

## 3. Bildcodierung und Kompression

Quellen:

- [EVC\\_Skriptum\\_CV, p.15](#) bis [EVC\\_Skriptum\\_CV, p.19](#)

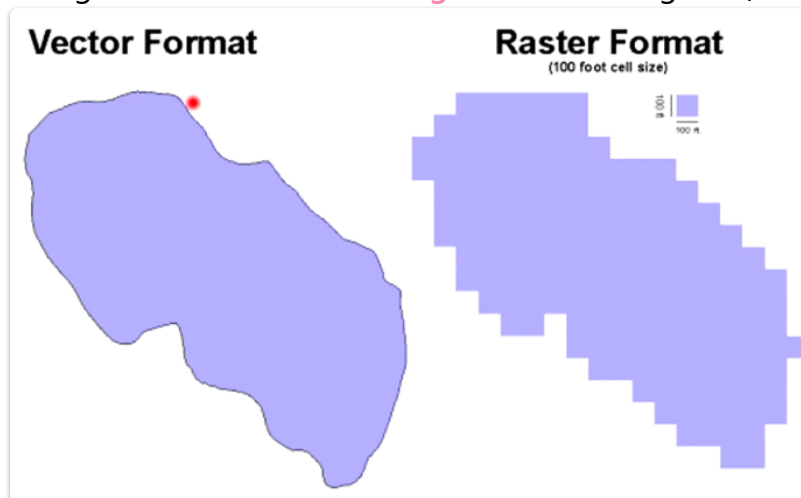
## Digitales Bild-Dateienformat

### Kontinuierliche vs. digitale Daten

- Wenn eine Zahl unendlich viele **mögliche Werte** annehmen kann, spricht man von **kontinuierlichen** oder **analogen** Daten
- **Computer** können mit analogen Daten nicht direkt arbeiten
- Daher müssen diese Daten **digitalisiert** werden, um sie für den Computer bearbeitbar zu machen
- Dieser Umwandlungsprozess erfolgt z.B. beim **Scannen von Fotos** oder beim Fotografieren mit **digitalen Kameras**

### Digitale Bilder

- Ein digitales Bild ist eine **numerische Repräsentation** eines zweidimensionalen Bildes
- Das Bild kann entweder aus **Vektorbeschreibungen** oder einem **Raster** bestehen
- **Rasterbild**: Ein Raster von **diskreten Werten** (Pixel), bei dem jeder **Bildpunkt** mit seinem **Helligkeitswert** oder **Farbwert** gespeichert wird
- **Vektorbild**: Der Bildinhalt wird in **geometrischen Objekten** dargestellt und die Rasterung erfolgt erst bei der **Darstellung** auf einem Endgerät (z.B. **Display** oder **Drucker**)



### Speicherung von Bilddaten

- Digitale Bildinformationen müssen in einem **Bilddatenformat** abgespeichert werden

- In der Frühzeit der digitalen Bildverarbeitung (bis etwa 1985) gab es eine große Anzahl unterschiedlicher **Dateiformate**, was zu vielen notwendigen **Konvertierungsprogrammen** führte
- Heute gibt es standardisierte **Dateiformate**, die den **Austausch** von Bilddaten erleichtern und die **langfristige Lesbarkeit** fördern

## Speichergröße eines Rasterbildes

- Ein Rasterbild enthält ein **Pixelraster**, das für jede Rasterzelle eine bestimmte Anzahl an **Bits** zur Farbgebung bereitstellt – dies entspricht der **Farbtiefe**
- Ein Bild mit **LxN** (L Zeilen und N Spalten), **2B** (B = Anzahl der Bits pro Rasterzelle), **c** Farbkomponenten kann unkomprimiert als:
  - **L x N x B x c** gespeichert werden
  - Beispiel: Bei einer Bildgröße von **1024x768**, **3 Farbkomponenten (RGB)** und **256 (28) Graustufen** ergibt sich:
    - **1024 x 768 x 3 x 8 = 18,87 MBit → 2,36 MByte**
- **Bilddatengröße** korreliert positiv mit der Anzahl der **Pixel** und der **Farbtiefe** (Bits pro Pixel)
- **Prüfungs-ähnliches Beispiel** dazu:

- Wie viel Speicherplatz benötigt man für die Speicherung des Bildinhaltes bei einem **RGB Farbbild** der Größe **1.024x768**, wenn pro Farbkanal **4.096** verschiedene Werte kodiert werden sollen?
  - $1.024 \times 768 = \mathbf{786.432}$  Pixel
  - $4.096 = 2^{12} \Rightarrow \mathbf{12}$  Bit/Pixel
  - RGB = **3** Farbkanäle

$$\text{Size} = L \times N \times B \times c$$

$$\Rightarrow 786.432 \cdot 12 \cdot 3 = 28.311.552 \text{ Bit}$$

$$\Rightarrow 28.311.552 : 8 = 3.538.944 \text{ Byte}$$

$$\Rightarrow 3.538.944 : 1.024 = \mathbf{3.456 \text{ KiloByte (KB)}}$$



## Raster-Bildformate

### Raw-Bildformat

- **Raw-Bildformat** ermöglicht es, auf die tatsächlich von der Kamera aufgenommenen Bilddaten zuzugreifen
- Speichert für **jeden Pixel** den entsprechenden **Farbwert** ohne **Nachbearbeitung** (bei 1-Chip-Kameras werden Rot-, Grün- und Blauwerte entsprechend dem **CFA-Muster** gespeichert)
- **RAW-Format** stellt kein „richtiges“ Farbbild dar, da **2/3 der Farbinformation** interpoliert werden müssen

- Muss zur **farbigen Anzeige** umgewandelt werden

## Konventionelle Raster-Bildformate

- Beispiele für konventionelle Raster-Bildformate:
  - **Bitmap (BMP)**
  - **Portable Network Graphics (PNG)**
- Ein modernes Raster-Bildformat ist in der Lage, **zweidimensionale digitale Bilder** beliebiger **Breite**, **Höhe** und **Auflösung** abzuspeichern

## Struktur von Rasterbilddateien

- Eine Rasterbilddatei besteht aus **Strukturen fixer Größe (Header)** und **variabler Größe** (bildabhängig)
  - Die Strukturen erscheinen in einer vordefinierten Sequenz
    - Beispiel: **BMP**: Der **Bitmap File Header** speichert allgemeine Informationen über die Bitmap-Datei (14 Bytes)
  - **Metainformationen** werden in jedem individuellen Dateiformat gespeichert
- 

## Vektor-Bildformate

- **Vektor-Bildformat** beinhaltet eine **geometrische Beschreibung**, die problemlos für jede gewünschte **Anzeigegröße** gerendert werden kann
- **Rasterisierung**: An einem bestimmten Punkt müssen **alle Vektorgrafiken** rasterisiert werden, um auf einem digitalen Bildschirm angezeigt werden zu können
  - siehe: [5. Rasterisierung](#)

## Plotter und Vektordaten

- **Plotter** sind Drucker, die Vektordaten zum **Zeichnen von Grafiken** verwenden

## Computer Graphics Metafile (CGM)

- **CGM** ist ein freier und offener internationaler Standard für die Speicherung von **2D-Vektorzeichnungen**, **Rasterbildern** und **Text**
- Der Standard wird in Bereichen wie **technische Illustration**, **Kartografie**, **Visualisierung** und **elektronische Publikationen** verwendet
- Alle grafischen Elemente werden in **Quelltextdateien** spezifiziert, die anschließend zu einer **Binärdatei** oder **Textdarstellung** kompiliert werden
- **CGM** stellt Instrumente für den Austausch von Grafikdaten bei der Darstellung zweidimensionaler grafischer Informationen zur Verfügung, unabhängig von einer bestimmten **Anwendung**, **Plattform**, **System** oder **Gerät**

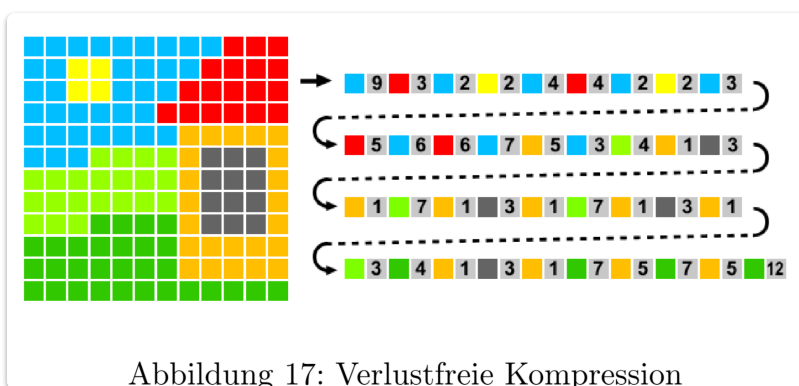
## Windows-Metafile (WMF)

- **WMF** wurde 1990 entwickelt
- **WMF** ermöglicht den **Datenaustausch** zwischen Anwendungen und beinhaltet sowohl **Vektorgrafiken** als auch **Bitmap-Komponenten**

## Bildkompression

- **Ziel der Bildkompression:** Reduzierung irrelevanter und redundanter Bildinformationen, um die Daten effizient zu speichern oder zu übertragen.
- **Arten der Kompression:**
  - **Verlustfrei (lossless):** Keine Daten gehen verloren, Bildqualität bleibt erhalten. Wird oft für medizinische Bilder, technische Zeichnungen oder Comics verwendet.
  - **Verlustbehaftet (lossy):** Daten gehen verloren, jedoch oft unmerklich. Wird für natürliche Bilder wie Fotografien verwendet, da es die Dateigröße stark reduziert.
- **Verlustbehaftete Kompression:**
  - Produziert **kompressionsartefakte** bei niedriger Bitrate.
  - In vielen Fällen als **visuell verlustfrei** bezeichnet, wenn der Verlust für den menschlichen Betrachter nicht wahrnehmbar ist.

## Verlustfreie Datenkompression



- Erlaubt eine exakte Rekonstruktion der Originaldaten. Wird in Bereichen genutzt, in denen die vollständige Datenintegrität wichtig ist (z. B. bei medizinischen Daten oder technischen Zeichnungen).

## Prozess der Kompression

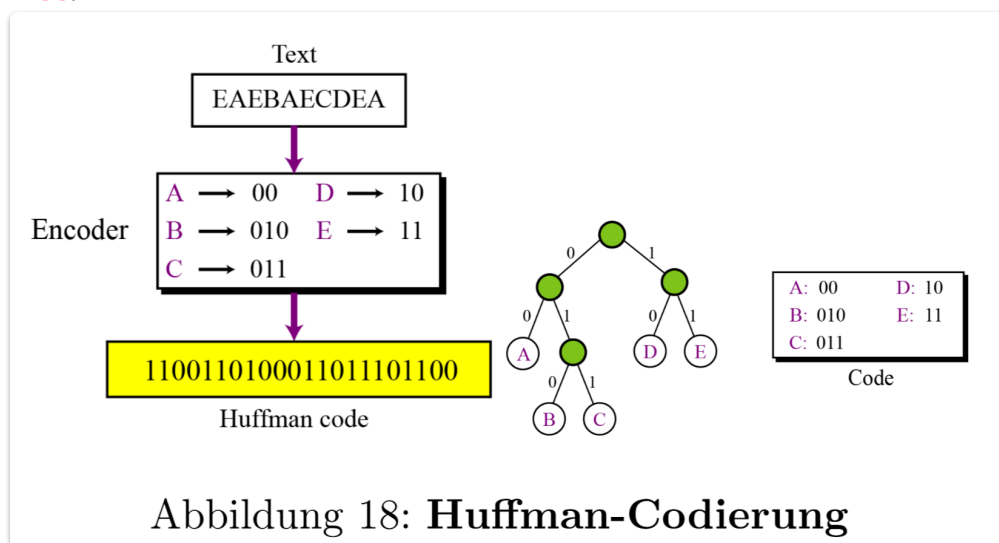
1. **Erstellung eines statistischen Modells** der Eingabedaten.
2. **Abbildung der Daten auf eine Bitreihe**, wobei häufig vorkommende Daten kürzere Bitfolgen erzeugen als seltene.

## Run Length Encoding (RLE)

- Ein grundlegender Kompressionsalgorithmus, der **Datenwiederholungen** speichert.
- Beispiel: Eine lange Sequenz gleicher Farben (z. B. bei Icons, Linienzeichnungen).
- **Nachteil:** Für natürliche Bilder, die keine langen Wiederholungssequenzen haben, kann es zu einer **Vergrößerung der Dateigröße** führen.
- **Entropiecodierung:** Sie erstellt **kurze Codes für häufige Symbole** und **lange Codes für seltene Symbole**. Dies reduziert die durchschnittliche Länge der Codes und verbessert die Kompression.

## Huffman-Codierung

- **Binärbaum:** Wird erstellt, bei dem die **Blätter** die Symbole und deren **Wahrscheinlichkeit (Häufigkeit)** enthalten. Die **Knoten** sind entweder Blätter oder interne Knoten.
- **Codeerstellung:**
  1. Beginnt mit den **Blättern**, die die Häufigkeit jedes Symbols enthalten.
  2. Zwei Knoten/Blätter mit den **geringsten Wahrscheinlichkeiten** werden zu einem neuen Knoten zusammengeführt. Der neue Knoten erhält eine Wahrscheinlichkeit, die der Summe der beiden Kinder entspricht.
  3. Der Vorgang wird wiederholt, bis nur noch ein Knoten übrig bleibt – der **Huffman Tree**.



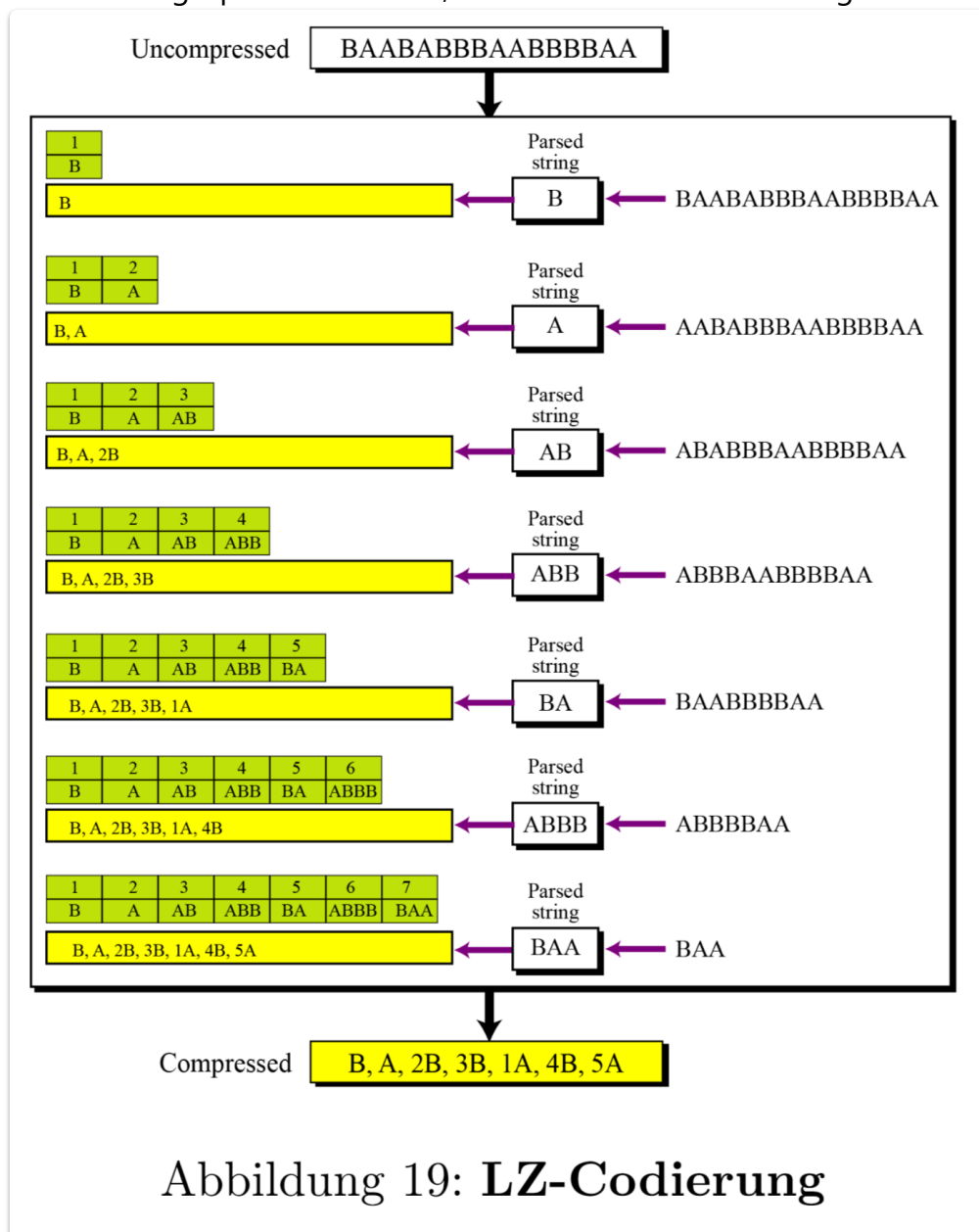
- **Ziel:** Der **Huffman Tree** ermöglicht die effiziente Zuordnung von **binären Codes** zu Symbolen, wobei häufige Symbole kürzere Codes erhalten und seltene Symbole längere.

## Lempel-Ziv (LZ) Kompressionsverfahren

Das **Lempel-Ziv (LZ) Kompressionsverfahren** basiert auf der **Wiederholung** von Daten und speichert diese Wiederholungen als Referenzen in einer **Tabelle**.

- **Codierung:**
  - **8-Bit Datensequenzen** werden als **12-Bit Code** komprimiert.
  - **Codes 0-255** repräsentieren **einzelne Zeichen** (1-Zeichen-Sequenzen).

- **Codes 256-4059** repräsentieren **Sequenzen**, die in einer Tabelle gespeichert sind.
- **Kompressionsprozess:**
  1. **Tabelle initialisieren:** Alle **1-Zeichen-Strings** werden zu Beginn als Einträge in der Tabelle gespeichert.
  2. **Längster String finden:** Der längste String **W**, der mit den aktuellen Eingabedaten übereinstimmt, wird identifiziert.
  3. **Tabellenindex ausgeben:** Der **Index für W** wird ausgegeben, und der String W wird vom Input entfernt.
  4. **Neuen String hinzufügen:** Der **neue String W + das nächste Symbol** wird zur Tabelle hinzugefügt.
  5. **Wiederholen:** Der Prozess geht weiter, bis das gesamte Eingabedaten verarbeitet sind.
- **Ziel:** Der Algorithmus **reduziert die Daten**, indem häufig wiederholte Sequenzen als Referenzen gespeichert werden, anstatt sie mehrfach abzulegen.



Bildformate die Verlustfreie Kompression verwenden:

### GIF (Graphics Interchange Format):

- **Einführung:** 1987 von CompuServe.
- **Farbe:** Unterstützt bis zu **256 Farben** (8 Bit pro Pixel) aus dem 24-Bit RGB Farbraum.
- **Verwendung:** Besonders geeignet für **kleine Bilder** (z. B. Icons) und **Animationen**.
- **Einschränkungen:** Weniger geeignet für **Fotografien** aufgrund der begrenzten Farbpalette.

### PNG (Portable Network Graphics):

- **Entwicklung:** Entwickelt als Verbesserung und Ersatz für GIF.
- **Farbe:** Unterstützt **24-Bit RGB** oder **32-Bit RGBA** (mit Transparenz) sowie **Graustufen**.
- **Kompression:** Verlustfreie Kompression mittels **PKZIP**.
- **Verwendung:** Besonders für den **Webbereich** geeignet, aber nicht für **hochqualitative Druckgrafiken**.
- **Farbräume:** Unterstützt nur **RGB**, keine CMYK.

### TIFF (Tagged Image File Format):

- **Verwendung:** Beliebt bei **Grafikern, Fotografen, Verlagen** und **Wissenschaftlern**.
- **Unterstützung:** Kann **Graustufenbilder**, **Indexbilder** und **Vollfarbenbilder** speichern.
- **Besonderheit:** Kann mehrere **Bilder** mit unterschiedlichen Eigenschaften in einer Datei speichern.
- **Kompression:** Unterstützt **verschiedene Kompressionsverfahren** (z. B. **LZW**, **ZIP**, **JPEG**, **CCITT**).
- **Verwendung:** Wird häufig für **Dokumentenarchivierung**, **wissenschaftliche Anwendungen** und in der **Digitalfotografie** verwendet.

## Verlustbehaftete Datenkompression

- **Prinzip:** Entfernungen von Datenteilen, die für das menschliche Wahrnehmungssystem nicht oder nur schwer erkennbar sind.
- **Ziel:** Die Daten werden so komprimiert, dass möglichst wenig Qualität verloren geht, aber die Dateigröße erheblich reduziert wird.
- **Transformation:** Daten werden in eine neue Domäne umgewandelt, die die relevanten Informationen effizienter darstellt.

### JPEG-Kompression:

- **Entwickelt:** Von der **Joint Photographic Experts Group (JPEG)**, 1990 als **ISO-Standard** etabliert.
- **Ziel:** Durchschnittliche Datenreduktion von **1:16**, ideal für **fotografische Bilder**.

## Codierungsprozess in JPEG:

### 1. Farbraumkonversion und Downsampling:

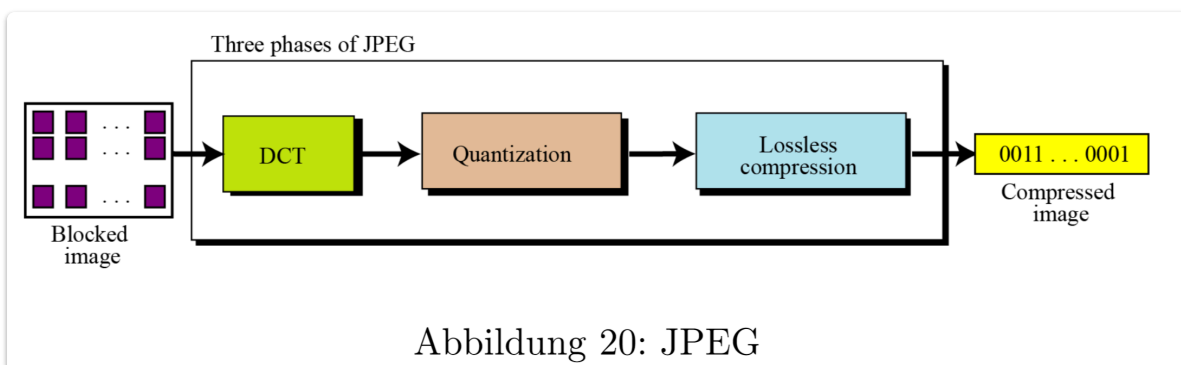
- Das Bild wird von **RGB** (Rot, Grün, Blau) in den **YCbCr-Farbraum** umgewandelt, wobei Y die Helligkeit (Luminanz) und Cb sowie Cr die Farbinformationen (Chrominanz) darstellen.
- **Downsampling** der Chrominanzkanäle (Cb und Cr) erfolgt, da das menschliche Auge weniger empfindlich auf Farbdetails als auf Helligkeitsdetails reagiert.

### 2. Kosinustransformation und Quantisierung:

- Das Bild wird in **8x8 Blöcke** unterteilt, und für jeden Block wird eine **diskrete Kosinustransformation (DCT)** durchgeführt, um die Frequenzen des Bildes zu berechnen.
- Die resultierenden **Spektralkoeffizienten** werden **quantisiert**, wobei hohe Frequenzen stärker reduziert werden, da diese weniger zur Wahrnehmung der Bildschärfe beitragen.

### 3. Verlustfreie Kompression:

- Nach der Quantisierung wird der Datenstrom mittels **verlustfreier Kompression** (z.B. **Laufängenkodierung** oder **Huffman-Kodierung**) weiter komprimiert, um die Dateigröße zu minimieren, ohne zusätzliche Informationen zu verlieren.



Das JPEG-Verfahren nutzt also **wahrnehmungspsychologische Erkenntnisse** zur Reduktion von Bilddaten und ermöglicht eine hohe Kompressionsrate bei gleichzeitig guter Bildqualität.

## Diskrete Cosinus Transformation (DCT)

### DCT (Diskrete Kosinustransformation):

- Die DCT ist eine Variante der **Fouriertransformation**, die Signale (hier Bildpixel) in **Cosinuswellen** unterschiedlicher **Frequenz** und **Amplitude** zerlegt.
- Ziel: **Frequenz- und Amplitudenverteilung** der Bildpixel sichtbar zu machen, um zu verstehen, welche Bildteile hohe oder niedrige Frequenzen enthalten.



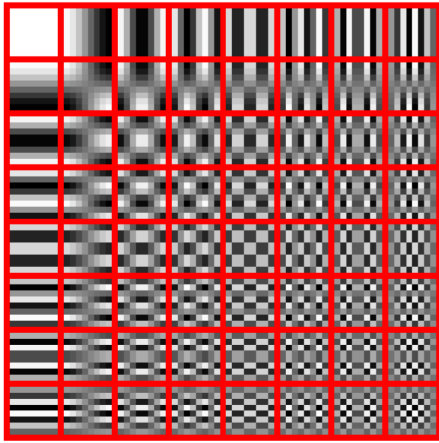


Abbildung 21: Die DCT ist eine Variation der Fouriertransformation.

- **Formel für die 2D-DCT (für einen 8x8 Block):**

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

- $u$  und  $v$ : **Horizontale** bzw. **vertikale Ortsfrequenz** ( $0 \leq u, v < 8$ ).
- $f(x, y)$ : Pixelwert am Punkt  $(x, y)$ .
- $F(u, v)$ : DCT-Koeffizient, der das Signal in Frequenzkomponenten zerlegt.
- $\alpha(u)$  und  $\alpha(v)$ : **Skalierungsfaktoren** zur Wahrung der Orthonormalität.

### JPEG-Kompression:

- **Quantisierung:** Die zentrale Methode der **verlustbehafteten** Kompression bei JPEG. Jeder DCT-Koeffizient wird durch einen vordefinierten Quantisierungswert geteilt und auf den nächsten Integer gerundet.
  - **Häufige Quantisierungswerte:** Niedrige Frequenzen erhalten **kleinere Quantisierungswerte** (präziser), während **hohe Frequenzen** größere Werte erhalten (weniger präzise), da das menschliche Auge weniger empfindlich gegenüber hohen Frequenzen ist.
  - Das führt dazu, dass **hochfrequente Komponenten auf 0** gerundet werden und **niedrigere Frequenzen eine hohe Genauigkeit** behalten.
- **Ergebnis:** Die Quantisierung reduziert die Bildgröße, indem sie **unnötige Bilddetails entfernt**, insbesondere bei den hohen Frequenzen.
  - **Visuell kann die Kompression das Bild auf ein Fünftel seiner Originalgröße reduzieren**, ohne dass große visuelle Qualitätseinbußen sichtbar werden.

### Kompressionsartefakte:

- Bei zu starker **Kompression** (zu hoher Quantisierung) können **Blockartefakte** auftreten, da das Bild in 8x8-Blöcke unterteilt wird und hohe Frequenzen unzureichend dargestellt

werden.

- **Schwächen:**
  - JPEG zeigt Schwächen bei **abrupten Übergängen** (wie Kanten oder Text).
  - **Blockbildung:** Bei sehr starker Kompression können die 8x8 Blöcke sichtbar werden, was das Bild unnatürlich erscheinen lässt.

### Optimierung für natürliche Bilder:

- JPEG wurde speziell für **natürliche fotografische Bilder** entwickelt und ist nicht ideal für **Computergrafiken** oder **Bilder mit scharfen Kanten**, bei denen die Blockbildung besonders auffällt.

## Video Kompression

- Ziel: **Reduzierung der Redundanz** in Videodaten, um die **Datenmenge** zu verringern und effizienter zu speichern oder zu übertragen.
- Kombination aus **räumlicher Bildkompression** (ähnlich wie bei Bildern) und **zeitlicher Bewegungskompensation** (bezieht sich auf die Bewegung zwischen den Frames).

### Techniken:

#### 1. Verlustbehaftete Kompression:

- Entfernt große Mengen an Daten, aber der visuelle Unterschied ist oft **kaum erkennbar**.
- Es gibt einen **Kompromiss** zwischen Videoqualität, Kompressionsaufwand und den Systemanforderungen.

#### 2. Räumliche Bildkompression:

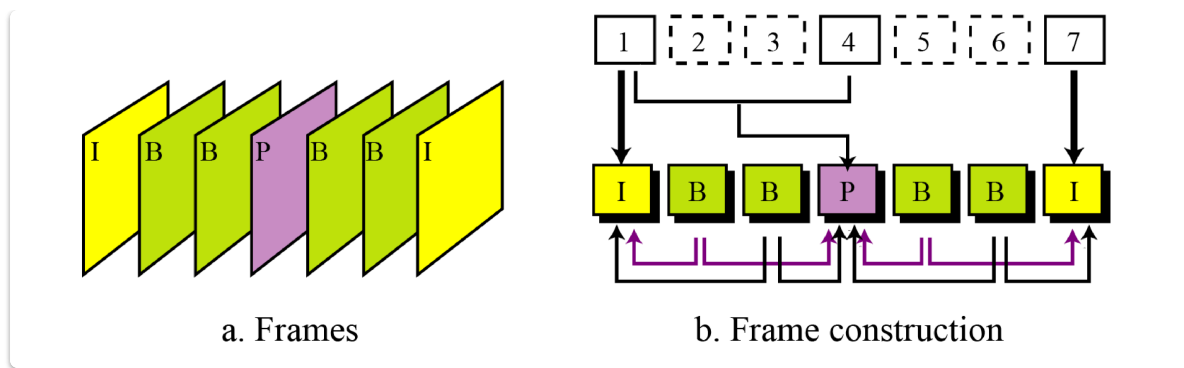
- Komprimiert das Bild innerhalb eines einzelnen Frames (ähnlich wie JPEG).

#### 3. Zeitliche Bewegungskompensation:

- Verwendet **Makroblöcke** (quadratische Bildausschnitte), die Unterschiede zwischen Frames messen.
- Bei viel Bewegung im Video müssen mehr Daten codiert werden, da mehr Pixel sich zwischen den Frames ändern.

### Interframe- und Intraframe-Kompression:

- **Interframe-Kompression:**
  - Verwendet **Frames davor und danach** (B/P-Frames) zur Kompression.
  - Beispiel: Ein Frame wird nur dann gespeichert, wenn sich etwas verändert hat, ansonsten wird er durch ein Referenzbild ersetzt.
- **Intraframe-Kompression:**
  - Komprimiert nur den aktuellen Frame (ähnlich wie Bildkompression, z.B. JPEG).



### Video-Kompressionstechniken:

1. **Veränderungen innerhalb eines Frames:** Wenn sich ganze Makroblöcke eines Frames verändern, kann der Kompressor Anweisungen wie **Verschieben, Rotieren oder Aufhellen** an den Dekompressor senden, um die Veränderung zu rekonstruieren.
2. **Interframe-Kompression:** Komprimiert Bereiche, die sich nicht verändert haben, durch einfachen **Verweis auf den vorherigen Frame**.

### MPEG-Kompression:

- **MPEG** (Moving Picture Experts Group) ist eine weit verbreitete Technik zur Video- und Audiokompression.
- **Asymmetrisch:** Codierung ist algorithmisch komplexer als Dekodierung (Vorteil im Broadcasting, da viele billige Dekodierer und wenige teure Codierer benötigt werden).

### MPEG-Standards:

1. **MPEG-1:**
  - Entwickelt für **Video CDs, SVCDs und DVDs** mit niedriger Videoqualität.
  - Ziel war es, Film und Ton auf die Bitrate einer **Compact Disc** zu kodieren.
2. **MPEG-2:**
  - Unterstützt **Zeilensprungverfahren** (Interlaced) und **High Definition** Auflösung.
  - Wichtig für **digitales Fernsehen, Kabelsignale** und **DVDs**.
3. **MPEG-4:**
  - Bietet **effizientere Codierung** und eignet sich auch für **Computergrafik-Applikationen**.
  - Wird neben MPEG-2 auch für **Blu-ray Discs** verwendet.

