

Multi-Media

RGB (Red Green Blue)

RGB (252, 178, 92) = Orange

#ff0000 = Red

#00ff00 = Green

#0000ff = Blue

#ffff00 = Yellow

#000000 = Black

#00bc00 = Green

CMYK (Cyan Magenta Yellow Key)

C: 100% M: 0% Y: 0% K: 0% = Cyan

C: 0% M: 100% Y: 100% K: 0% = Red

C: 100% M: 100% Y: 0% K: 0% = Blue

C: 100% M: 100% Y: 100% K: 0% = Black

C: 0% M: 0% Y: 0% K: 100% = Black

- R = Rastergrafik
- V = Vektorgrafik

Format	Browser unterstützt	Grafiktyp	a-Kanal	Transparenzfarbe	Geeignet für...
JPG	Ja	R	Nein	Nein	Bilder, verlustfreie Komprimierung
GIF	Ja	R	Nein	Ja	Grafik, Bildcompositing, Animation
PNG	Ja	R	Ja	Nein	Bilder, 2D-Grafik
TIF	Nein	R	Ja	Nein	Bilder für Archivierung, Verlustlos
BMP	Ja	R	Nein	Nein	Bilder, kein bevorzugtes Format
WEBP	Ja	R	Ja	Nein	Bilder, Animationen
SVG	Ja	V	n.a.	n.a.	2D-Grafik

RGB, CMYK, HEX-Darstellung von RGB und CMYK

YCrCb Nur Umrechnung RGB zu Y (Luminanz)

$$Y = 0.299 * R + 0.587 * G + 0.114 * B$$

Verständnisfragen

- Alphakanal → Transparenzkanal (z. B. in PNG)
- GIF-Format
 - Farbtabelle: Nur 256 Farben
 - Transparenzfarbe: Nur eine Farbe kann transparent sein

C: 0% M: 46% Y: 38% K: 22% = Pink-Red

DPI (dots per inch) -> Drucker

- dpi = Anzahl Farbpunkte pro 2.54cm Breite.
- 1 Inch = 1 Zoll = 2.54cm
- Formel: $\text{px} / \text{dpi} = \text{x cm}$

PPI (pixel per inch) -> Bildschirm

- ppi = Anzahl Pixel pro 2.54cm Breite.
- Formel: $x = D / \sqrt{a^2 + b^2}$
 - Horizontale Pixel = $a \times x \times \text{ppi}$
 - Vertikale Pixel = $b \times x \times \text{ppi}$

Speicherbedarf RGB (1B/Kanal)

Bsp. 1000x600 Pixel: $1000 \times 600 \times 3 = 1'800'000 \text{ Byte}$

- Formel - HD (x) Seitenverhältnis (a:b) :
 - $x / a * b = y \text{ Pixel/Bildzeile}$
 - $x * y * 3 = z \text{ Byte/Bild}$
 - Speicherbedarf bei v-min-Video (60 * v) mit 25 Bilder/s (s):
 - $z * s * 60 * v = w \text{ Byte auf Speichermedium.}$

Bildformate

- Animation: Unterstützt einfache Animationen
- Rastergrafik vs. Vektorgrafik
 - Rastergrafik: Pixelbasiert (Fotos, JPG, PNG)
 - Vektorgrafik: Mathematisch beschrieben (SVG, AI) → Skalierbar ohne Qualitätsverlust

Analog-Digital-Umwandlung

1. Sampling → Zeitdiskretisierung (z. B. 44.1 kHz für Audio)
2. Quantisierung → Wertdiskretisierung (Bit-Tiefe)
3. Kodierung → Speicherung als Binärdaten

Videonormen

Norm	Auflösung	Bildmodus
HD 720p	1280x720	Progressiv
Full HD 1080p	1920x1080	Progressiv

Speicherplatzberechnung bei Bildern/Videos

Bildspeicherplatz (unkomprimiert): $\text{Speicher (Bytes)} = \text{Breite} \times \text{Höhe} \times \text{Farbtiefe (in Byte)}$

Videospeicherplatz: $\text{Gesamtgröße} = \text{Bildgröße} \times \text{FPS} \times \text{Dauer (Sekunden)}$

Codes vs. Container

- Codecs → Komprimieren & dekomprimieren Daten (z. B. H.264, VP9)

- **Container** → Verpackt Video, Audio, Untertitel (z. B. MP4, MKV, AVI)

Verlustbehaftete Komprimierung

- **Farbreduktion** → Weniger Farben (GIF, PNG)
- **Auflösungsreduktion** → Pixelanzahl verringern
- **Subsampling (Farbinformationen reduzieren)**
 - **4:4:4** → Keine Reduktion
 - **4:2:2** → Horizontale Farbinformation halbiert
 - **4:1:1** → Noch stärkere Reduktion
 - **4:2:0** → Horizontale & vertikale Reduktion

Subsampling-Berechnung

$$\text{Ersparnis} = 100\% - \left(\frac{\text{Datenmenge mit Subsampling}}{\text{Originaldatenmenge}} \times 100\% \right)$$

JPG Blockartefakte

- Durch DCT entstehen 8×8-Pixelblöcke
- **Starke Komprimierung** → Sichtbare Blockartefakte

Intraframe vs. Interframe-Komprimierung

- **Intraframe (innerhalb eines Bildes)** → Komprimierung jedes Einzelbildes
- **Interframe (über mehrere Bilder hinweg)** → Komprimiert durch Vergleich mit vorherigen/nächsten Bildern

GOP (Group of Pictures) Sequenz

- **I-Frame (Intra-Frame):** Vollständiges Bild
- **P-Frame (Prädiktiv-Frame):** Speichert nur Änderungen zum letzten Bild
- **B-Frame (Bidirektional-Frame):** Speichert Unterschiede zu vorherigen & nächsten Bildern
- **Beispiel:** GOP25 → Alle 25 Bilder enthält ein vollständiges I-Frame