

# MuMeTech-CheatSheet

## Definition

### MuMeTech

ist rechnergeführt, unabhängig, diskret und kontinuierlich.

### Kompression

Daten/Datenkanäle werden auf bestimmte Auflösung/Genauigkeit/Abtastrate reduziert (Bei unterschiedlicher Reduzierung je nach Kanal, nennt man es Subsampling).

### Übertragungsmodi

- **synchron** Der Sender sendet direkt an den Empfänger, es kann erst weitergesendet werden, wenn die Daten empfangen werden. (Handy)
- **asynchron** Die Daten werden während der Übertragung zwischengepuffert, womit der Sender nicht auf den Empfänger warten muss. (Post,Email)
- **isochron** Zeitraster ist fest, konstante Periode und Datenrate. (USB)

### Medienarten

- **Perzeptionsm.** Wahrnehmung
- **Repräsentationsm.** Darstellung
- **Präsentationsmedium** Ausgabe
- **Speichermedium** Physikalischer Datenspeicher

## Kompressionsarten

### Huffmann

Zeichen werden nach ihrer Häufigkeit geordnet mit verschiedenen langen Codes repräsentiert.

### Laufängenkodierung

Fasst direkt aufeinanderfolgende Zeichenketten zusammen. (*aaabbb*  $\Rightarrow$  *3a3b*)

## Erdrüsse

### FFT

Spaltet komplexes Signal in mehrere reine Sinusschwingungen auf, welche addiert das Originalsignal ergeben.

### DCT

Wichtige Elemente der darzustellenden Daten werden mit mehr Bandbreite versehen (Links-Oben-Bild)

## Audio

### Begriffe

- **Phon** Empfundene Lautstärke im Verhältnis zu 1000 Hz Sinus. Skaliert normal (nicht log.).
- **Dezibel** Logarithmisch ausgedrückte Lautstärke 6dB Unterschied bedeuten Verdoppelung der Lautstärke.
- **Frequenzamplitude** Amplitude wird angegeben in Dezibel und bestimmt die Lautstärke. Beschreibt maximale Auslenkung der Sinuswelle.
- **Klang** Schallwelle die vom menschlichen Ohr als bestimmter Ton wahrgenommen wird.

### Analog2Digital

1. **Vorverarbeitung** Filterung (Störgeräusche), Verstärkung (Dynamikausnutzung)  
Im zweiten Schritt erfolgt eine Frequenzbandbegrenzung (Tiefpassfilter) auf 1/2 der Abtastfrequenz (Shannon Abtasttheorem)
2. **Abtastung** In konstanten Intervallen wird der Wert des Eingangssignals entnommen.
3. **Quantisierung** Diskretisierung des bei der Abtastung ermittelten Wertes
4. **Kodierung** Binärkodierung der Signalproben

### Zusammenfassend

Aus einem zeitkontinuierlich ablaufendem Vorgang werden Signalproben genommen und in ihrer Amplitude quantisiert und in eine computergerechte Darstellung gebracht.

### Kodierungsmethoden

#### Verlustbehaftet

- **PulseCodeModulation - PCM** 3 Schritte
  - **Schritt 1** Abtastung mit zeitlich konst. Rate
  - **Schritt 2** Quantisierung der Werte
  - **Schritt 3** Kodierung in binärcode  
Die Kodierung erfolgt linear.

- **DPCM** Differenzielle PCM  
Die Quantisierung erfolgt anhand der Differenz zu einer Vorhersage.
- **DeltaModulation** Eine DPCM mit nur einem Bit. Wertebereich  $-/+1$ . Die Schätzwerte nehmen dabei immer an, dass der neue Abtastwert gleich dem vorherigem ist.
- **Adaptive differenzielle PCM** Ähnlich DPCM, jedoch mit dynamischer Vorhersage. Angepasste Quantisierung, dadurch bessere Quali.

### Kompressionsverfahren für Audio

#### Datenreduktion

Filterung der Daten (z.B. nach psychoakustik).

### Datenkompression

Verlustfreie Komprimierung der Daten.

### mp3

1. PCM (768Kbit/s)
2. Filterbank für 32 Subbänder / FastFourierTrans für 1024 Abtastwerte
3. FFT  $\Rightarrow$  PsychAkModel nun wird anhand derer und der Subbänder quantisiert.
4. Audiodatenkodierung mit Huffmann, Nebeninfos codiert
5. BitstromFormatierung und Fehlerkorrektur

### MIDI

Datenübertragungsprotokoll für Musikdaten.Übertragen werden Steuerinformationen zwischen elektronischen Instrumenten, welche von Programm interpretiert werden können.  
Inhalt zum Beispiel: Anschlagstärke, Lautstärke, MidiKanalnummer (4Bit), Spurname

Format 0 Alle Midikanäle sind in einer Spur zusammengefasst, somit keine gleichzeitigen Anschläge verschiedener Instrumente (Klingelton)

Format 1 Jeder Kanal hat eigene Spur, somit können auch gleichzeitige Anschläge realisiert werden.

Format 2 Im Format 2 besteht jede Spur (Track) aus unabhängigen Einheiten. Im Gegensatz zu SMF 1 können also mehrere Spuren dieselbe MIDI-Kanal-Nummer haben.

**THRU-Port** gibt parallel zum Outport eines Gerätes (z.B. Filter) das unbehandelte Inputsignal aus (z.B. für Aufnahmen).

### Beispielrechnungen

#### Einheiten:

$1MB \Rightarrow 10^6 Byte$  ||  $1MiB \Rightarrow 1024 \times 1024 Byte$

$1GB \Rightarrow 10^9 Byte$  ||  $1GiB \Rightarrow 1024 \times 1024 \times 1024 Byte$

**PCM: 44.1KHz,16Bit, stereo, 20min**

$44.100 \times 16 \times 2 \times 20 \times 60 \Rightarrow Bit$

$44.100 \times 2 \times 2 \times 20 \times 60 \Rightarrow 2116800 Byte \approx 2.1MB$

**MP3: 128kbit/s**

### Grafiken/Bilder

#### Farbmodi

- **RGB** RotGrünBlau.  
Additive Farbmischung mit drei Farbkanäle a 8Bit (default).
  - **Anwendungen** Monitordarstellung, Kamera
  - **Vorteile**  
Gut auf Geräten anzuwenden, die Lichtquellen aussenden.  
Direkt mit Algo bearbeitbar  
Darstellungskapazität vieler Farbnuancen

- **Nachteil**  
Probleme mit Darstellung von Schwarz  
Geräteabhängig.  
8 % des Farbraums sind nicht wahrnehmbare  
Farben  
Helligkeitskorrektur schwer  
Eignet sich nicht für Druck (Additiv/Substraktiv)
- **YUV** Darstellung durch Luminanz (Y) und Chrominanz (UV).
  - **Anwendungen** Analoges  
NTSC/PAL-Farbfernsehen
  - **Vorteile**  
Halbe Bandbreite von RGB  
Durch Subsampling optimierung möglich (siehe  
Subsampling)  
Vollständiger Farbraum abgedeckt  
Abwärtskompatibel zu Schwarz/Weiss  
Ausnutzung Wahrnehmungspsychologie  
Helligkeit separat im Gegensatz zu RGB (jeder  
Kanal muss angepasst werden)  
Progressive Vollbilder möglich
  - **Nachteil**  
Verteilung der Farbanteile der Cyan/Orange und  
Magenta/Grün ist ungleichmässig auf U und V,  
daher keine Bandbreitenreduktion möglich
- **YIQ** Darstellung durch Luminanz (Y), sowie den  
Farbdifferenzen I (Cyan/Orange) und Q (Magenta/Grün)  
Irgendwie zu YUV verdreht! WHY? How much?
  - **Anwendungen** Altes analoges  
NTSC-Farbfernsehen
  - **Vorteile**  
Ähnlich YUV  
Kommt wahrscheinlich nicht in der Klausur dran  
(Jonas)
  - **Nachteil**  
Nur überm Teich im Gebrauch

## JPEG

### JPEG-Kodierung

- **Bildvorverarbeitung (verlustfrei)**
  - Grauwerttransformation (Kontrasterhöhung und  
Helligkeitsverbesserung)
  - Bildfilterung (Rauschunter-  
drückung,Kantenverstärkung,Glättung,...)
- **Bildverarbeitung (theo. verlustfrei)**
  - Abtastung und Digitalisierung der  
Bildinformationen.
  - Einteilung in 8x8-Pixel-Blöcke, wobei jeder Pixel  
mit 8bit kodiert wird (optimaler Kompromis  
zwischen Laufzeit und Quali; Zahl für DCT).

- DCT Der 8Bit-Farbwert wird vom Ortsbereich- in  
den Frequenzbereich transformiert.  
Das Ergebnis ist eine 8x8-Frequenzraummatrix  $S$ ,  
 $S_{00}$  entspricht dem Anteil der Frequenz 0  
(Grundfarbton), dieser ist der DC-Koeffizient.  
Alle anderen  $S_{ij}$  heissen AC-Koeffizienten und  
geben Auskunft über die Frequenzveränderungen  
(Farbver.) innerhalb des Blockes.  
Der letzte Eintrag  $S_{77}$  gibt dabei die höchste in  
beiden Richtungen auftretene Frequenz an.

- **Quantisierung (verlustbehaftet)** Erstellen einer  
ZickZack-Sequenz (Diagonaler Schnitt von links-oben  
an). Ausnutzung des PsychoVisuellenModells (PVM). Die  
Anwendung stellt eine Liste mit 64 Faktoren zur  
Verfügung. Anhand dieser werden die DCT-Koeffizienten  
gewichtet (und gerundet), wodurch die Frequenzwechsel  
an die Qualitätsanforderung angepasst werden.  
Wird die Qualität reduziert, so ist die rechte untere  
Dreiecksmatrix mit Nullen versehen. Dies kommt der  
Entropiekodierung zu Gute.
- **Entropiekodierung (verlustfrei)** Die resultierende  
Liste wird mit Huffman oder arithmetisch kodiert.

### JPEG-Modi

- **Sequenzielle mode** Das Bild wird in einem einzigen  
Durchlauf kodiert.
- **progressive mode** Das Bild wird in mehreren  
Durchläufen immer genauer kodiert.  
Vorteil: Schnelle (grobpixelige) Vorschau des Bildes
- **Hirachischer Modus** Das Bild wird in verschiedenen  
Auflösungen kodiert.  
Vorteil: Jede Anwendung greift sich ihre geeignete  
Auflösung heraus und muss nicht rekodieren.
- **lossless mode** Verlustfreie Kodierung des Bildes

## Netzwerk/Internet

### IP-Adresse/Subnetzmaske

IP-Adresse wird zur genauen Identifikation eines Host genutzt  
und besteht aus 32Bit.  
Sie wird in ClassA-D eingeteilt. Hierzu wird die IP mit der  
Subnetzmaske AND-Verknüpft, das Ergebnis ist die NetzID.  
Eine ClassX-Subnetzmaske besteht aus führenden Einsen,  
gefolgt von Nullen. Abkürzend wird nur die Anzahl der Einsen  
angegeben:

#### 192.168.1.23/24 (SM: 255.255.255.0)

11000000.10101000.00000001.00010111 (IP-Adresse)  
11111111.11111111.11111111.00000000 (Subnetzmaske)  
11000000.10101000.00000001.00000000  $\Rightarrow$  (NetzwerkID)  
00000000.00000000.00000000.00010111  $\Rightarrow$  (HostID)

### Subnetting

Um die z.T. riesigen Netze logisch zu unterteilen, kann von der  
ClassA-D Begrenzung abgewichen werden,  
daraus ergeben sich dann kleine Teilnetze.

- **Vorteil**
  - Durch kleine Subnetze wirken sich z.B. Broadcasts  
nicht auf das Riesenetz aus.
  - Hostgruppen können getrennt werden.
- **Nachteil** Kompromiss:  
'grosses Netz mit Broadcast-Probleme' vs 'kleine Netze  
die mit Router verbunden werden müssen'

		Adressen
123.45.64.0/18	- Provider	16384
123.45.64.0/20	- Kunde A	4096
123.45.64.0/28	- A1	16
123.45.64.16/28	- A2	16
123.45.80.0/20	- Kunde B	4096
123.45.96.0/19	- Kunde C	8192

### Übertragungsarten

- **unicast (1:1)** Normale Netzwerkverbindung z.B.  
Client/Server (Mail, http)
- **broadcast (1:n)** Ein Host kommuniziert mit allen  
Knoten im Netzwerk (Subnetz), zB. um einen  
DHCP-Server zu finden
- **multicast (1:m)** Ein Host sendet Daten an mehrere  
Empfänger (z.B. Videostream), die Bandbreite erhöht  
sich nicht mit der Anzahl der Empfänger.
- **multipeer (m:m)** Mehrere Hosts senden und  
empfangen gleichberechtigt in einer Hostgruppe. (z.B.  
Videokonferenz)

## AJAX

Dient dem Asynchronen Datenaustausch um Inhalte dynamisch nachzuladen. Dabei muss stets nur das Frame der Webseite nachgeladen werden, welches verändert wird.

- **Voraussetzungen** JavaScript, XML-HTTP-Request. Zur Darstellung wird HTML, JavaScript und DOM genutzt.
- **Vorteile**
  - Daten können verändert werden, ohne die Seite komplett laden zu müssen.
  - Webanwendungen können schneller auf Benutzereingaben reagieren.
  - Kein unnötiges Nachladen von statischen Inhalten
- **Nachteil**
  - Abhängigkeit von JavaScript (5% der User haben es nicht aktiviert)
  - Vor-/Zurückbutton des Browsers nicht mehr funktionstüchtig, da Browser i.d.R. nur die statischen Daten speichern
  - Lesezeichen setzen, etc. nicht mehr möglich
  - Testing der Anwendung aufwendig
  - Die Latenzzeit des Server kann nachteilig wirken (hohe Serverlast → Unzufriedenheit des Users)

## HTML5

Liste von neuen Tags:

- **audio** Definiert Audioinhalte
- **video** Definiert Videoinhalte
- **time** Datumsdefinitionen
- **article** Artikel
- **canvas** 2D-Bitmap-Zeichenfläche
- **details** Detailinformationen zu einem Element
- **section** Erleichtert das Abgrenzen von unterschiedlichen Inhalten, soll das div-Tag ersetzen

Weiter wurde der DOCTYPE in HTML5 auf einen HTML-Typen zusammengestrichen, HTML4 kennt noch drei Doctypes.  
Die Sprache befindet sich z.Zt. noch in der Entwicklung (MARCO).

## HTTP

Übermittlungsprotokoll für das WWW, erfunden 1995 von Tim Berners Lee am CERN in Genf.

```
TCP-Verbindungsaufbau (Handshake)
-> GET http://URL (Client fordert Inhalt an)
<- HTTP/1.X 200 OK (Response von Server)
```

## MPEG

### MPEG1

#### Subsection

Dient zur Speicherreduktion indem Chrominanz im Vergleich zur Luminanz mit veringierter Abtastrate gespeichert wird (PAVM), da Helligkeit- besser als Farbunterschiede wahrgenommen werden.

#### Subsampling

- 4:4:4 (MPEG1 verwendet)  
Unkomprimiert, Farb und Helligkeitsinformationen werden gleich häufig abgetastet.
- 4:2:2 Abtastrate der Farbkanaele in horizontaler Richtung halb so gross wie in vertikaler Richtung. Anwendung beim Analog-Farbsehen wegen dem Zeilensprungverfahren.
- 4:2:0 Wird bei digitalen Bildern wie JPEG und MPEG angewendet. Abtastung in beide Richtungen identisch.

#### Kodierung

- **1. Schritt:** Bildaufbereitung: Konversion der Farbraums (von 24 Bit RGB in Y,Cr,Cb), Anwendung der Kodierung
- **2. Schritt:** 4.2.2 Subsampling
- **3. Schritt:** Einteilung in Makro-Blocke
- **4. Schritt:** Bewegungsvorhersagealgorithmus: Vgl des aktuellen Makro-Blocks mit den umgebenden und den Makro-Blocken des vor- und nachfolgenden Einzelbilds. Ähnlich wie bei der DPCM wird nur die Differenzinformation gespeichert und der Vektor kodiert.
- **5. Schritt:** DCT
- **6. Schritt:** Bildverarbeitung: Kodierung (des Videostroms)
- **6.1. verschieden Arten von Einzelbildern (Frames):**
  - **6.1.1. I-Frame (Intracoded Picture):**  
Standbild JPEG-ähnlich ohne zusätzliche Infos. (Kompression gering, wie bei JPEG, jedoch in Echtzeit)
  - **6.1.2. P-Frame (Predictive Coded Picture):**  
Bezieht sich auf das vorhergehende I-Frame, enthält Vektoren und Bilddifferenzinformationen. (Grössere Kompr. als bei I)
  - **6.1.3. B-Frame (Bidirectional Coded Picture):** Vergleicht folgendes (I- oder P-Frame) mit vorangehendem (I- oder P-)Frame und stellt Durchschnittswert dar. (Grösste Kompression)
  - **6.1.4. D-Frame (Direct Coded Picture):** Bei geeigneter Speicherung der I-Frame überflüssig. Nur DC-Koeffizienten werden kodiert, daher schlechte Qualität.  
Nur ein Farbwert pro 8x8 Block wird gespeichert, Verwendung für schnellen Vorlauf.

- **7. Schritt:** Quantisierung: verlustbehaftet  
Für I-Frame gleich wie bei JPEG, bei P- und B-Frames werden nur die DCT-kodierten Anteile quantisiert.  
Bewegungsvektoren für P- und B-Frames werden hier allesdings verlustfrei gespeichert.
- **8. Schritt:** Entropiekodierung, Lauflängenkodierung, anschliessend Huffman.

### Vergleich MPEG1 - MPEG2

Verbesserungen bei MPEG2: Höhere Auflösung und Bitraten, Interlaced Videosequenzen, niedrigere Audioabtastraten möglich, Unterstützung verschiedener Qualitätsprofile, versch. Subsamplingformate.

### Unterschiede MPEG2 zu H.261

Vorteile MPEG2: Breitere Unterstützung, hohe Qualität bei guter Kompression, keine Echtzeit/Latenzanforderungen  
Vorteile H.261: Kontinuierliche Bitraten, daher für Videokonferenz oder ähnliche Echtzeitanwendungen geeignet, geringe Latenz.

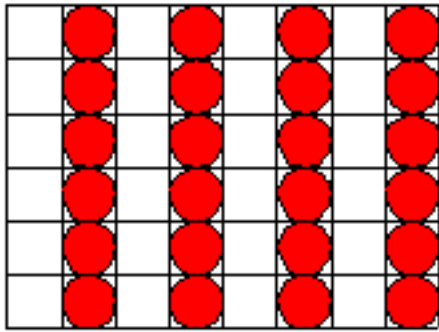
## MPEG4

### Beispielrechnung

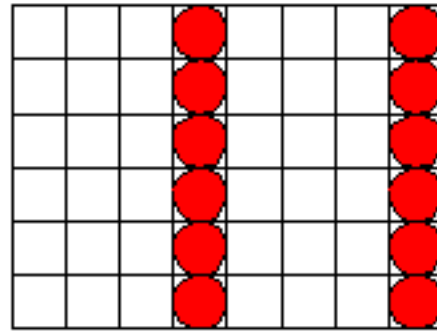
#### MPEG

Bei 800x600, 24Bit, 25fps, 60s sind es pro Minute:  
 $800 \times 600 \times 24 \times 25 \times 60 \Rightarrow 4320 \times 10^6 \text{ Byte}$

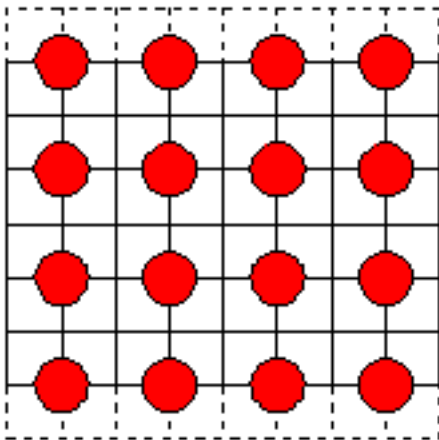
## CD/DVD



4:2:2 (ITU-R BT.601-4, D-1)



4:1:1 (DV-NTSC, DVCPRO)



4:2:0 (DV-PAL, MPEG2-ATSC)

□ Luma samples (Y), 13.5 MHz  
720 active samples/line

● Chroma samples (CrCb)  
13.50 MHz (4:2:2)  
6.75 MHz (4:1:1, 4:2:0)