Fachbereich Informatik

Lehrgebiet Multimedia und Internetanwendungen

Prof. Dr.-Ing. Matthias L. Hemmje
Dipl. Inf. Holger Brocks

# Audioformate

## Seminararbeit
## Multimediatechnologien
## SS 2007

Birgit Ianniello
2007

# Inhalt

## *Vorwort*

„Die Welt ist Klang – Nada Brahma“
Dies ist der Titel eines außergewöhnlichen Beitrags von Ernst Joachim Berendt über die Wichtigkeit von Klängen in unserer Welt und des Universums. Herr Berendt geht sogar so weit zu behaupten, dass die Entstehung unseres Universums ohne Klang nicht denkbar wäre. Vielleicht ist das ja wahr – Warum sonst ist der Nachweis des Urknalls für uns von so immenser Bedeutung?
Diese Seminararbeit handelt jedoch nicht von philosophischen Betrachtungen über Töne und Klänge, Harmonien und Dissonanzen, sondern vielmehr von technischen Grundlagen zur Erzeugung derselben. Der Schwerpunkt liegt dabei in der Digitalisierung und Komprimierung von Audiosignalen.
Ich hoffe, dass es mir trotz eher technischem Schwerpunkt gelungen ist das Thema Audio nicht minder interessant gestaltet zu haben als Herr Berendt und wünsche viel Vergnügen beim Durchlesen dieser Lektüre.

Birgit Ianniello

## *1.Einleitung*

Wir sind ständig von Schallwellen umgeben und nehmen diese auch mehr oder weniger bewusst wahr. Im Gegensatz zu den Augen, ist der Mensch nicht in der Lage seine Ohren zu verschließen und so die Dauerbeschallung einfach auszublenden. Das menschliche Hörspektrum reicht von 20 Hz bis 20 000 Hz und Schallwellen in eben diesen Frequenzen werden Töne genannt. Jedoch gibt es auch Schallwellen, die nicht durch die Ohren wahrgenommen werden und dennoch unseren Körper beeinflussen.
Ein Beispiel hierzu ist der sog. Infraschall mit Frequenzen so um 10 Hz. Werden wir diesen Frequenzen ausgesetzt, so können sie Unwohlsein und erhöhten Pulsschlag hervorrufen. Diesen Effekt macht sich auch die Unterhaltungsindustrie zu Nutze und so werden Filme mit Infraschallsoundtracks sehr intensiv empfunden [hen03].
Bei Schall handelt es sich also um Wellen, die sich mehr oder weniger stark verbreiten. Mithilfe von zum Beispiel Schallplatten, kann dieser Schall analog konserviert werden und ist im Prinzip jederzeit abspielbar. Jedoch ist die gleichmäßige Abspielgeschwindigkeit genauso wichtig, wie die Unterbrechungsfreie Wiedergabe – ansonsten empfinden wir die Klangabfolgen schnell als unangenehm. Wegen dieser Eigenschaften sind Tonträger zeitkontinuierliche bzw. zeitkritische Medien. Die Schallwellen sind zudem auch noch wertekontinuierlich, denn jede simple Sinuswelle besteht aus unendlich vielen Werten. Die Kunst der Digitalisierung besteht also darin, ein analog so nicht per Computer zu verarbeitende Audiosignal in computergerechte Häppchen zu zerlegen und dies möglichst so zu tun, dass beim anschließenden Abspielen der neu entstandenen Audiodatei zumindest der Wiedererkennungseffekt beim Hörer eintritt.

## *2. Digitalisierung*

Wie weiter oben erläutert, sind die Klänge, die von unserer Umgebung an unser Ohr gelangen analog und können so zunächst nicht im Computer weiter verarbeitet werden, denn dieser beruht ja auf einer digitalen Logik. So müssen die analogen Schallwellen digitalisiert werden. Dieser Vorgang verläuft nach dem *Drei-Stufen-Modell*: Abtasten, Diskretisieren und Kodieren der Schallwelle und wird im folgenden Abschnitt näher erläutert.

## 2.1 Sampling, Quantisierung und Kodierung

Beim *Sampling* und der *Quantisierung* findet eine Abtastung des zeit- und wertekontinuierlichen Signals in gewissen Intervallen statt. Die Einheit der *Samplingrate* wird mit *Hz* also *Hertz* angegeben. Alle Werte, die nicht zu den festgelegten Zeitpunkten erfasst worden sind, werden verworfen. Nach dem Sampling ist das Signal zwar in eine endliche Anzahl von Werten zerlegt, diese können aber immer noch unendlich sein. Deshalb folgt nun der zweite Schritt: die *Quantisierung*
Die für den Computer nicht verarbeitbaren Werte – wie etwa 1/3 oder Pi - werden auf den nächsten diskreten Wert gerundet. Welcher Wert der am nächsten liegende Wert ist, wird durch die *Quantisierungsintervalle* festgelegt. Was bei der Abtastung die Samplingrate ist, ist beim Diskretisieren die Bitrate. Durch die Quantisierung entstehen Ungenauigkeiten, deren Effekt auch als *Quantisierungsrauschen* bekannt ist. Je höher die gewählte Bitrate zur Quantisierung gewählt wird, desto geringer ist die Wahrscheinlichkeit, dass störendes Quantisierungsrauschen auftritt.
Der letzte Schritt des Drei-Stufen-Modells ist die *Kodierung*. Hierbei werden die verschiedenen Quantisierungsintervalle mit binären Codewörten gekennzeichnet. Das so entstandene digitalisierte Signal wird mitsamt dem *Quantisierungsfehler* übertragen, der jedoch bei geschickter Wahl der Bitrate nicht im Bereich des hörbaren liegt.

Hier der gesamte Vorgang im Überblick:

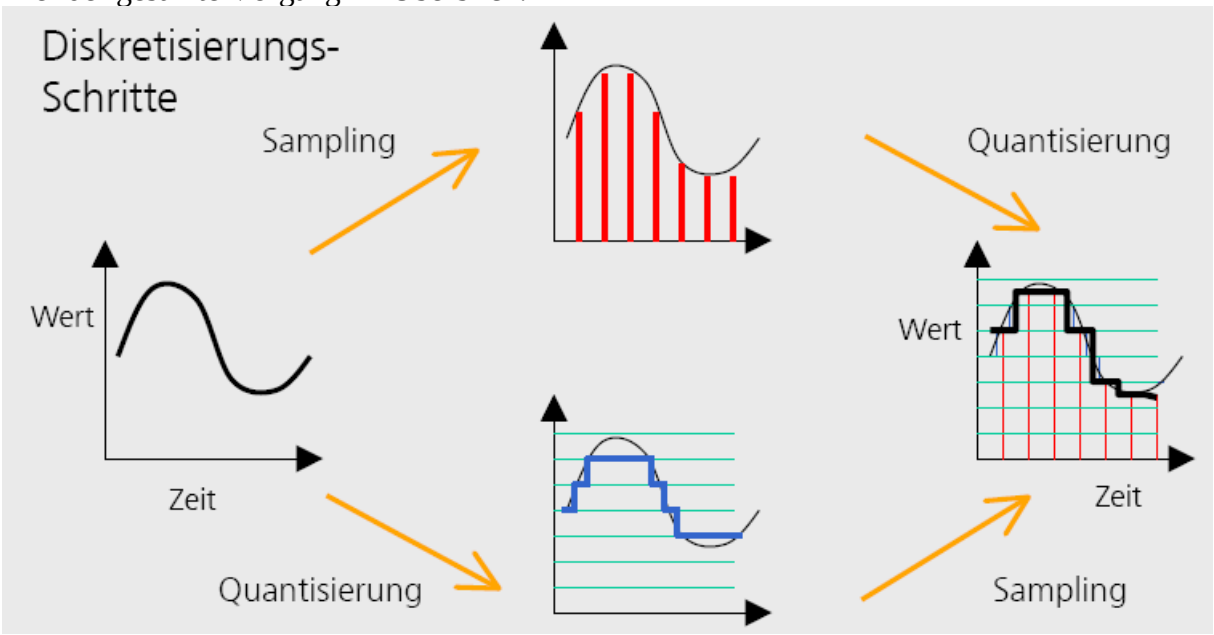


*Abb.2.1.1 Sampling und Quantisierung, Quelle [wat96]*

Das hier beschriebene Verfahren zur Analog / Digitalwandlung von Signalen wird auch *Waveform-Encoding*, bzw. *PCM* (**P**ulse **C**ode **M**odulation) genannt.

## 2.2 PCM

Es gibt unterschiedliche PCM-Verfahren. Welches der Verfahren zur Digitalisierung des Klanges ausgewählt werden sollte, hängt vom Anspruch an das Ergebnis ab. Sicher ist die lineare PCM bei hinreichend kleinen Quantisierungsintervallen sehr genau am Original, soll diese Datei jedoch anschließend zum Beispiel über das Internet übertragen werden, ist eine entsprechend hohe Bandbreite nötig. Die differenzielle PCM hingegen braucht eine ausreichend große Rechenleistung – dafür sind die Dateien kleiner. Für die Kodierung von Sprachdateien bieten sich die nicht-linearen Verfahren an, denn es bei starken Signalen reichen größere Quantisierungsintervalle wohingegen bei schwachen Signalen eine genauere Näherung gefragt ist. Es gilt also je nach Einsatzgebiet die Vorteile gegen die Nachteile abzuwägen. Die drei grundlegend verschiedenen Verfahren der PCM sind:

**Lineare PCM**
Die Quantisierungsintervalle sind alle gleich groß. Das so digitalisierte Signal hat eine gute Qualität – allerdings auf Kosten einer hohen Bitrate.

**Dynamische PCM**
Die Größe der Quantisierungsintervalle wird dynamisch angepasst, wobei zum Beispiel die Quantisierungsintervalle bei leisen Passagen kleiner gewählt werden und so das Quantisierungsrauschen geringer ausfällt. Die dynamische Anpassung kann mit Hilfe einer Logarithmischen Funktion erreicht werden.

**Differenzielle PCM**
Die Abweichungen zwischen den einzelnen abgetasteten Werten sind oft nur gering. Bei diesem Verfahren wird lediglich die Differenz gespeichert zwischen den Werten gespeichert. Optimale Ergebnisse werden hierbei mit *Predictive Coding* erzielt (siehe unten). Das Prinzip der **DPCM** ist noch weiter entwickelt worden zu **ADPCM**, also **A**daptive **D**ifferential **P**ulse **C**ode **M**odulation.

Wie aus Abbildung 3.2.1 hervorgeht ist das Herausfinden der optimalen Abtastrate von essentieller Bedeutung. Deshalb zum Abschluss dieses Kapitels noch ein kleiner mathematischer Exkurs:

## 2.3 Abtasttheorem

Ist die Abtastrate zu gering, kann das Signal nur fehlerhaft abgebildet werden. Dem entgegen steht der Wunsch, möglichst wenige Werte zu erhalten, damit die anschließende Verarbeitung nicht zu umfangreich gerät. Von der optimalen Abtastrate handelt das berühmte *Abtasttheorem* nach *Nyquist*, *Kotelnikow*, *Raabe* und *Shannon* welches ein grundlegendes Theorem der Nachrichtentechnik, Signalverarbeitung und Informationstheorie ist:

Claude Elwood Shannon formulierte das Theorem 1948 als Ausgangspunkt seiner Theorie der maximalen Kanalkapazität, d.h. der maximalen Bitrate in einem frequenzbeschränkten, rauschbelasteten Übertragungskanal. Dabei stützte er sich auf Überlegungen von Harry Nyquist (1928) zur Übertragung endlicher Zahlenfolgen mittels trigonometrischer Polynome und auf die Theorie der Kardinalfunktionen von Edmund Taylor Whittaker (1915) und seinem Sohn John Macnaughten Whittaker (1929) [2]. Unabhängig davon wurde das Abtasttheorem 1933 von Wladimir Alexandrowitsch Kotelnikow [3] in der sowjetischen Literatur eingeführt, was im Westen allerdings erst in den 1950er Jahren bekannt wurde. Ansätze zur Interpolation mittels Kardinalreihen oder ähnlicher Formeln lassen sich bis in die Mitte des 19. Jahrhunderts zurückverfolgen. *Quelle: Wikipedia*

Eine Signalfunktion, die nur Frequenzen in einem beschränkten Frequenzband enthält, wobei *f*max gleichzeitig die höchste auftretende Signalfrequenz ist, wird durch ihren diskreten Amplitudenwert im Zeitabstand $T_0 \leq \frac{1}{2 \cdot f_{\max}}$ vollständig bestimmt.

Aus diesem Theorem lässt sich folgern, dass ein Signal durch eine Abtastfrequenz die doppelt so hoch ist wie die höchste im Signal vorkommende Frequenz vollständig bestimmt werden kann. Die höchste Frequenz sei *f*max und wir erhalten $f_A \geq 2 \cdot f_{\max}$

Da der Mensch Frequenzen hören kann die von etwa 20Hz bis 22 kHz liegen, reicht dem Abtasttheorem zufolge bei *f*max := 22 kHz eine Samplingrate >= 44 kHz vollkommen aus um analoge Audiosignale optimal abzutasten.

## 3. Komprimierung von Audiodaten

Nach der Digitalisierung der Audioinformationen liegen diese zwar nun als abspielbare Dateien vor, nehmen jedoch ziemlich viel Platz ein. Eine Minute Hörerlebnis benötigt in dieser Form immerhin schon rund 10 MB. Um also ein Album mit ungefähr einer Stunde Laufzeit abzuspeichern, sind etwa 600 MB nötig – soviel wie bequem auf eine CD passt. Dies ist übrigens kein Zufall, sondern die Größe der CD wurde so dimensioniert, dass genau Beethovens Neunte dirigiert von Karajan auf einer CD-ROM abgespeichert werden kann. Zur Abspeicherung auf Festplatten, für die Übertragung übers Internet oder gar für mobile Player ist diese Datenmenge viel zu groß und so wurden die verschiedenen Ansätze zur Audiokomprimierung verfolgt und weiterentwickelt.

### 3.1 Huffman Codierung

Die Huffman Codierung [huf52] gehört zum *Entropy Coding* Verfahren genannt beruht auf der Idee, dass der benötigten Platz um eine bestimmte Anzahl von Zeichen im Binärcode darzustellen erheblich verringert werden kann, wenn die Zeichen, die häufig vorkommen nur wenig Stellen als Binärzahl beanspruchen. Zu diesem Zweck wird also ein Binärbaum angelegt, bei dem die längsten Pfade zu den am seltensten vorkommenden Knoten führen. Bei jedem Knoten, den der Pfad berührt wird der Binärzahl entweder eine 1 oder 0 hinzugefügt – abhängig von der Richtung in die der Pfad anschließend weiterführt.
Beispiel:
Der Satz „Barbara mag Rhabarber“ braucht bei 8-Bit ASCII Kodierung 168 Bits Speicherplatz. Um nach Huffman diesen Platz zu verringern, wird zunächst die Anzahl der Vorkommen für jedes Zeichen gezählt und als Gewichtung gemerkt:

| B a r b  m g R h e |
|---|
| 1 6 4 3 2 1 1 1  1 1 |

Nun wird jedes Zeichen gemeinsam mit seiner Gewichtung als Knoten dargestellt:

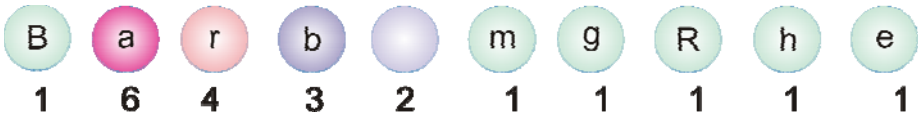


*Abb:3.1.1 Knotengewichtung*

Jetzt werden jeweils die Knoten mit der geringsten Gewichtung paarweise zusammengefasst und erhalten einen Elternknoten, der die Summe der Gewichtung der Kinder als eigene Gewichtung bekommt. Dies wird so lange gemacht, bis ein Binär-Baum entstanden ist:

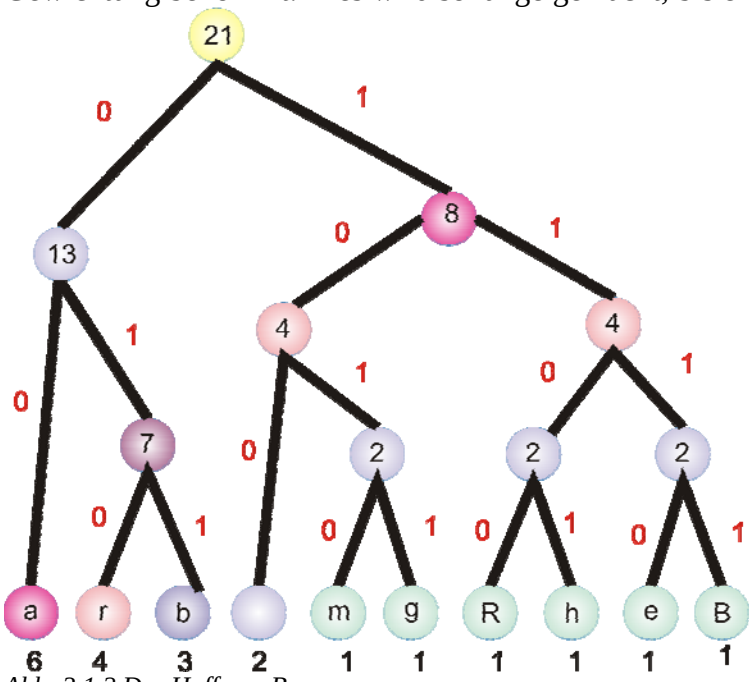


*Abb: 3.1.2 Der Huffman Baum*

Die neuen Binärcodes für die Zeichen werden nun gefunden, indem von der Wurzel – hier 21 – bis hinunter zum jeweiligen Buchstaben der Pfad verfolgt wird. Die jeweiligen Binärziffern werden entsprechend notiert. Da zum Beispiel nach „a“ zwei linke Kanten führen ist die neue Kodierung für „a“ gleich „00“. Genauso wird zur Kodierung mit den restlichen Zeichen verfahren und es entsteht folgende Codetabelle:

| a | r | b | | m | g | R | h | e | B |
|---|---|---|---|---|---|---|---|---|---|
| 00 | 010 | 011 | 100 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 |

Der so entstandene Code ist:
„11110001001100010001001010001011100110011010001100010011111 0010“
und es werden nur noch 63 Bits benötigt um die Information zu speichern.
Die mit Huffman codierten Zeichen haben also eine variable Bitlänge, die angepasst ist auf die Häufigkeit des Erscheinens des einzelnen Zeichens. Bei dem Satz „Barbara mag Rhabarber“ werden für die binäre Darstellung höchstens 4 Bit pro Zeichen verwendet, wodurch wertvoller Speicherplatz gespart wird ohne dass der Inhalt der Botschaft verändert wird. In Bezug auf Audiocodierung wird mit dem Huffmanverfahren durchschnittlich eine Reduktion von 1:2 erreicht. Da die Codierungsmethode nicht eindeutig ist, muss die Struktur des Baumes bei der so codierten Datei mitgeliefert werden.

## 3.2 Verdeckungsschwelle

Mit Verdeckungsschwelle bzw. *Maskierung* ist ein Verfahren gemeint, welches sich einen Selektionseffekt des menschlichen Ohres zu nutze macht. Diesen Effekt kann jeder nachvollziehen, der schon einmal versucht hat sich an einer Autobahn zu unterhalten. Die Stimme des Gegenübers wird unhörbar sobald ein LKW vorbeibraust. Die leisen Töne werden also durch gleichzeitiges Auftreten von lauten Tönen überdeckt. Dieses Phänomen wird auch

*simultane Verdeckung* genannt. Töne, die sowieso nicht wahrgenommen werden können verbrauchen also nur Speicherplatz und können deshalb aus der Datei eliminiert werden. Der Maskierungseffekt hält sogar noch über die eigentliche Abspielzeit des lauten Signals an. Ein extremes Beispiel hierzu: Jeder mit normalem Hörvermögen, der eine Viertelstunde neben einem laufenden Presslufthammer verbracht hat, wird auch eine ganze Zeit lang danach kaum noch etwas hören können. Flüstertöne nach dieser Art Störgeräusch kommen beim Zuhörer nicht an und können deshalb eingespart werden. Diese Verschiebung der Verdeckungsschwelle wird *zeitliche Maskierung* genannt.
Der deutsche Physiker Heinrich Barkhausen (1881 - 1956) hat den Maskierungseffekt verwendet, um den menschlichen Hörbereich in 24 sog. kritische Bänder aufzuteilen. Er hat dabei festgestellt, dass die Breite dieser Bänder nicht konstant ist, sondern sich mit der mittleren Bandfrequenz verändert. Als Einheit für die Breite der Bänder wird in der Psychoakustik *Bark* verwendet.

| Z<br>Bark | $f_u$, $f_o$<br>Hz | $f_m$<br>Hz | Z<br>Bark | Δfg<br>Hz | 10 lg Δfg*<br>dB |
|---|---|---|---|---|---|
| 0 | 0 | 50 | 0,5 | 100 | |
| 1 | 100 | 150 | 1,5 | 100 | 20 |
| 2 | 200 | 250 | 2,5 | 100 | 20 |
| 3 | 300 | 350 | 3,5 | 100 | 20 |
| 4 | 400 | 450 | 4,5 | 110 | 20 |
| 5 | 510 | 570 | 5,5 | 120 | 21 |
| 6 | 630 | 700 | 6,5 | 140 | 21 |
| 7 | 770 | 840 | 7,5 | 150 | 22 |
| 8 | 920 | 1.000 | 8,5 | 160 | 22 |
| 9 | 1,080 | 1.170 | 9,5 | 190 | 23 |
| 10 | 1,270 | 1.370 | 10,5 | 210 | 23 |
| 11 | 1,480 | 1.600 | 11,5 | 240 | 24 |
| 12 | 1,720 | 1.850 | 12,5 | 280 | 25 |
| 13 | 2,000 | 2.150 | 13,5 | 320 | 25 |
| 14 | 2,320 | 2.500 | 14,5 | 380 | 26 |
| 15 | 2,700 | 2,900 | 15,5 | 450 | 27 |
| 16 | 3,150 | 3,400 | 16,5 | 550 | 27 |
| 17 | 3,700 | 4,000 | 17,5 | 700 | 28 |
| 18 | 4,400 | 4,800 | 18,5 | 900 | 29 |
| 19 | 5,300 | 5,800 | 19,5 | 1,100 | 30 |
| 20 | 6,400 | 7,000 | 20,5 | 1,300 | 32 |
| 21 | 7,700 | 8.500 | 21,5 | 1,800 | 32 |
| 22 | 9,500 | 10,500 | 22,5 | 2,500 | 34 |
| 23 | 12,000 | 13,500 | 23,5 | 3,500 | 35 |
| 24 | 15,500 | | | | |

* 10 lg Δfg = Zunahme des Pegels (in dB) von Ausschnitten aus weissem Rauschen bei Verbreiterung des Frequenzbandes von 1 Hz auf die Breite Δfg der Frequenzgruppe.
Z = Frequenzgruppe, $f_u$ = untere Grenzfrequenz, $f_o$ = obere Grenzfrequenz, $f_m$ = Mittenfrequenz, Δfg = Bandbreite

*Abb:3.2.1 Wertetabelle zur Einheit Bark*

Für Frequenzen (*f*) unter 500 Hz ist ein Bark gleich *f*/100 und für die Frequenzen über 500 Hz wird ein Bark mit 9+4*log(*f*/1000) berechnet, wie bei [hen03] nachgelesen werden kann. Aufgrund der Signalstärke in einem Frequenzbereich kann anhand der *Barkhausen-Bänder* die Dauer der Maskierung errechnet werden und Informationen zu den verdeckten Tönen können ohne Beeinträchtigung des Hörgenusses entfernt werden.

## 3.3 Predictive Coding

Dies ist eine Methode, bei der aus den bereits abgespielten Signalen das wahrscheinlich folgende Signal errechnet wird. Entspricht das folgende Signal nicht der Vorhersage, wird lediglich die Differenz zum vorher angenommen Signal gespeichert. Diese Differenzen brauchen weniger Speicherplatz als das ursprüngliche Signal und deshalb kann die Audiodatei so effektiv komprimiert werden.
Tilman Liebchen von der TU Berlin hat mit diesem Verfahren den bekannten *LPAC* - **L**ossless **P**redictive **A**udio **C**ompression - entwickelt, der Audiodateien , sog. *pac*-Dateien, für Unixsysteme verlustfrei komprimieren kann. Dieser Codec hat mittlerweile seinen Nachfolger in MPEG-4 *LAC* **L**ossless **A**udio **C**oding bzw. in Ogg *FLAC* **F**ree **L**ossless **A**udio **C**oding gefunden.

## 3.4 Transform Coding

Durch Transformationscodierung werden Daten in einen besser zu komprimierenden, mathematischen Raum übertragen. Ein hierbei weit verbreitete Verfahren ist die *Fourier-Transformation* von Zeit- nach Frequenzbereich. Die Schallwellen werden als Polynome aufgefasst und die Koeffizienten werden auf Häufigkeit ihres Vorkommen untersucht. Hierbei fallen kleinere Koeffizienten weg und so entsteht der übersichtliche Frequenzraum.

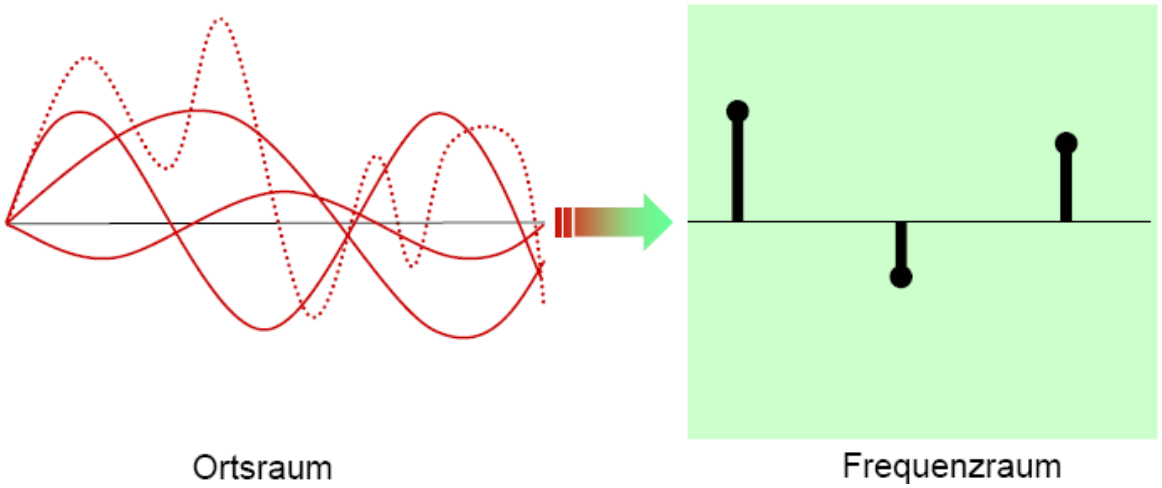


*Abb.:3.4.1 Fourier Transformation [mei04]*

Die Formel für die DFT lautet für 2n:

$$f_j = \sum_{k=0}^{2n-1} x_k \, e^{-\frac{2\pi i}{2n} jk} \qquad j = 0, \ldots, n-1.$$

Die effektivsten Verfahren dieser Art sind die sog. *DCT* – **D**iskrete **C**osinus **T**ransformation und die *FFT*, also die schnelle Fourier- Transformation, die deshalb als schnell bezeichnet werden kann, weil sie nach dem klassischen *Divide-and-Conquer* Prinzip arbeitet. Ausführliche Informationen zu den Berechnungsverfahren können bei [ste00] nachgelesen werden.

## 3.5 Sub Band Coding

Das zu kodierende Audiosignal wird durch eine Filterbank vom Zeit- in den Frequenzbereich transformiert, wobei es in zum Beispiel bei MP3 in 32 Frequenzbänder mit je 625 Hz (Subbänder) gleicher Breite unterteilt wird.

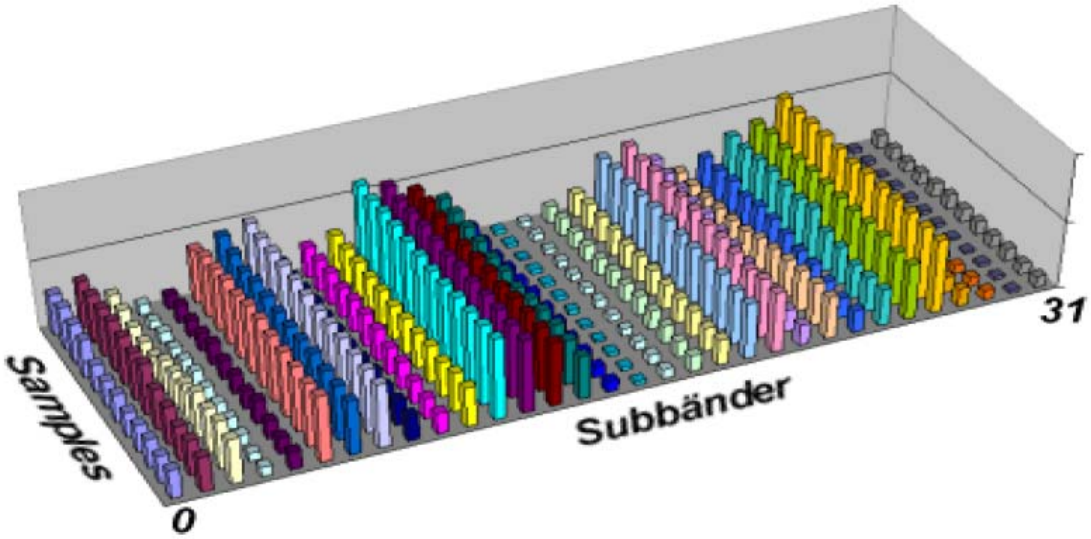


*Abb.34.5.1 Subband Coding, Quelle [kle05]*

Es wird davon ausgegangen, dass immer nur bestimmte Frequenzmaxima benötigt werden für die korrekte Klangwiedergabe. Dieses Verfahren wird auch *selektive Frequenztransformation* genannt und eignet sich insbesondere gut zur Komprimierung von Sprachdateien [ste00]. Bei MP3 wird für 32 eingelesene Samples pro Subband ein Sample ausgegeben. Die Subbänder werden durch eine modifizierte diskrete Cosinus-Transformation (*MDCT*) jeweils nochmals in 18 Teilbereiche unterteilt. Dadurch ergibt sich eine höhere Spektralauflösung. Eine mögliche Überlappungsgefahr der Bänder wird dadurch vermindert und somit wird auch die Wahrscheinlichkeit des Auftretens von Aliasing -Artefakten vermindert.

## *4. Formate und Codecs*

Zur Abspeicherung auf digitalen Medien gibt es mittlerweile jede Menge verschiedener Verfahren, wobei folgend auf die wichtigsten eingegangen wird. Der Schwerpunkt liegt auf dem MP3 Verfahren, denn die Entwicklung dieses Verfahrens hat einen nachhaltigen Einfluss auf die Multimedia – Welt gehabt und ist bis heute am weitesten verbreitet.

## 4.1 WAV

Das Wave Form Audio File Format zur Abspeicherung von Audiodateien wurde gemeinsam von IBM und Microsoft entwickelt und heißt eigentlich RIFF – WAVE, denn es ist ein Bestandteil des RIFF – **R**esource **I**nterchange **F**ile **F**ormat - von Windows. Bei WAV handelt es sich um ein unkomprimiertes Dateiformat und die Daten werden häppchenweise in sog. Chunks unterteilt und abgespeichert:

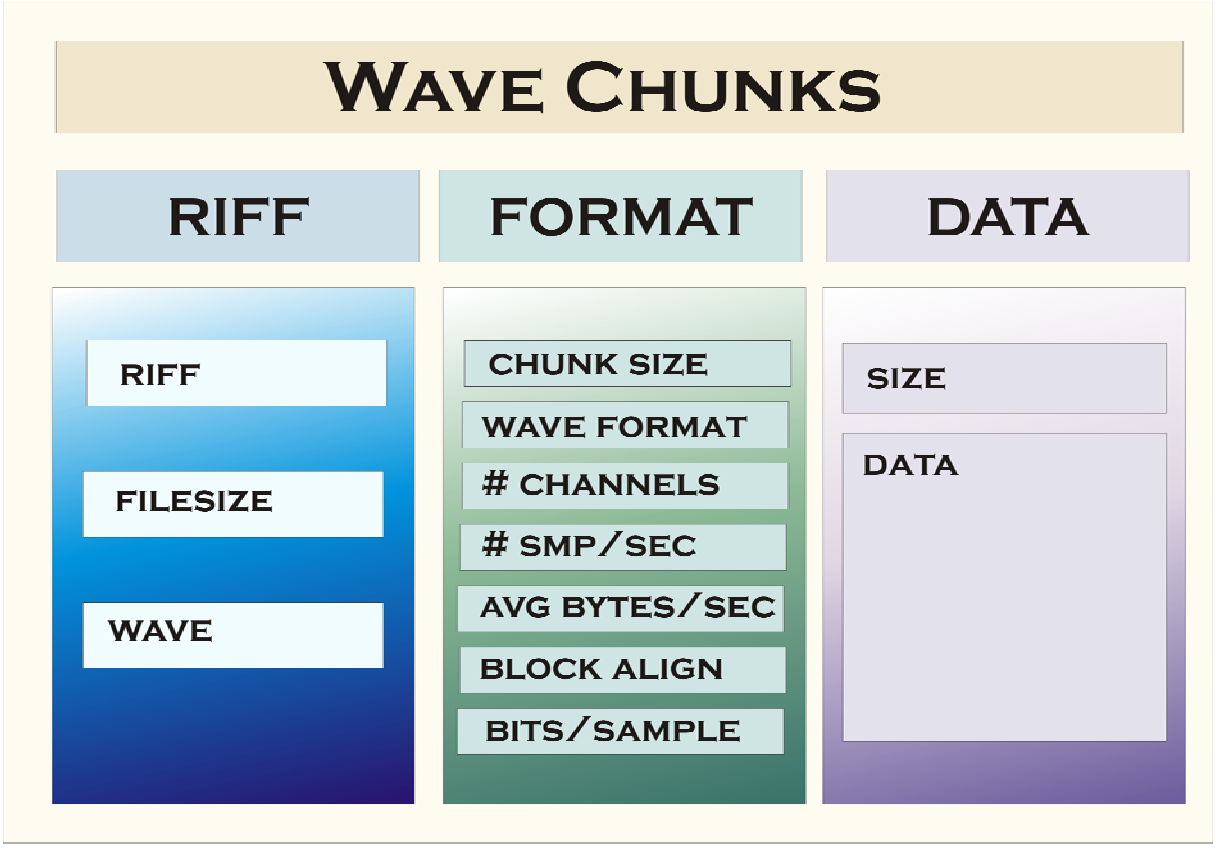


*Abb 4.1.1 Dateikonzept von WAV*

Im *Riff-Chunk* werden die Audioattribute wie WAVE angegeben wo hingegen im Format Chunk Aussagen über die genauen Einstellungen zur wav-Datei gemacht werden. Erst im Datenchunk befinden sich die eigentlichen Audiodaten. Diese 3 Chunks sind das notwendige Grundgerüst zur Abspeicherung von wav-Dateien. Es existieren jedoch noch weiter Chunks wie zum Beispiel *Cue* - und *Playlist* Chunk, die eingesetzt werden um die Abspielreihenfolge festzulegen.

## 4.2 MIDI

Das Akronym *MIDI* steht für den englischen Begriff **M**usical **I**nstrument **D**igital **I**nterface welcher die ursprüngliche Idee, die hinter diesem ungewöhnlichen Audio-Datenformat steckt sehr treffend umschreibt. Das MIDI Protokoll wurde 1983 entwickelt [mei04], um die Kommunikation zwischen verschiedenen Synthesizern zu ermöglichen. Es werden Kontrollsignale zu der jeweiligen Hardware gesendet, die darauf hin den gewünschten Ton produziert. Das heißt, dass die Hardware die Musik erzeugt und lediglich die Steuersignale gespeichert werden müssen. Diese Technik sorgt für vergleichsweise kleine Dateien – allerdings ist die Qualität der Wiedergabe immer von der beim Empfänger installierten Hardware abhängig.

Soll diese Methode zum Abspeichern von Audiodateien genutzt werden, stellt sich ein weiterer Nachteil heraus: Das Abspeichern von Gesang ist nahezu unmöglich. Es kann bestenfalls die Melodie gespeichert werden. So kann beispielsweise der Song *Daniel* von Elton John in MIDI in nur 20 KB abgespeichert werden [ekn], auf die Stimme von Elton John muss der Musikfreund beim Anhören leider (oder zum Glück?) verzichten.
Allerdings wird dieser Nachteil durch nicht zu unterschätzenden Vorteile wieder aufgewogen von denen ich hier einen besonders hervorheben möchte: Da die Töne als Steuersignale abgespeichert werden, können sie 1:1 als Noten wiedergegeben werden und der interessierte Musiker hat mit Programmen wie *Capella* [cap] die Möglichkeit sich die Notenblätter von Songs im MIDI – Format ausdrucken zu lassen.

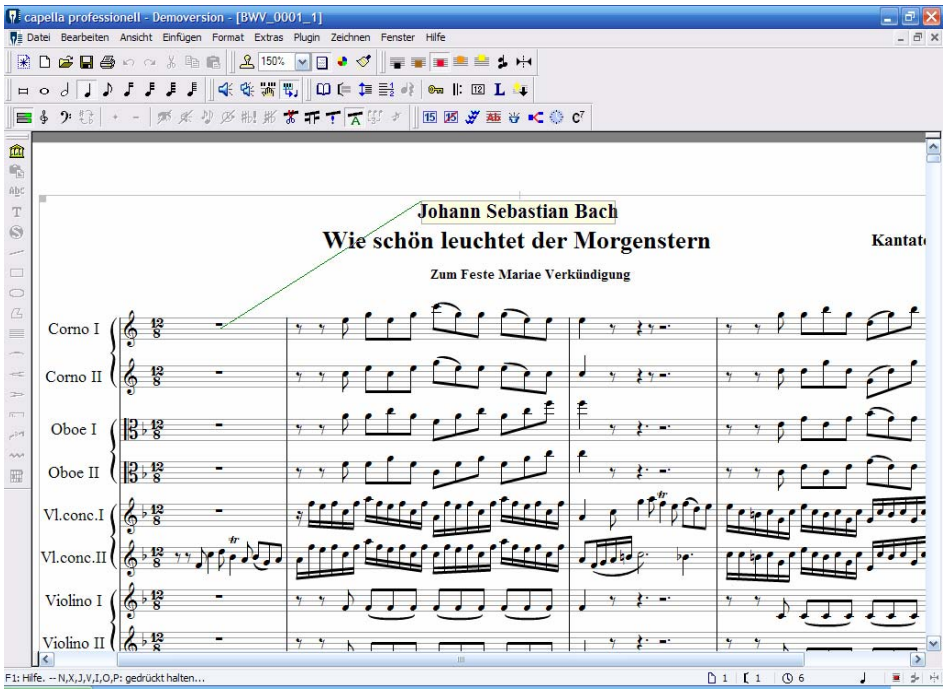

*Abb.4.2.1 MIDI Dateibearbeitung mit Capella*

Wer über ein MIDI Keyboard verfügt, kann dieses als Eingabeinstrument nutzen und sich die eingespielten Noten mit Programmen wie Capella direkt als Partitur anzeigen lassen.
Zur Verbindung von MIDI fähigen Geräten werden 2 Kabel mit 5-Poligen Steckern benötigt, die die beiden Geräte direkt MIDI –In beziehungsweise MIDI – Out miteinander verbinden. Alternativ dazu kann der Computer über die USB-Schnittstelle mit dem Keyboard verbunden werden. Im einschlägigen Fachhandel sind zum Beispiel USB-MIDI Hubs und entsprechende Kabel erhältlich.
Die ankommenden Steuersignale werden von der Hardware mittels eines *Wavetables* in möglichst realistische Klänge umgewandelt. Besitzer von Soundkarten ohne Wavetable können diese durch Installation sogenannter Soundfonts entsprechend ergänzen [sfar]
Eine Erweiterung des Standard MIDI Formates ist *General MIDI*. Hier sind die Kanäle festgelegt, die für die jeweiligen Instrumente stehen [midi]. GM1 ist ein Standard, der 1991 von der Firma Roland entwickelt wurde und als kleinster gemeinsamer Nenner von den unterschiedlichen Firmen gesehen worden ist. Allerdings haben sowohl *Roland* als auch *Yamaha* noch zusätzliche Standards – GS und XS – entwickelt, was dazu führte, dass die unterschiedlichen Geräte nicht mehr einfach miteinander verkoppelt werden konnten. Seit 1998 haben sich Yamaha und Roland auf GM2 als gemeinsamen Nenner geeinigt und so wird dieser Standard bis heute gemeinsam genutzt und weiterentwickelt.

## 4.3 MP3

Das MP3 Format wird zur Zeit weltweit am häufigsten verwendet um Audiodateien verkleinert abzuspeichern. Eigentlich heißt es MPEG 1 Layer 3 Format, wobei das Akronym MPEG für **M**otion **P**icture **E**xperts **G**roup steht, denn dieses Format ist zur Komprimierung von Videodaten gedacht – es wird jedoch vermehrt der Audiocodec dieses Formates benutzt. Der Ansporn zur Entwicklung war, ein Audioformat zu entwickeln welches die Versendung von hochwertigen aber wertvolle Übertragungszeit sparenden Audiodateien über ISDN ermöglichte. Der technische Sprung sollte so eindrucksvoll sein, wie seinerzeit die Umstellung von schwarz/weiß auf Farbfernsehen. Als das MP3 Verfahren jedoch vor 15 Jahren entwickelt wurde, glaubte in Deutschland noch kaum jemand daran, dass es möglich sein könnte einen Chip zu kreieren, der dermaßen komplizierte Vorgänge wie zum Beispiel das psychoakustische Modell beinhalten könnte. Auch wurde vermutet, dass die Kosten für eine etwaige Entwicklung immens hoch wären und so räumte die deutsche Wirtschaft dem Projekt keinerlei Erfolgschancen ein. Somit war Geburtsort des ersten MP3-Players zwar Erlangen - denn hier entwickelten das Fraunhofer IIS und der Halbleiterhersteller Intermetall 1994 das weltweit erste Modell, jedoch der wirtschaftliche Siegeszug von MP3 begann zunächst außerhalb Deutschlands, wie auch aus dem Interview mit Prof. Gerhäuser und Prof. Brandenburg [bac07] hervorgeht.


*Abb.4.3.1 Der erste MP3-Player, Quelle [MP3]*

Der Prototyp des ersten MP3-Players wog etwa 2 Kilo und speicherte die Audiodaten mit einer Minute Spieldauer auf einem eingebauten EPROM ab. Dies war zwar noch nicht von großem praktischem Nutzen, stellte jedoch einen Durchbruch dar, denn nun war es endlich möglich Musik ohne den Einsatz mechanischer Teile abzuspielen.

## Überblick zum Komprimierungsvorgang

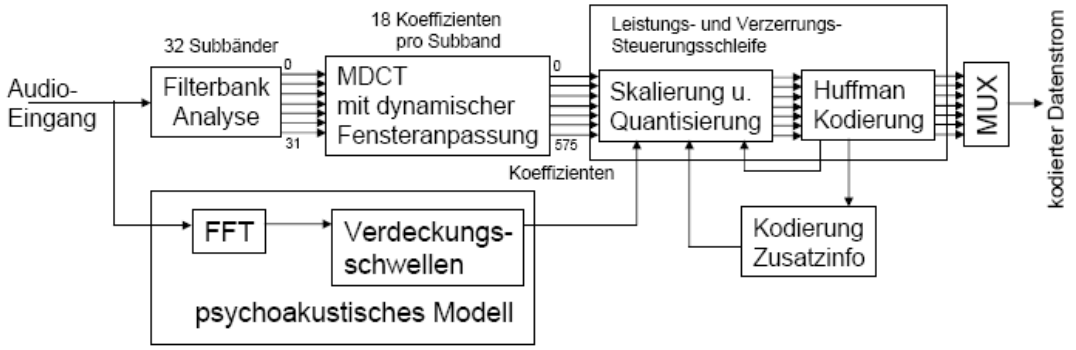


*Abb.4.3.2 MP3 Komprimierungsschema, Quelle [mei04]*

Wie aus obiger Abbildung ersichtlich durchläuft der Audiodatenstrom einen relativ komplexen Bearbeitungsvorgang der Kodierung, der sich aber lohnt. Bei MP3 können durch die variablen Bitraten enorme Komprimierungen erreicht werden. So wird bei Telefonqualität mit 8 kBit/s eine Komprimierung von 96:1 erzielt. Für den Anspruch auf CD-Qualität wird die Audiodatei immerhin bei 128 kBit/s im Verhältnis 12:1 verkleinert.

## Aufbau des MP3-Encoders

**Filterbank**
Das Eingangssignal wird in einer hybriden Polyphasen/MDCT-Filterbank in unterabgetastete Spektralkomponenten zerlegt. Die Filterbank und die entsprechende inverse Filterbank im Decoder bilden zusammen ein Analyse-/Synthese-System.

**Wahrnehmungsmodell**
Mittels psychoakustischer Regeln wird eine Abschätzung der (zeit- und frequenzabhängigen) Maskierungsschwellen berechnet.

**Quantisierung und Codierung**
Die Spektralkomponenten werden quantisiert und codiert, wobei das dabei entstehende Quantisierungsrauschen, soweit möglich, unter der Maskierungsschwelle gehalten wird.

**Erzeugung des Bitstroms**
Ein Bitstrom-Formatierer setzt aus den quantisierten und codierten Spektralkoeffizienten und Seiteninformationen wie z.B. der Bit-Verteilung den MP3-Bitstrom zusammen.

**Mono und Stereo**
MP3 funktioniert sowohl mit Mono- als auch mit Stereo-Audiosignalen. Eine Technik namens "Joint Stereo Codierung" kann für die effiziente kombinierte Codierung des rechten und linken Kanals genutzt werden. Alternativ wird Mitten-/Seiten-Codierung oder Intensitäts-Stereocodierung eingesetzt. Die letztere Methode ist besonders bei niedrigen Bitraten nützlich, verändert aber eventuell das Klangbild.

**Mehrkanal-Audio**
Das herkömmliche MP3-Format ist in der Lage, Audiosignale mit einem oder zwei Kanälen zu codieren. 2004 hat das Fraunhofer IIS eine rückwärtskompatible Erweiterung für Mehrkanalton im 5.1-Kanal-Format eingeführt, den sog. MP3 Surround.

**Abtastraten**
MP3 funktioniert mit verschiedenen Abtastraten. Bei MPEG-1 Layer III sind 32 kHz, 44,1 kHz und 48 kHz definiert. In MPEG-2 sind zusätzlich Abtastraten von 16 kHz, 22.05 kHz und 24 kHz zugelassen. "MPEG-2.5" ist der Name einer vom Fraunhofer IIS eingeführten Erweiterung für MP3, die schon bei sehr niedrigen Datenraten zufrieden stellend arbeitet und die zusätzlichen Abtastraten 8 kHz, 11,025 kHz und 12 kHz einführt.

**Datenrate**
Bei MP3 bleibt die Wahl der Datenrate – in bestimmten Grenzen – dem Programmierer oder dem Nutzer des MP3-Encoders überlassen. Der Standard definiert ein Set von Datenraten zwischen 8 kBit/s und 320 kBit/s. Außerdem muss der MP3-Decoder die Umschaltung von Datenraten zwischen einzelnen Datenblöcken unterstützen. Durch Verwendung der so genannten Bitsparkasse ist so die Codierung mit variablen und konstanten Datenraten bei jedem Wert innerhalb der Grenzen des Standards möglich.

**Datenrate und Audioqualität**
Im MP3-Encoder wird die verfügbare Datenrate so verteilt, dass das Quantisierungsrauschen möglichst nicht hörbar ist. Bei niedrigen Datenrate kann das Quantisierungsrauschen also hörbar werden. Die Audioqualität des codierten Materials ist deshalb direkt von der Datenrate abhängig. Obwohl MP3 im Bereich von 8 kBit/s bis 320 kBit/s genutzt werden kann, ist zu empfehlen, Datenraten ab 80 kBit/s für Mono oder 160 kBit/s für Stereosignale zu verwenden. Für Anwendungen mit sehr niedrigen Datenraten sind die MPEG-4 Audiocodecs besser geeignet als MP3.
*Quelle [MP3]*

Natürlich muss die Kodierung auch wieder rückgängig gemacht werden können. Auch dieser Vorgang stellt einen nicht unerheblichen Rechenaufwand dar. Nachfolgend das

**Schema zur Dekodierung**

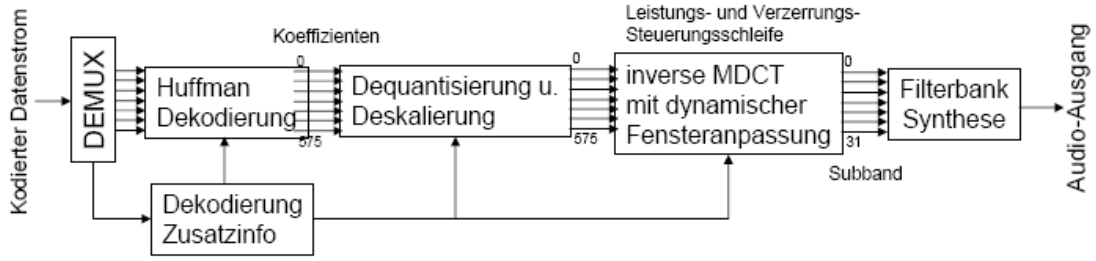


*Abb. 4.3.3 MP3 Decodierungsschema, Quelle [mei04]*

Zunächst durchläuft der kodierte Datenstrom den Huffman Decodierungsvorgang und wird dequantisiert. Nach der Umkehrung der MDCT erfolgt die Zusammenführung der unterabgetastete Spektralkomponenten, die auch Subbänder genannt werden. Nach Abschluss der Filterbanksynthese wird der Audiodatenstrom an den Audio-Ausgang weiter gegeben und die Audiodatei ist abspielbar.

Als der verbesserte Nachfolger zum erfolgreichen MP3 gilt MPEG2 / 4 AAC.[bac07]

## 4.4 Weitere Codecs

**Ogg**
Der etwas merkwürdig anmutende Name entstammt einem Computerspiel namens *Netrek* und bedeutet soviel wie „Etwas furchtlos angehen, aber dabei die Zukunft nicht aus den Augen verlieren." Bei Ogg-Vorbis und Ogg-Flac handelt es sich um lizensfreie Audiocodecs, deren Entwicklung begann als FhG/Thomson 1998 Lizenzgebühren für MP3-Encoder verlangten. Er ist MP3 recht ähnlich und insbesondere bei niedrigen Bitraten sogar noch besser. Mittlerweile haben ogg-Dateien eine große Verbreitung gefunden und sind insbesondere wegen der Streaming Technologie interessant. Ferner gibt es von Ogg noch den Text-Codec Writ, den Sprachdaten-Codec Speex und den Video-Codec Theora [vorb].

**RM**
Das Realmedia-Format von der Firma Realnetworks hat sich insbesondere in der Audiostream-Technologie bewährt und eignet sich sowohl zur Verarbeitung von Live-Streams als auch für On Demand-Streams. Der Player zur Wiedergabe kann frei im Web unter *www.real.com* heruntergeladen werden und dementsprechend weit verbreitet ist das RM-Format. Es eignet sich auch dazu zum Beispiel Radiosendungen live zu einem Audiostream zu enkodieren oder auch Vorlesungen inklusive Video als Stream über das Web zur Verfügung zu stellen. Für die Übertragung über das Internet wird die zu übertragende Datei in kleine Häppchen aufgeteilt, die on the fly komprimiert werden und jeweils einen eigenen Header erhalten. So kann der Anwender parallel zum herunterladen der Datei diese schon konsumieren. Optimal ist, wenn die Bandbreite beim Empfänger höher ist als die beim Sender. Da RM ein Format ist welches sich insbesondere für die schnelle live Übertragung im Internet eignet, ist die Qualität generell entsprechend niedrig gehalten. Zur Archivierung eignet es sich also weniger, wie eigentlich alle verlustbehafteten Formate. Je geringer die Bandbreite beim Empfänger, desto stärker wird die Audiodatei reduziert. Zur Komprimierung werden zunächst nur die Frequenzen entfernt, die für den Menschen unhörbar sind. Die Übertragungsqualität ist aber auch abhängig vom Inhalt der Datei. Soll es zum Beispiel um die Übertragung von Sprache gehen, können dementsprechend noch mehr irrelevante Frequenzen entfernt werden. Eine zu geringe Bandbreite sorgt jedoch für sehr schlecht zu verstehende Audiodateien und ist deshalb ein wichtiges Qualitätskriterium.

**WMA/ASF**
Bei dem **W**indows **M**edia **A**udio Codec handelt es sich um ein Containerformat, das von Microsoft entwickelt worden ist, eigens um Audiostreams im **A**dvanced **S**treaming **F**ormat über das Internet gut und schnell hörbar zu machen. Nach eigenen Angaben soll hier mit Bitraten die kleiner oder gleich 64 kbps sind CD-Qualität erreicht werden – was jedoch bei Hörproben subjektiv nicht so empfunden wurde. Da sich wma-Dateien jedoch nicht so leicht kopieren lassen, meinte man dass sie sich gut eigneten für die kommerzielle Verbreitung und Nutzung übers Web. Allerdings gibt es zum Beispiel eine relativ simple Methode über die Soundkarteneinstellungen auch diese Dateien zu kopieren und im WAVE-Format abzuspeichern. Da hierfür ein relativ hoher Zeitaufwand notwendig ist, haben findige Softwareentwickler aber auch schon für dieses Problem bereits 2004 eine Lösung entwickelt in Form von Programmen wie *TuneBite*, welches u.a. WMA-Dateien in OGG umwandelt und abspeichern kann.

Da also der eigentliche Algorithmus zum Kopierschutz relativ problemfrei ausgehebelt werden kann, ist die mitunter ziemlich schlechte Qualität dieser Audiodateien wohl das einzige Mittel, dass Audiophilisten von unerwünschten Downloads fernhält und somit eignet

sich das Format zumindest dazu Musikdateien zu veröffentlichen, die einen ersten Eindruck vom Künstler vermitteln und somit zum Kauf der Audio-CD animieren sollen.

**Dolby**
Die Firma Dolby hat mit den Audioformaten AC-1 bis AC-3 proprietäre Audiocodecs entwickelt, die nicht offen gelegt sind. AC-1 wurde seinerzeit im Hinblick auf HDTV entwicklt; AC-2 stellt eine verbesserte Variante im Vergleich zu AC-1 dar und AC-3 unterstützt das Mehrkanaltonverfahren für den *Surroundsound* mit Dolby 5.1. Die Ziffer 5 steht hierbei für die 3 Primärkanäle zusammen mit den beiden Surroundkanälen und die 1 für den niederfrequenten Basskanal. Dieses Format wird in den U.S.A für HDTV und DVD genutzt und ist deshalb nicht kompatibel zum europäischen HDTV, denn dies wird mit dem MP3-Verfahren kodiert. Weiterführendes über die Firma Dolby und die verschiedenen Formate kann im Internet unter *www.dolby.com* nachgelesen werden.

**NeXT/Sun Audio File Format**
Dieses Format wurde zeitgleich mit der Verbreitung den NeXT-Computern bekannt und ist eine Entwicklung von Sun Microsystems. Die übliche Dateiendung ist *au*. Der Header einer AU-Datei ist in 4 Byte Blöcke unterteilt und im *big-endian* Format abgelegt. Daran schließt sich ein Textbereich an, der die eigentlichen Audioinformationen beinhaltet. Das au-Format unterstützt viele Audiformatierungsmethoden aber zumeist wird das sog. *µ-law logarithmic encoding* verwendet. Ein Überblick über die verschiedenen Erscheinungsweisen einer AU-Datei verschafft die folgende Tabelle:

| 32 bit word | Feld | Beschreibung/ Inhalt Hexadecimal Zahlen in C-Notation |
|---|---|---|
| 0 | magic number | Der Wert `0x2e736e64` (vier ASCII Zeichen ".snd") |
| 1 | data offset | Offset der Daten in Bytes. Die kleinste gültige Zahl ist 24 (dezimal), da dies die Länge des Header ist (5 32-bit words) plus midestens noch 4 bytes für den *information chunk.* |
| 2 | data size | Dateigröße in Bytes. Falls unbekannt, sollte `0xffffffff` genutzt werden. |
| 3 | encoding | Datenkodierformat:<br>• 1 = 8-bit G.711<br>• 2 = 8-bit linear PCM<br>• 3 = 16-bit linear PCM<br>• 4 = 24-bit linear PCM<br>• 5 = 32-bit linear PCM<br>• 6 = 32-bit IEEE Fließkommazahl<br>• 7 = 64-bit Fließkommazahl<br>• 8 = Fragmentierte Samples |

| | | |
|---|---|---|
| | | • 9 = DSP Programm<br>• 10 = 8-bit Festkommazahl<br>• 11 = 16-bit Festkommazahl<br>• 12 = 24-bit Festkommazahl<br>• 13 = 32-bit Festkommazahl<br>• 18 = 16-bit linear mit Betonung<br>• 19 = 16-bit linear komprimiert<br>• 20 = 16-bit linear mit Betonung and komprimiert<br>• 21 = „Music-kit“ DSP Kommandos<br>• 23 = 4-bit ISDN µ-law komprimiert mit ITU-T G.721 ADPCM<br>• 24 = ITU-T G.722 ADPCM<br>• 25 = ITU-T G.723 3-bit ADPCM<br>• 26 = ITU-T G.723 5-bit ADPCM<br>• 27 = 8-bit G.711 A-law |
| 4 | sample rate | Die Anzahl der Samples/Sekunde (z.B.., 8000) |
| 5 | channels | Die Anzahl der ineinander verschränkten Kanäle (z.B.., 1 für mono, 2 für stereo, weitere Kanäle sind möglich – können aber vielleicht nicht vom Empfänger dekodiert werden) |

*Tab. 4.4.1 Kodiertypen bei au-Dateien, Quelle Wikipedia (englische Version)*

Aus der Tabelle wird ersichtlich, dass das AU-Format sowohl verlustfreie – wie PCM – als auch verlustbehaftete – wie ADPCM – Codecs unterstützt. Die Verkleinerung der Dateigröße anhand ADPCM beträgt in etwa 1:4. Das AU-Format ist auf Unix-Systemen sehr verbreitet.

## *Resumé*

Audiocodierung wird auch in Zukunft ein nicht mehr wegzudenkender Bestandteil unserer multimedial geprägten Welt sein. Sowohl im Internet aus auch in der Unterhaltungselektronik werden akzeptable Ergebnisse erst durch geschickt programmierte Komprimierungsalgorithmen erreicht. Doch sie sind nicht nur ein wichtiger Beitrag zu Muße und Unterhaltung im 21. Jahrhundert, sondern werden auch bei Spracherkennung und Prothetik eingesetzt. Durch Audiocodecs komprimierte Dateien sind heutzutage nicht nur klanglich einwandfrei, sondern eben auch klein genug um eine genügend schnelle Übertragungs- und Verarbeitungszeit zu leisten. Dies ist die Grundlage für den Einsatz in medizinischen Bereichen und somit kann die Entwicklung von passenden Audiocodecs auch eine große Hilfe und Unterstützung für zum Beispiel Patienten mit Gehörschäden sein. Bei kleinen Endgeräten und geringer Bandbreite sind die verlustbehafteten Codecs zurzeit die erste Wahl. Wenn es jedoch darum geht Audiodateien *persistent* zu archivieren, dann sollten eher verlustfreie Formate benutzt werden.

## *Literatur*

Benutzte Literatur sowie Empfehlungen zur Vertiefung des Themas

[sox] C. Bagwell, http://sox.sourceforge.net/AudioFormats.html, 1998
[flac] J. Coalson, http://flac.sourceforge.net/, 2007
[vorb] Xiph Open Source, http://www.vorbis.com/, 2005
[MP3] R. Ulrich, http://www.iis.fraunhofer.de/, 2007
[nerd] T. Arnold, http://www.omninerd.com/2004/12/10/articles/24, 2004
[mpeg] L.Chiariglione, http://www.chiariglione.org/mpeg/, 2006
[audi] M. Kremer, http://www.dasp.uni-wuppertal.de/ars_auditus/, 2004
[mei04] H. Sack, C. Meinel, WWW xpert.press , Springer-Verlag, 2004
[hen03] P. A. Henning, Multimedia, Carl Hanser-Verlag, 2003
[ste00] R. Steinmetz, Multimedia-Technologie, Springer-Verlag, 2000
[ekn]Tech JD, http://www.ekn.net/midi/Elton-John/index.html, 2003
[cap] E. Werner, http://www.whc.de/capella.cfm, 2007
[sfar] Personalcopy, http://www.personalcopy.com/sfarkfonts1.htm, 2003
[midi] MMA, http://www.midi.org/, 2007
[bac07] W.Back, http://www.media01-live.de/CC-Zwei-35.mp3, WDR, 2007
[kle05] M. Klein-Dasdamirov, Proseminar Audiokompression, TU-München, 2005
[huf52] D. Huffman, Minimum Redundancy Codes, IRE, Vol. 40 Sept. 1952
[wat96] J. Watkinson, Television Fundamentals, Elsevier, 1996

# *Index*

## R



## S



## T



## V



## W



## X



## Y



## Z