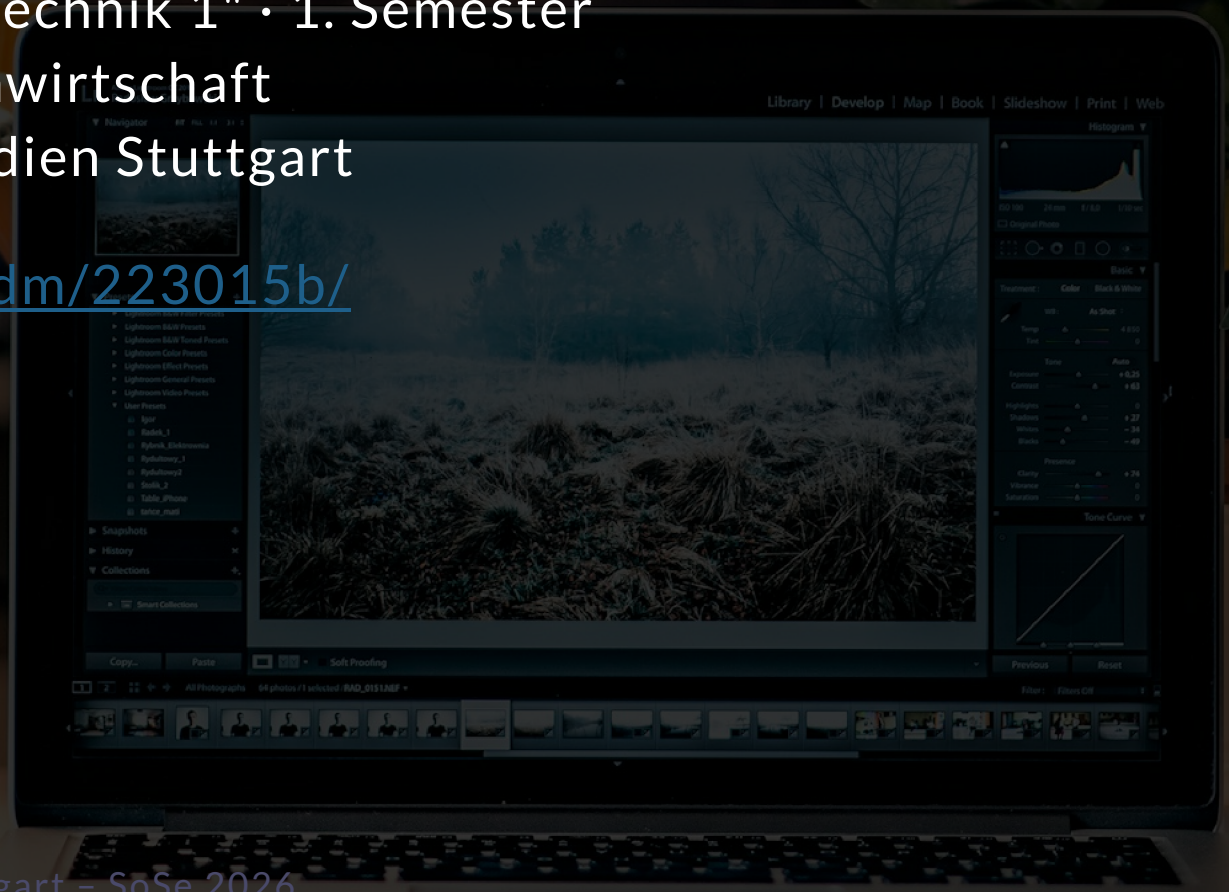
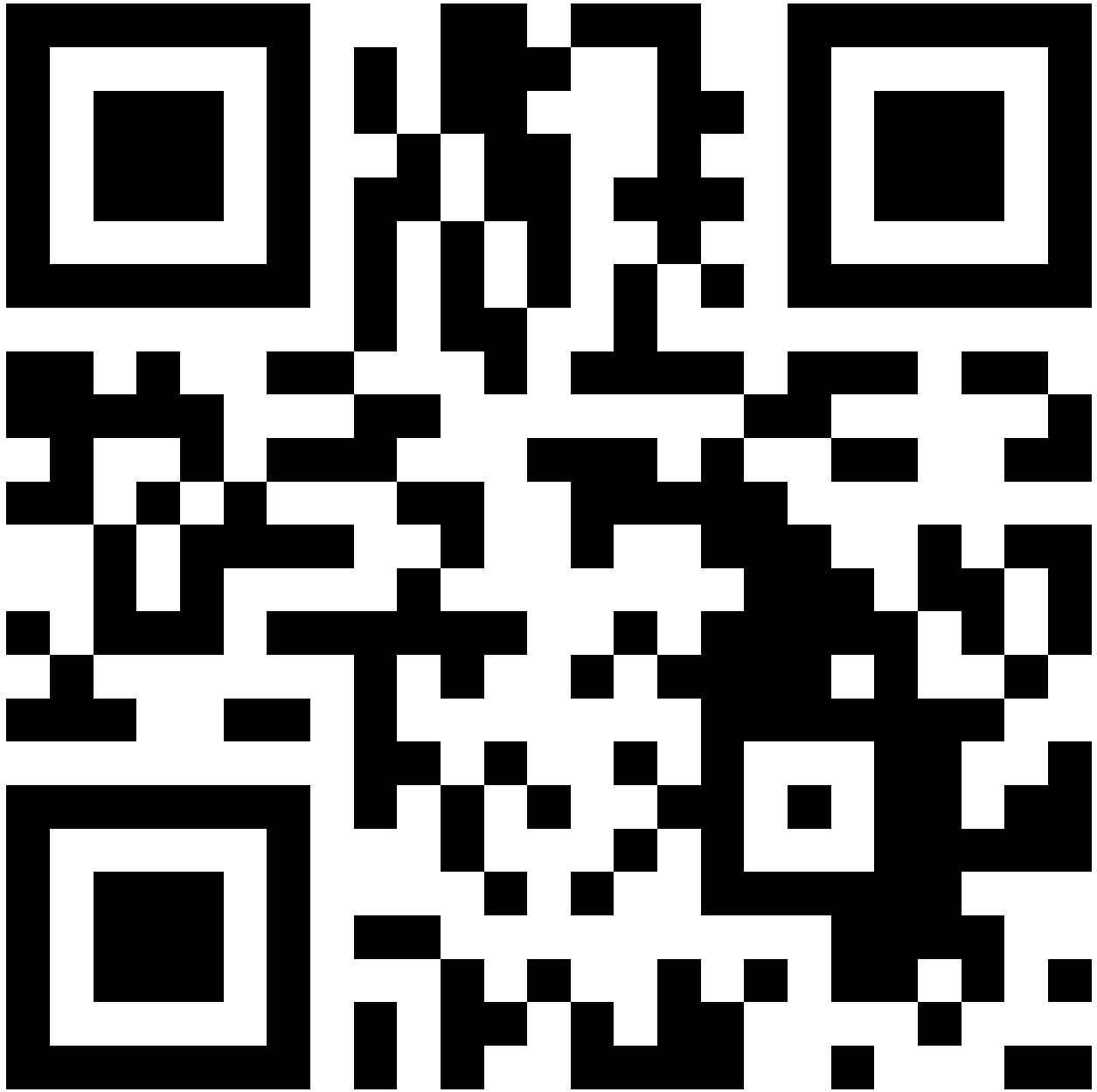
Dateiformate, Schnittstellen, Speichermedien & Distributionswege

223015b · Modul "Technik 1" · 1. Semester
Digital- und Medienwirtschaft
Hochschule der Medien Stuttgart

https://librete.ch/hdm/223015b/





Teil 2: Bild- & Videoformate

Warum verschiedene Dateiformate?

Ein Dateiformat definiert:

- Ob und wie Daten komprimiert werden
- Welche Metadaten enthalten sind
- Wie Daten *codiert* und *decodiert* werden (*Co·dec*)

Ziel	Bild	Audio	Dokument
Kleine Dateien	JPEG	MP3	—
Perfekte Qualität	PNG, RAW	FLAC	PDF
Animation/Video	GIF	—	—
Skalierbarkeit	SVG	—	PDF


Original Uncompressed
Heavy Compression

Digitale Bilder

Raster- und Vektorgrafiken

Was ist ein digitales Bild?

Ein digitales Bild ist ein Raster aus Farbpunkten (Pixel).
Jeder Pixel speichert einen RGB-Farbwert (3 Bytes).

Beispiel: Full HD (1920×1080)
= 2.073.600 Pixel × 3 Bytes = **6,2 MB**

Rastergrafiken

Aufbau: Liste von Pixeln mit Farbwerten (2D-Array)

Speicherbedarf (unkomprimiert):
Breite × Höhe × Farbtiefe (in Bytes)

Beispiele: JPEG, PNG, WebP

Bits (Farbtiefe)	Farben	Anwendung
1	2	Schwarz/Weiß (Fax)
8	256	Graustufen, GIF
24	16,7 Mio.	True Color (Standard)
32	16,7 Mio. + Alpha	Transparenz

Rastergrafik – Vertiefung

Ein Pixel ist die kleinste adressierbare Einheit. Bei 24-Bit-Farbtiefe speichert jeder Pixel drei 8-Bit-Werte (0–255) für Rot, Grün und Blau. Diese additive Farbmischung erzeugt 256^3 = 16,7 Millionen mögliche Farbtöne.

Speicherberechnung: `Breite × Höhe × Bytes pro Pixel`

Auflösung	Farbtiefe	Berechnung	Größe
1920×1080	24 Bit (3 B)	2.073.600 × 3	6,2 MB
4K (3840×2160)	24 Bit	8.294.400 × 3	24,9 MB
4K	32 Bit (4 B)	8.294.400 × 4	33,2 MB

Der Alpha-Kanal (32 Bit) speichert Transparenz als Wert von 0 (unsichtbar) bis 255 (vollständig sichtbar). PNG nutzt dies für weiche Kanten; JPEG unterstützt keinen Alpha-Kanal.

Das Problem der Skalierung

Vergrößern:
Fehlende Pixel müssen erfunden werden (Interpolation)

Verkleinern:
Pixel müssen zusammengefasst werden

Interpolationsverfahren:

- Nearest Neighbor: Schnell, pixelig
- Bilinear: Glättet, Standardverfahren
- Bicubic: Hohe Qualität, rechenintensiv
- Lanczos: Beste Qualität, mathematisch komplex

Vektorgrafiken

Speicherung als geometrische Primitive:

- Pfade (Bézierkurven mit Kontrollpunkten)
- Grundformen (Rechteck, Ellipse, Polygon)
- Text (Glyphen als Outlines)

SVG-Beispiel:

```
<circle cx="50" cy="50" r="40" fill="#ff0000"/>
```

SVG beschreibt WAS gezeichnet werden soll, nicht WIE jeder Pixel aussieht.

Vektorgrafik – Vertiefung

Vektorgrafiken speichern keine Pixel, sondern mathematische Beschreibungen: Koordinaten, Kurvenparameter (Bézier-Kontrollpunkte), Füllfarben, Strichstärken. Der Renderer berechnet die Pixel erst bei der Ausgabe – daher beliebig skalierbar.

Bézier-Kurven (Pierre Bézier, 1962, für Renault-Karosserien entwickelt):

- Definiert durch Ankerpunkte und Kontrollpunkte
- Kubische Bézier: 2 Anker + 2 Kontrollpunkte
- Mathematisch exakt reproduzierbar

Aspekt	Raster	Vektor
Skalierung 10×	Pixel sichtbar	Perfekt scharf
Foto-Realismus	Gut geeignet	Unpraktikabel
Dateigröße Logo	Wächst mit Auflösung	Konstant (~5 KB)
Editierbarkeit	Destruktiv	Nicht-destruktiv

Rasterisierung: GPU wandelt Vektordaten in Pixel um. Geschieht bei jeder Darstellung neu – deshalb ist ein 4K-Monitor schärfer als ein 1080p-Monitor bei gleichem SVG.

Raster- und Vektorgrafiken

	Raster	Vektor
Optimal für	Fotos, komplexe Bilder	Logos, Icons, Illustrationen
Skalierung	Qualitätsverlust	Verlustfrei
Dateigröße	Abhängig von Auflösung	Abhängig von Komplexität
Formate	JPEG, PNG, WebP	SVG, PDF, AI
Bearbeitung	Pixel-basiert	Objekt-basiert

Menschliche Wahrnehmung

Psychovisuelle Kompression

Die Schwächen des Auges

Menschen sehen:

- Helligkeit besser als Farbe
- Große Flächen besser als feine Details
- Niedrige Frequenzen besser als hohe

JPEG nutzt das aus:

- Farbauflösung reduzieren (Helligkeit behalten)
- Glatte Flächen effizient speichern
- Hohe Frequenzen (feine Details) verwerfen

Psychovisuelle Wahrnehmung – Vertiefung

Die Netzhaut enthält ~120 Mio. Stäbchen (Helligkeitswahrnehmung) aber nur ~6 Mio. Zapfen (Farbwahrnehmung). Dieses 20:1-Verhältnis erklärt, warum JPEG Farbinformationen stärker reduzieren kann als Helligkeitsinformationen.

Räumliche Frequenz beschreibt, wie schnell sich die Helligkeit über eine Bildfläche ändert:

- **Niedrig:** Himmel, Wand – große einheitliche Flächen
- **Hoch:** Haare, Texturen, Schrift – schnelle Wechsel

Das Auge ist ein Tiefpassfilter: Hohe Frequenzen (feine Details) werden schwächer wahrgenommen. JPEG verwirft daher zuerst die hohen Frequenzen – der Qualitätsverlust bleibt meist unsichtbar.

Biologisches Limit	Ausnutzung in JPEG
Farbauflösung ~20× geringer	Chroma Subsampling (4:2:0)
Hohe Frequenzen unscharf	DCT + Quantisierung
Kontrast-Maskierung	Artefakte in Texturen versteckt

Grenzen der Kompression: JPEG-Artefakte

Bei starker Kompression sichtbar:

- **Posterization:**
  Farbverläufe werden stufig
- **Blocking:**
  8×8-Blöcke werden sichtbar
- **Ringing:**
  "Geister" an scharfen Kanten

JPEG-Qualität in der Praxis

Quality	Typische Größe (12 MP)	Artefakte
100	2–3 MB	Minimal
85–90	200–400 KB	Kaum sichtbar
60	~100 KB	Bei genauem Hinsehen
30	~50 KB	Deutlich sichtbar

Sweet Spot: 85–90

~10× Kompression, für Menschen kaum unterscheidbar

JPEG-Kompression

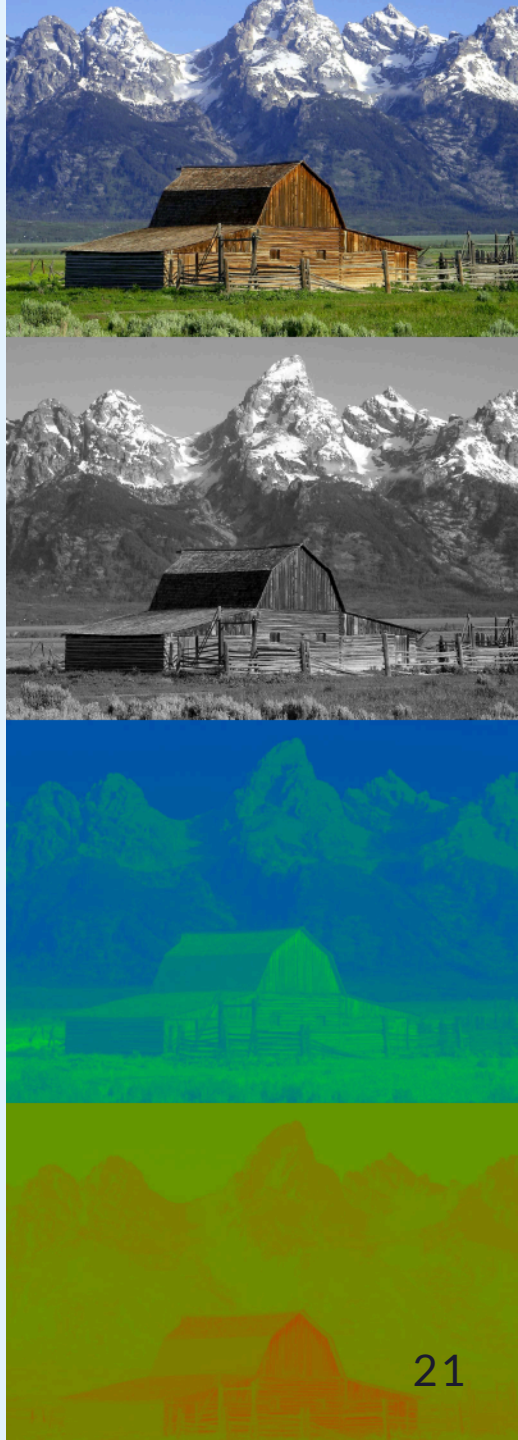
Sechs Schritte im Detail

JPEG Schritt 1: Farbraumkonversion

RGB → Y'CbCr

- **Y** = Helligkeit (Luminanz)
- **Cb** = Blau-Gelb-Anteil (Chrominanz)
- **Cr** = Rot-Grün-Anteil (Chrominanz)

Warum?
Y (Helligkeit) behält volle Auflösung
Cb/Cr (Farbe) kann reduziert werden



Farbraumkonversion – Vertiefung

RGB → YCbCr nutzt die Biologie des menschlichen Auges: 120 Mio. Stäbchen (Helligkeit) vs. nur 6 Mio. Zapfen (Farbe) – ein 20:1-Verhältnis. Die Transformation erfolgt über eine lineare Matrix:

```
Y  =  0.299·R + 0.587·G + 0.114·B
Cb = -0.169·R - 0.331·G + 0.500·B + 128
Cr =  0.500·R - 0.419·G - 0.081·B + 128
```

Die Gewichtung (G dominiert mit 59%) entspricht der spektralen Empfindlichkeit des Auges bei Tageslicht. Y enthält alle Schärfeinformation; Cb/Cr können reduziert werden.

Chroma Subsampling – Notation J A b (bezogen auf 4×2 Pixel):

Schema	Farbdaten	Einsatz
4:4:4	100%	Postproduktion, Grafik
4:2:2	50%	Broadcast, Pro-Video
4:2:0	25%	JPEG, H.264, Streaming

Bei 4:2:0 teilen sich 4 Pixel einen Farbwert, behalten aber individuelle Helligkeit → 50% Dateneinsparung bei kaum sichtbarem Unterschied.

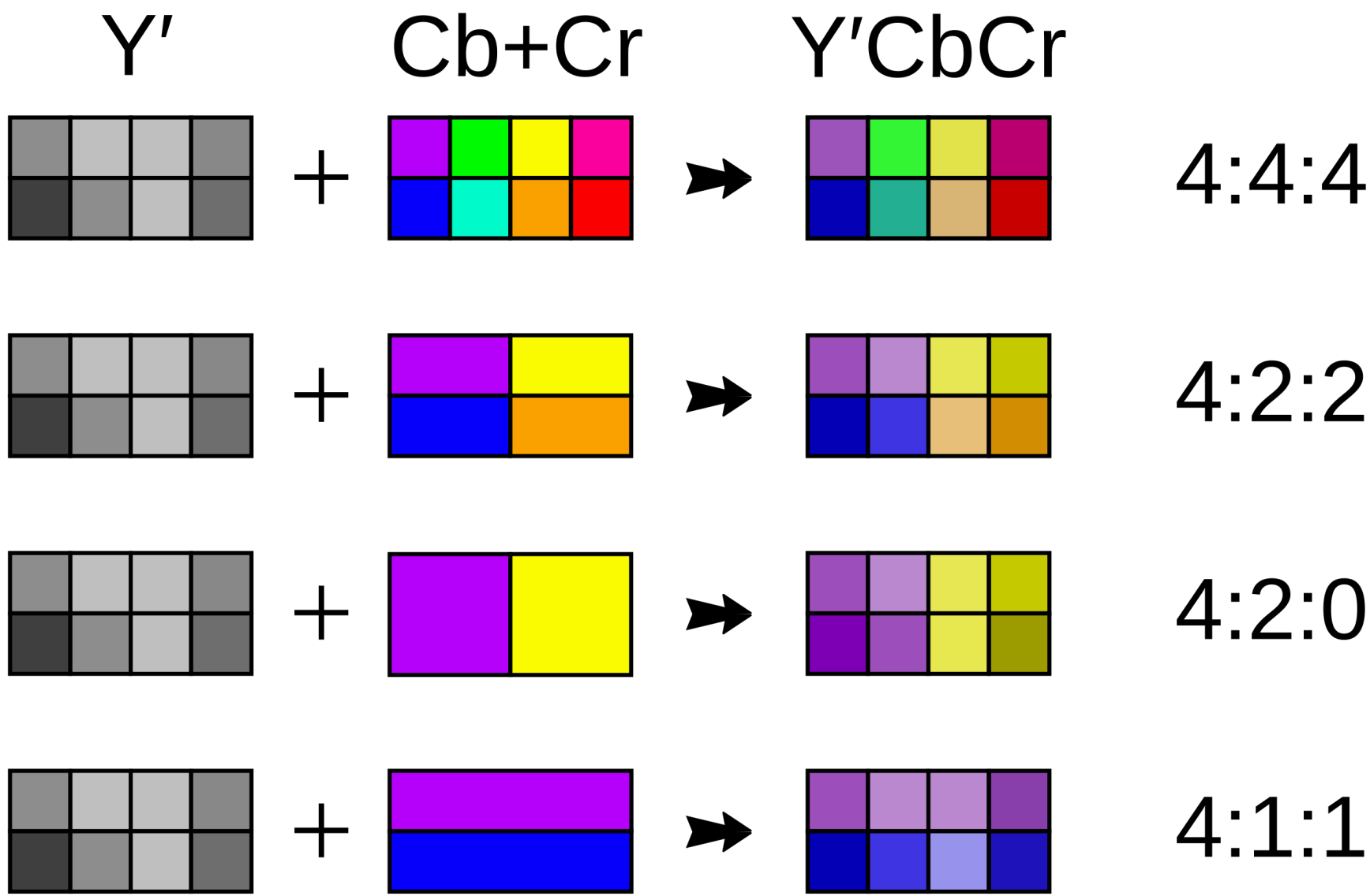
JPEG Schritt 2: Chroma Subsampling

Notation `J:a:b` (bezogen auf 4×2 Pixel-Block):

- **J** = Referenzbreite (immer 4)
- **a** = Farbsamples in Zeile 1
- **b** = Farbsamples in Zeile 2

Schema	Bedeutung	Farbdaten
4:4:4	Volle Farbauflösung	100%
4:2:2	Halbe horizontale Auflösung	50%
4:2:0	Viertel Auflösung (2×2 teilt Farbe)	25%

4:2:0 ist JPEG-Standard


Y′
Cb+Cr
Y′CbCr
4:4:4
4:2:2
4:2:0
4:1:1

JPEG Schritt 3: Block-Aufteilung

Das Bild wird in 8×8-Pixel-Blöcke zerlegt

Jeder Block wird unabhängig verarbeitet.
Bei 1920×1080: 240 × 135 = **32.400 Blöcke**

Level Shift:
Pixelwerte um −128 verschieben

- Vorher: 0 bis 255
- Nachher: −128 bis +127

Warum?
DCT arbeitet besser mit Werten um Null

JPEG Schritt 4: DCT (Discrete Cosine Transform)

Jeder 8×8-Block wird transformiert:
64 Pixelwerte → 64 Frequenzkoeffizienten

Die 64 Koeffizienten:

Position	Name	Bedeutung
(0,0)	DC	Durchschnittshelligkeit
Rest	AC	Helligkeitsänderungen

Energy Compaction:
90% der Information in den ersten 10–15 Koeffizienten
DCT selbst ist **verlustfrei** und reversibel!

JPEG Schritt 5: Quantisierung

Hier passiert der Datenverlust!

Die DCT hat sortiert – jetzt wird aufgeräumt:

- Wichtige Werte (niedrige Frequenz) → präzise behalten
- Unwichtige Werte (hohe Frequenz) → vergröbern oder Null setzen

Ergebnis:
Von 64 Werten pro Block bleiben oft nur 5–15 übrig
Rest wird zu Nullen → extrem gut komprimierbar

Quality-Einstellung:
Hoch = mehr Werte behalten = größere Datei
Niedrig = mehr Nullen = kleinere Datei, mehr Artefakte

Original Image

“Quantized” Image

https://newsletter.maartengrootendorst.com/p/a-visual-guide-to-quantization

JPEG Schritt 5b: Zigzag & RLE

Nach Quantisierung: Viele Nullen (v.a. bei hohen Frequenzen)

Zigzag-Scan: Matrix diagonal durchlaufen
→ Nullen sammeln sich am Ende

RLE (Run-Length Encoding):
`0 0 0 0 0 0 0 0` → `(8, 0)` = "acht Nullen"

JPEG Schritt 6: Huffman-Coding

Verlustfreie Kompression der Restwerte

Idee: Variable Bitlänge statt fester 8 Bit
Häufige Werte → kurze Codes

Zeichen	Häufigkeit	Code
e	40%	`0` (1 Bit)
a	25%	`10` (2 Bit)
i	20%	`110` (3 Bit)
o	10%	`1110` (4 Bit)
u	5%	`1111` (4 Bit)

Huffman-Coding – Vertiefung

David Huffman entwickelte 1952 als Student am MIT einen optimalen Algorithmus für präfixfreie Codes – ursprünglich als Hausaufgabe, die zur Veröffentlichung führte.

Algorithmus (Bottom-Up-Baumkonstruktion):

1. Alle Symbole nach Häufigkeit sortieren
2. Die zwei seltensten Symbole zu einem Knoten kombinieren
3. Wiederholen bis nur noch die Wurzel übrig ist
4. Codes ablesen: links = 0, rechts = 1

Präfixfreiheit: Kein Code ist Anfang eines anderen → sofort dekodierbar ohne Trennzeichen.

Eigenschaft	Huffman	Arithmetisch
Einheit	Ganze Bits	Fraktionale Bits
Optimalität	Optimal für ganze Bits	Näher an Entropie
Geschwindigkeit	Schneller	Langsamer
JPEG-Einsatz	Standard (Baseline)	Optional (selten)

JPEG verwendet zwei Huffman-Tabellen: eine für DC-Koeffizienten (Durchschnittswerte), eine für AC-Koeffizienten (Frequenzen). Die Tabellen sind im JPEG-Header gespeichert.


Compre::
JPEG
33
10.0 ns

Andere Bildformate

PNG, GIF, WebP, AVIF

15:33 Uhr weiter

PNG: Verlustfrei mit Transparenz

PNG = Portable Network Graphics (1996)

Entstehung: GIF-Patent-Streit → Community entwickelt Alternative

Features:

- Verlustfrei (Lossless)
- Alpha-Transparenz (8-Bit, 256 Stufen)
- Millionen Farben (24/48 Bit)
- Patent-frei

Ideal für: Grafiken, Screenshots, Text, Logos

GIF: Der Meme-Veteran

GIF = Graphics Interchange Format (1987)

Features:

- 256 Farben (8-Bit Palette)
- Verlustfrei (innerhalb der Palette)
- Animationen

Das Patent-Drama (1994):
Unisys fordert Lizenzgebühren für LZW-Kompression
→ "Burn All GIFs!"-Kampagne
→ PNG als Alternative

Heute: Kulturell unsterblich (Memes, Reaktionen)

WebP & AVIF: Moderne Alternativen

WebP (Google, 2010):

- Lossy und Lossless
- Transparenz und Animationen
- 25–35% kleiner als JPEG

AVIF (2019):

- Basiert auf AV1-Video-Codec
- 50% kleiner als JPEG
- HDR-Unterstützung, patent-frei

Browser-Support 2025: WebP universell, AVIF wächst

WebP & AVIF – Vertiefung

WebP entstand 2010 aus Googles VP8-Videocodec (On2 Technologies, für $133M gekauft). Statt I-Frames für Video werden sie als Einzelbilder verwendet. WebP nutzt Intra-Frame-Prediction, die benachbarte Blöcke zur Vorhersage verwendet – effizienter als JPEGs blockweise DCT.

AVIF basiert auf dem AV1-Videocodec, entwickelt 2015–2018 von der Alliance for Open Media (Google, Apple, Netflix, Amazon, Microsoft, Mozilla). Nach dem Patent-Chaos von H.265/HEVC vereinten sich die Konkurrenten für einen lizenzfreien Standard.

Aspekt	WebP	AVIF
Basis-Codec	VP8/VP9	AV1
Kompression vs. JPEG	25–35% besser	50% besser
HDR/Wide Gamut	Nein	Ja (10/12 Bit)
Encoding-Geschwindigkeit	Schnell	Sehr langsam
Browser-Support 2025	97%+	93%+

Warum JPEG dominiert: Kameras, Bildbearbeitungssoftware und Content-Management-Systeme sind auf JPEG optimiert. Der Wechsel erfordert Infrastruktur-Updates über die gesamte Pipeline.

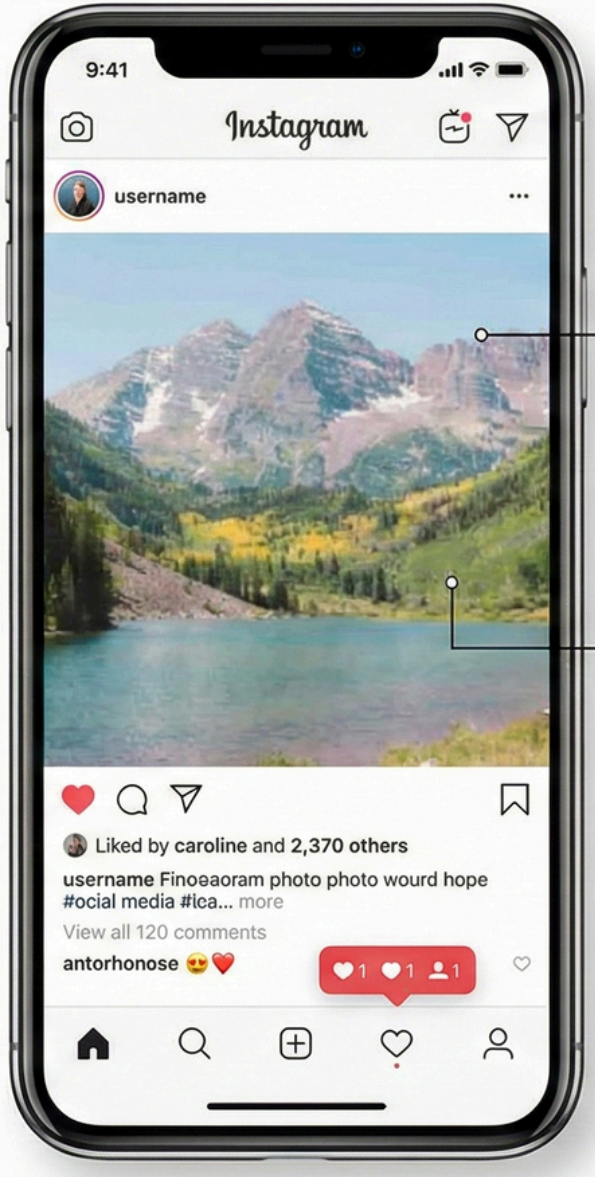
Formatwahl in der Praxis

Anwendung	Format
Fotos fürs Web	JPEG (85), WebP
Screenshots	PNG
Logos, Icons	SVG, PNG
Animationen	GIF, WebP, APNG
Archivierung	TIFF, PNG, RAW
Social Media	Was die Plattform erlaubt

ORIGINAL PHOTO

SOCIAL MEDIA COMPRESSION

INSTAGRAM UPLOAD



Warum Instagram eure Fotos "ruiniert"

Die Upload-Pipeline:

1. Euer Foto: 12 MP, 8 MB
2. Instagram skaliert: max. 1080px Breite
3. Re-Kompression: JPEG Quality ~75
4. Ergebnis: 200–400 KB

Warum?

- Speicherkosten (Milliarden Fotos)
- Ladezeiten (Mobile-First)
- Bandbreite (günstiger für alle)

Video

Bilder + Zeit + Audio

Das Größenproblem bei Video

4K-Video (3840×2160), unkomprimiert:

3840 × 2160 × 3 Bytes = **24,8 MB pro Frame**

× 30 fps = **744 MB/Sekunde**

× 60 Sekunden = **44,6 GB pro Minute**

Ein 2-Stunden-Film: über 5 Terabyte

Container und Codec

Container = Dateiformat (z.B. MP4)
Die "Box", die verschiedene Streams zusammenpackt:

- Video-Stream
- Audio-Stream(s)
- Untertitel
- Metadaten

Codec = Kompressionsalgorithmus (z.B. H.264)
Bestimmt, WIE komprimiert wird

Container vs. Codec – Vertiefung

Container (Multiplexer-Format) organisiert mehrere Datenströme mit Timing-Informationen. Er enthält keine Kompressionslogik, sondern synchronisiert Video, Audio, Untertitel und Metadaten.

Codec (Coder-Decoder) definiert den Kompressionsalgorithmus. Derselbe Container kann verschiedene Codecs enthalten – die Dateiendung verrät den Codec nicht.

Container	Entwicklung	Typische Codecs	Besonderheit
MP4	ISO/MPEG	H.264, H.265, AAC	Web-Standard, DRM-fähig
MKV	Matroska	Alle	Beliebig viele Streams, Kapitel
WebM	Google	VP9, AV1, Opus	HTML5-optimiert, lizenzfrei
MOV	Apple	ProRes, H.264	Professionelle Produktion

Metadaten im Container:

- Timecodes für Frame-genaue Synchronisation
- Kapitelmarken, Thumbnails
- Sprach-Tags für Audio/Untertitel
- HDR-Metadaten (MaxCLL, MaxFALL)

Praktisches Problem: Eine `.mp4`-Datei mit AV1-Codec spielt auf älteren Geräten nicht ab, obwohl sie MP4 „unterstützen" – der Hardware-Decoder fehlt für AV1.

Gängige Container

Container	Verwendung
MP4 (.mp4)	Web, Streaming, universell
MKV (.mkv)	Archiv, viele Streams, offen
MOV (.mov)	Apple-Ökosystem
WebM (.webm)	Web, nur VP9/AV1 + Opus
AVI (.avi)	Legacy, veraltet

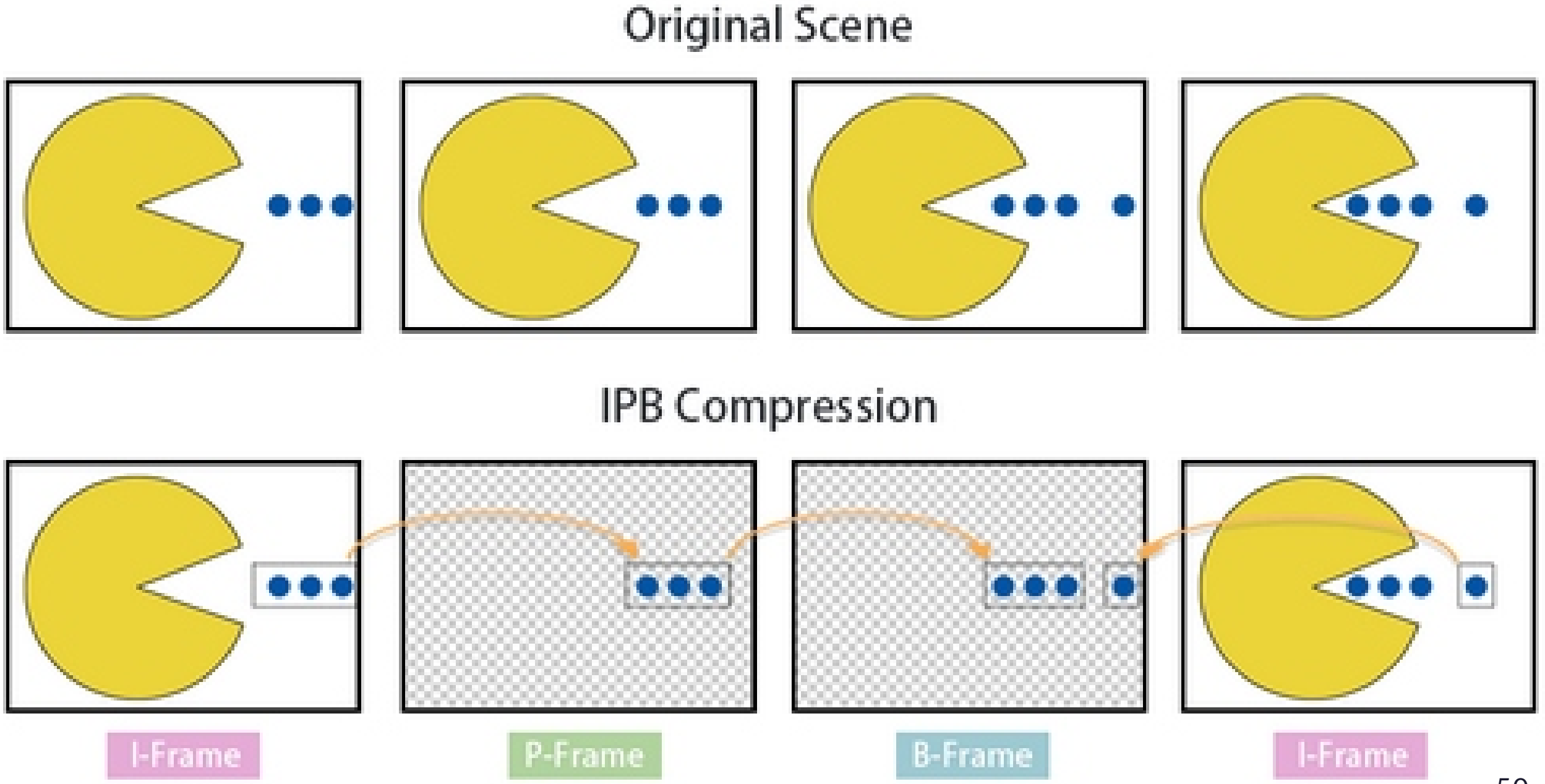
Video-Codecs im Überblick

Codec	Jahr	Status
H.264/AVC	2003	Universal, überall
H.265/HEVC	2013	Effizienter, Patent-Chaos
VP9	2013	YouTube, patent-frei
AV1	2018	Zukunft, patent-frei

Container + Codec = Video

Video-Kompression

Raum und Zeit nutzen


Original Scene
IPB Compression
I-Frame
P-Frame
B-Frame
I-Frame

Drei Kompressionsprinzipien

1. Spatial Compression (Intra-Frame)
Jedes Bild einzeln komprimieren (wie JPEG)

2. Temporal Compression (Inter-Frame)
Nur Änderungen zwischen Bildern speichern

3. Motion Compensation
Bewegung beschreiben statt Pixel kopieren

Spatial Compression (Intra-Frame)

Jedes Bild einzeln komprimieren – wie JPEG

Analysiert Redundanz *innerhalb* eines Frames:

- DCT (Frequenzanalyse)
- Quantisierung
- Entropie-Coding

→ **I-Frame (Keyframe)**
Vollständiges Bild, unabhängig dekodierbar

Temporal Compression (Inter-Frame)

Nur Änderungen zwischen Bildern speichern

Frame-Typ	Referenziert	Typische Größe
I-Frame	Nichts (Keyframe)	100%
P-Frame	Vorherige Frames	~30%
B-Frame	Vorherige + zukünftige	~15%

GOP (Group of Pictures):
I - B - B - P - B - B - P - B - B - I

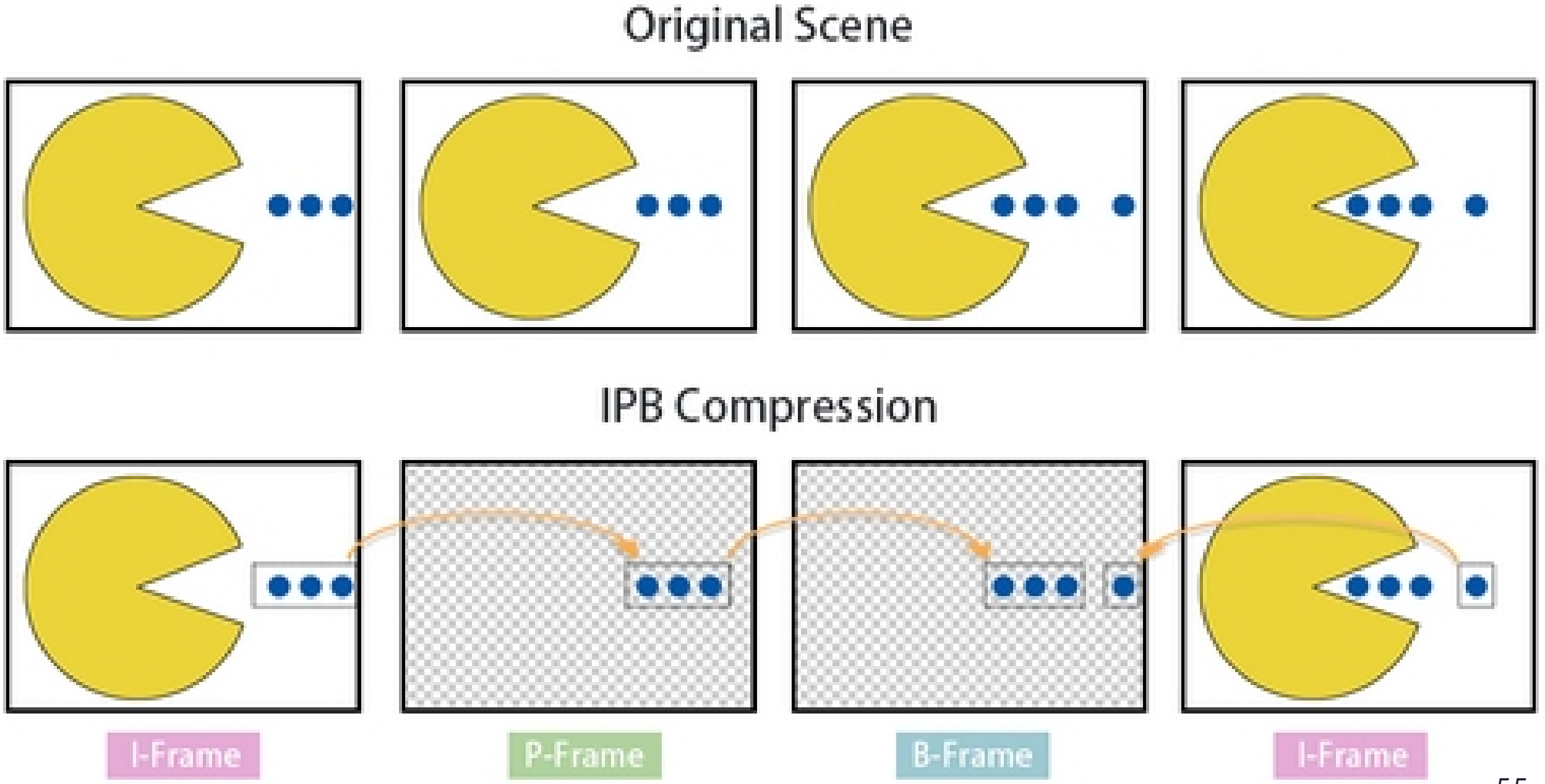
Motion Compensation

Bewegung beschreiben statt Pixel kopieren

Beispiel: Ein 16×16 Pixel-Block

Frame 1: Block an Position (100, 200)
Frame 2: Block an Position (120, 200)

Statt Block zweimal speichern:
→ Motion Vector: "verschiebe um (+20, 0)"


Original Scene
IPB Compression
I-Frame
P-Frame
B-Frame
I-Frame

H.264 / AVC

Advanced Video Coding (2003)

Warum dominant?

- Exzellente Kompression (~100:1 möglich)
- Hardware-Decoder in jedem Gerät seit ~2010
- YouTube, Netflix, Blu-ray – alles H.264

Features:

- Variable Block-Größen (16×16 bis 4×4)
- Deblocking-Filter (reduziert Artefakte)

H.264/AVC – Vertiefung

H.264 (2003) ermöglichte erst YouTube (2005), Netflix-Streaming (2007) und Blu-ray. Vor H.264 war MPEG-2 Standard – H.264 erreichte bei gleicher Qualität die halbe Bitrate.

Technische Innovationen:

- **Variable Blockgrößen:** 16×16 bis 4×4 Macroblocks (MPEG-2: nur 16×16)
- **Intra-Prediction:** Blöcke werden aus Nachbarn vorhergesagt
- **In-Loop Deblocking:** Filter reduziert Blockartefakte vor der Referenzierung
- **CABAC:** Arithmetic Coding ersetzt Huffman (10–15% effizienter)

Profile	Anwendung	Max. Auflösung
Baseline	Videotelefonie, ältere Geräte	480p
Main	Broadcast, Streaming	1080p
High	Blu-ray, professionell	4K

Hardware-Ubiquität: Seit 2010 hat jedes Smartphone, jede GPU, jeder Smart-TV einen H.264-Hardware-Decoder. Encoding in Echtzeit braucht keine CPU – das ermöglichte erst mobiles Video-Streaming und Videotelefonie.

Patent-Pool (MPEG-LA): ~2.000 Patente von 30+ Unternehmen. Endnutzungs-Streaming ist lizenzfrei; Hardware-Hersteller zahlen ~$0,20/Gerät.

Das Patent-Problem

H.264 ist nicht frei

MPEG-LA (Patent Pool):

- 2.000+ Patente von ~30 Unternehmen
- Apple, Microsoft, Sony, Panasonic…

Lizenzgebühren:

- Hardware-Decoder: $0,20 pro Einheit
- "Internet Broadcast": Kostenlos (YouTube etc.)

Problem: Open-Source-Projekte in Grauzone

H.265 / HEVC: Effizienter, aber...

High Efficiency Video Coding (2013)

50% bessere Kompression als H.264

Das Problem: Patent-Chaos

Drei konkurrierende Patent-Pools:

- MPEG-LA
- HEVC Advance
- Velos Media

Unklare Kosten, rechtliche Unsicherheit
→ Viele bleiben bei H.264 oder wechseln zu AV1

VP9: Googles Antwort

VP9 (2013)

Google kaufte On2 Technologies (2010, $133M)
VP8 → VP9 → AV1

Eigenschaften:

- Ähnliche Effizienz wie H.265
- Patent-frei (laut Google)
- YouTube nutzt VP9 für 4K

Nachteil: Höherer CPU-Aufwand als H.264

AV1: Die offene Zukunft

AV1 (2018)

Alliance for Open Media:
Google, Netflix, Amazon, Microsoft, Apple, Mozilla...

Eigenschaften:

- 30% besser als H.265
- Royalty-free, Open Source
- 8K, HDR, hohe Frame-Rates

Stand 2025:
YouTube, Netflix nutzen AV1 für 4K/8K
Hardware-Encoder in aktuellen GPUs

AV1 – Vertiefung

Die **Alliance for Open Media** (2015) vereinte Konkurrenten nach dem Patent-Chaos von H.265/HEVC. Drei separate Patent-Pools (MPEG-LA, HEVC Advance, Velos Media) machten H.265-Lizenzierung unberechenbar – die Industrie wollte einen garantiert lizenzfreien Standard.

Gründungsmitglieder: Google, Mozilla, Cisco, Netflix, Amazon, Microsoft. Apple trat 2018 bei – historisch, da Apple sonst eigene Standards bevorzugt.

Technische Innovation	Beschreibung
Superblocks	Bis 128×128 Pixel (H.264: max 16×16)
Prediction Modes	56 Intra-Modi (H.264: 9)
Transform	10 verschiedene Transformtypen
Film Grain Synthesis	Filmkorn wird als Parameter übertragen

Encoding-Performance: Software-Encoding ist 50–200× langsamer als H.264. Erst Hardware-Encoder (Intel ab Gen 12, NVIDIA RTX 40, Apple M3) machen Echtzeit-Encoding praktikabel.

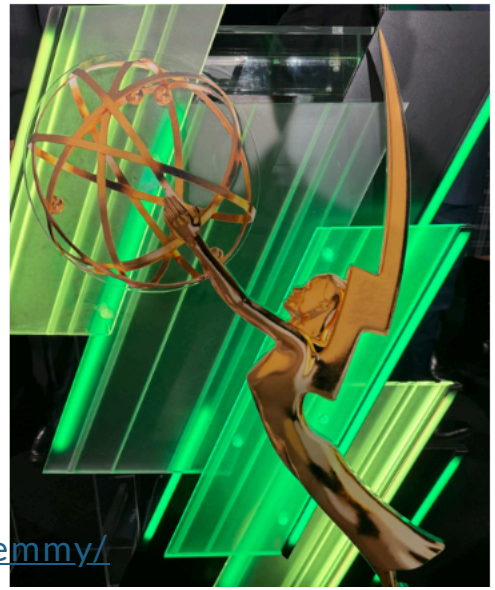
Adoption 2025: YouTube und Netflix nutzen AV1 für 4K/8K-Streams. 2024 gewann AV1 einen Emmy für technische Innovation – offene Standards können Industriestandard werden.

OUR WORK

When a video codec wins an Emmy

DECEMBER 5, 2025 BRIAN GRINSTEAD

It's not every day a video codec wins an Emmy. But yesterday, the Television Academy honored the AV1 specification with a Technology & Engineering Emmy Award, recognizing its impact on how the world delivers video content.



https://blog.mozilla.org/en/mozilla/av1-video-codec-wins-emmy/

Fragen & Diskussion

Kontakt: lb-czechowski@hdm-stuttgart.de
Folien: librete.ch/hdm/223015b

Lizenz & Attribution

Selbstlernen: Bildkompression

1. Öffne squoosh.app
2. Lade ein Foto hoch
3. Vergleiche: JPEG (verschiedene Quality) vs. WebP vs. AVIF
4. Beobachte: Dateigröße, Artefakte, Ladezeit

Fragen zum Erkunden:

- Ab welcher Quality werden Artefakte sichtbar?
- Wie viel kleiner ist WebP bei gleicher Qualität?

Selbstlernen: Video analysieren

1. Video herunterladen (z.B. Big Buck Bunny)
2. Mit MediaInfo analysieren: Container, Codec, Bitrate
3. Optional: Mit HandBrake konvertieren

- H.264 vs. H.265 bei gleicher Qualität
- Größe und Encoding-Zeit vergleichen

Tools:

- MediaInfo (Online oder Desktop)
- HandBrake (Desktop)