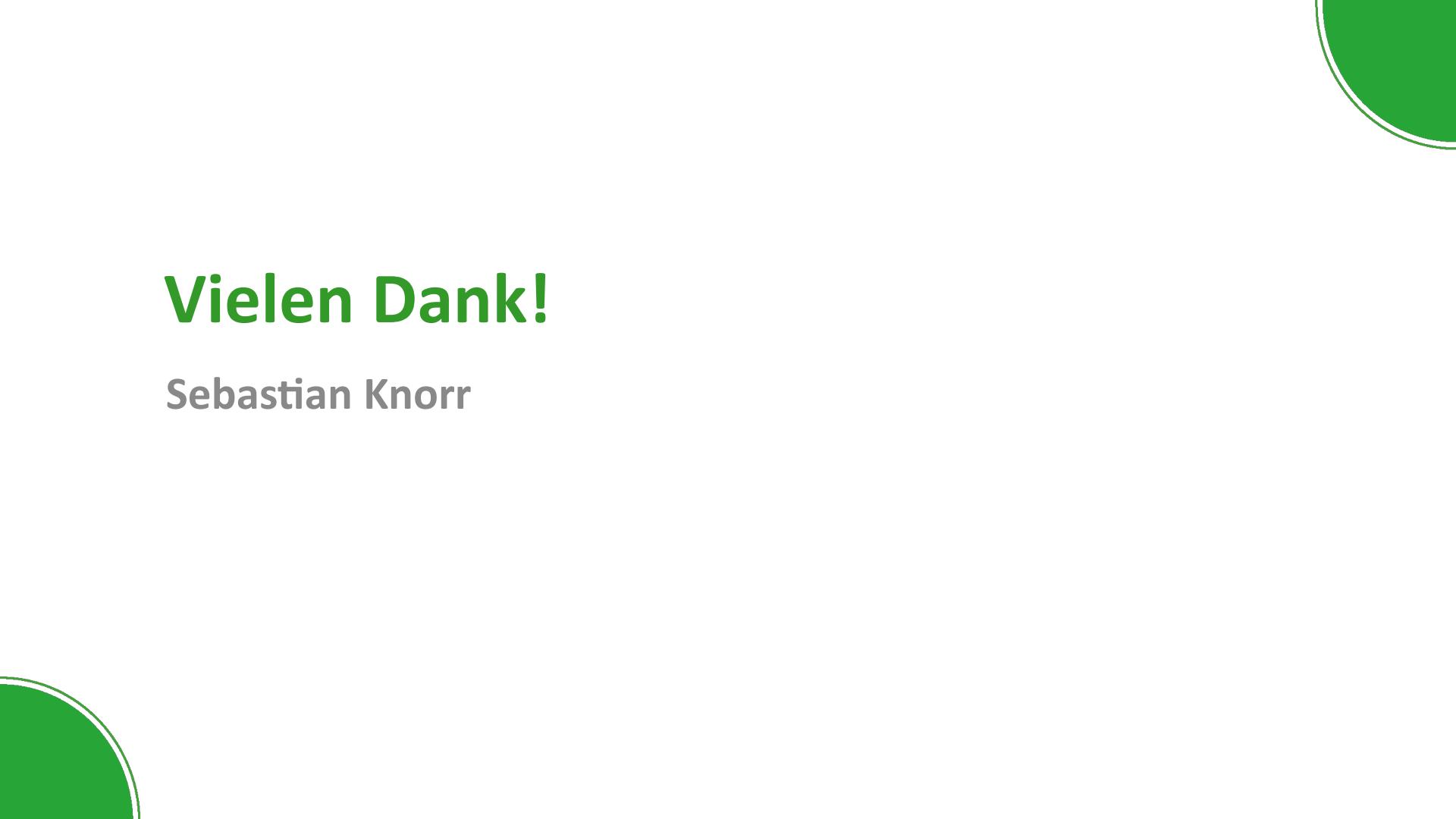


Version 2

Photoshop

Nächste Vorlesung:

Digitale Bilder und Formate



Vielen Dank!

Sebastian Knorr