Seminararbeit Kompressionsalgorithmen: MP3 und Ogg Vorbis

Erik Noorman

1. Oktober 2012

Zusammenfassung

Diese Seminararbeit befasst sich mit den verlustbehafteten Audiokompressionsverfahren MP3 und Ogg Vorbis. Im ersten Kapitel werden physikalische Grundlagen erläutert, welche zum Verständnis der im zweiten Kapitel beschriebenen psychoakustischen Phänomene unserer Wahrnehmung dienen. Das dritte und vierte Kapitel befasst sich mit MP3 und Ogg Vorbis im Detail, welche sich die psychoakustischen Phänomene zu Nutze machen, um die Audiodateien zu komprimieren. Im fünften Kapitel werden die Vor- und Nachteile beider Verfahren erörtert.

1 Physikalische Grundlagen

1.1 Schall

Schall entsteht durch die Bewegung oder Vibration eines Objektes. Diese Bewegung manifestiert sich in dem umgebenden elastischen Medium (normalerweise Luft) als Abfolge von Druckschwankungen. Die Moleküle werden näher aneinander gedrückt und dann weiter auseinander gezogen als im Ruhezustand. Die Schallwelle bewegt sich weg von dem schallerzeugenden Objekt. Die Moleküle hingegen bewegen sich nicht in Richtung der Schallwelle, sondern vibrieren um eine Ruhelage [6, S. 2].

Schall ist also die mechanische Ausbreitung einer Druckwelle in einem elastischen Medium. Dabei werden die Moleküle in Gasen, Flüssigkeiten oder Festkörpern bewegt, beziehungsweise zum Schwingen angeregt.

1.1.1 Schalldruckpegel

Schall kann als Variation des Schalldrucks über die Zeit $p(t)$ beschrieben werden. Verglichen mit dem Wert des Luftdrucks sind die zeitvariierenden Schalldrücke, welche durch Schallquellen erzeugt werden, extrem klein. Daher ist es sinnvoll, Schall als Verhältnis des Schalldrucks zu einem Bezugsschalldruck p_0 darzustellen. Da das auditive System mit einem großen Be-

reich von Schalldrücken umgehen kann, ist es umständlich mit den Schalldrücken direkt zu arbeiten. Stattdessen wird der Schalldruckpegel L verwendet, ein logarithmisches Maß, welches dem Verhältnis zweier Schalldrücke entspricht [1, S. 1]:

$$L = 20 \cdot log(\frac{p}{p_0})dB$$

Der Bezugswert p_0 ist standardisiert auf $p_0 = 20\mu Pa$.

1.2 Digitalisierung von Audiosignalen

Schall ist zunächst analog, also zeit- und wertkontinuierlich und muss daher zur Speicherung und Verarbeitung auf dem Computer digitalisiert bzw. diskretisiert werden. Hierzu muss das Signal in festen Zeitabständen abgetastet und die Werte zu diesen Zeitpunkten, quantisiert werden.

1.2.1 Abtastung

Im Jahre 1928 zeigte Harry Nyquist, dass ein Signal, dessen Fouriertransformation Null ist über einer Frequenz von $W\ Hz$, durch $2 \cdot W$ äquidistante Abtastpunkte pro Sekunde akkurat repräsentiert werden kann [7, S. 372].

Das Abtasttheorem besagt also, dass die Abtastfrequenz f_{abtast} mindestens zweimal so hoch sein muss, wie die höchste Frequenz in unserem Signal f_{max} [7, S. 429].

$$f_{abtast} > 2 \cdot f_{max}$$

1.2.2 Quantisierung

Bei der Quantisierung werden die Werte des analogen Signals in eine feste Anzahl von Stufen, auf- bzw. abgerundet. Die Anzahl der Stufen entspricht üblicherweise einer Zweierpotenz, wobei der Exponent der Anzahl von Bits entspricht, die notwendig sind, um den Wert abzuspeichern.

Wird ein Signal mit einem festen Wertebereich, wie in Abbildung 1, mit 4 Bit quantisiert, wird der Wertebereich zunächst in $2^4 = 16$ Stufen unterteilt. Ein Messwert wird dann auf den Stufenwert mit der geringsten Abweichung auf- bzw. abgerundet und der Index der Stufe ausgegeben.

1.2.3 Quantisierungsrauschen

Beim Auf- bzw. Abrunden wird ein nicht zu vernachlässigender Fehler gemacht, der sich im quantisierten Signal als Rauschen darstellt.

Die blaue Fläche in Abbildung 1 entspricht dem bei der Quantisierung hinzugefügte Quantisierungsrauschen. Wenn man das ursprüngliche Signal vom quantisierten Signal abzieht, erhält man das durch die Quantisierung

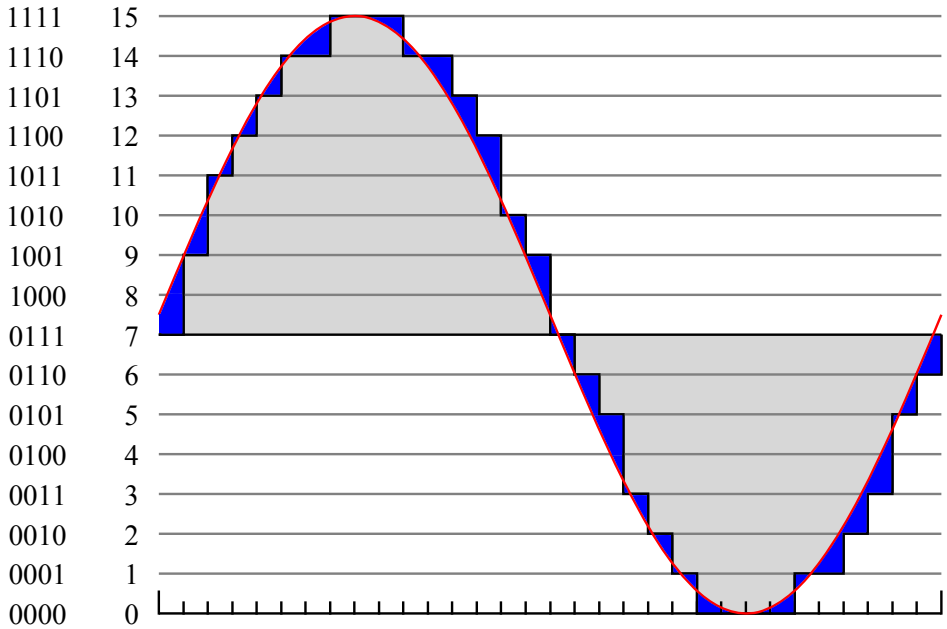

Abbildung 1: Quantisierung einer Sinusfunktion, Quelle: adaptiert von `http://de.wikipedia.org/wiki/File:Pcm.svg`

hinzugefügte Rauschen. Um nun eine Aussage über die Qualität der Quantisierung machen zu können, berechnet man das Signal-Rausch-Verhältnis (englisch: signal-to-noise ratio, SNR) [7, S. 198]:

$$SNR = 10 \cdot \log \frac{P_{Signal}}{P_{Rauschen}} dB$$

wobei P_{Signal} die mittlere Signalleistung und $P_{Rauschen}$ die mittlere Rauschleistung ist. Die mittlere Leistung eines Signals x in einem Intervall der Länge $T = [t_1, t_2]$ ist der Mittelwert über die quadrierten Koeffizienten des Signals [7, S. 198]:

$$P = \frac{1}{T} \cdot \int_{t_1}^{t_2} x_\tau^2 d\tau$$

oder für das diskrete Signal der Länge $N = [n_1, n_2]$ [7, S. 198]:

$$P = \frac{1}{N} \cdot \sum_{n=n_1}^{n_2} x_n^2$$

1.3 Fouriertransformation

Obwohl alle Audiosignale über ihre Druckvariation über die Zeit spezifiziert werden können, ist es sinnvoll, komplexe Signale auf eine andere Art darzustellen. Diese Darstellung basiert auf einem Theorem von Fourier. Fourier

zeigte, dass nahezu jeder komplexe Schwingungsverlauf in eine Serie von Sinuskurven mit spezifischen Frequenzen, Amplituden und Phasen, zerlegt werden kann [6, S. 3–4]. Mithilfe der Fouriertransformation kann ein gegebenes Zeitsignal in die Frequenzdarstellung überführt werden. Diese Methode wird in Abbildung 2 verdeutlicht: Die linke Spalte zeigt vier Schwingungsverläufe, deren Komplexität nach unten zunimmt. Die mittlere Spalte zeigt die Zerlegung dieser Schwingungsverläufe in Sinuskurven. Die rechte Spalte zeigt das korrespondierende Frequenzspektrum.

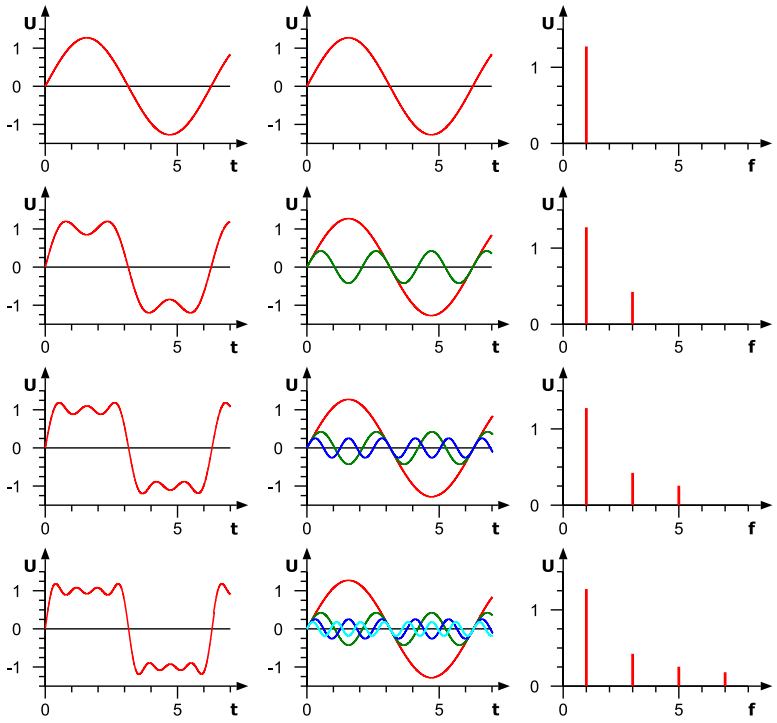

Abbildung 2: Zerlegung eines Schwingungsverlaufs in Sinuskurven und Frequenzspektrum, Quelle: adaptiert von
`http://de.wikipedia.org/wiki/Datei:Fourier_synthesis.svg`

1.3.1 Frequenzspektrum

Das Frequenzspektrum, oder auch Spektrum, enthält die Frequenzkomponenten eines Signals und wird repräsentiert durch die graphische Darstellung der Energie, Intensität, Amplitude oder den Pegel als Funktion der Frequenz [6, S. 404].

1.4 Modifizierte diskrete Kosinustransformation (MDCT)

Da die diskrete Fouriertransformation, bei zeitlich begrenzten Signalen, das Signal mehrfach hintereinander setzt, entstehen meist harte Sprünge im Si-

gnal, die vorher nicht vorhanden waren. Im resultierenden Frequenzspektrum sind dann Verzerrungen in Form von zusätzlichen Frequenzen, welche eigentlich nicht im ursprünglichen Signal vorhanden sind. Die diskrete Kosinustransformation spiegelt das Signal und vermeidet damit diese harten Sprünge. Die modifizierte diskrete Kosinustransformation verwendet zusätzlich Fenster, welche sich zu 50% mit dem vorherigen und dem folgenden überlappen. Dies hat den Vorteil, dass zeitliche Verzerrungen bei der Rücktransformation sich gegeneinander aufheben [7, S. 416–419].

2 Psychoakustische Effekte der Wahrnehmung

Dieses Kapitel beschreibt die Fähigkeit des auditiven Systems schwachen Schall wahrzunehmen, wenn kein anderer Schall präsent ist. Ebenso wird beschrieben wie man ein Schallereignis wahrnimmt, wenn Störschall präsent ist.

2.1 Hörschwelle

Die absolute Hörschwelle, oder auch Ruhehörschwelle, ist definiert als der Schalldruck, mit der ein einzelnes Schallereignis in Abwesenheit beliebigen Störschalls, also in absoluter Ruhe, von unserem auditiven System gerade noch wahrgenommen werden kann. Der normalhörende Mensch kann Frequenzen von 20 Hz bis ca. 20 kHz wahrnehmen [6, S. 55–56]. Alterung und Schädigung des Gehörs können unser wahrnehmbares Frequenzspektrum schmälern [1, S. 20].

Abbildung 3 zeigt die Hörfläche des normalhörenden Menschen als Schalldruckpegel in Abhängigkeit von der Frequenz. Die Hörschwelle verläuft nicht linear: Das auditive System ist besonders sensible für die mittleren Frequenzen zwischen 1–5 kHz und fällt zu den hohen und niedrigen Frequenzen stark ab.

2.2 Maskierungseffekte

Die Hörschwelle für eine spezifische Frequenz hängt nicht nur von der absoluten Hörschwelle ab, sondern ist auch abhängig von anderem Schall, welcher zeitnah und/oder frequenznah auftritt.

Machen wir uns dies an einem Beispiel klar: Wenn wir ein leises Gespräch an einem Bahnsteig führen und es fährt ein Zug direkt an uns vorbei, kann es sein, dass wir nur noch die Lippenbewegung von unserem Gesprächspartner sehen, jedoch kein Wort mehr akustisch wahrnehmen können. Sobald der Zug vorbei gefahren ist, können wir unser Gegenüber wieder einwandfrei hören. Der vom Zug erzeugte Schall hat also unsere Hörschwelle angehoben.

Wird ein Schallereignis, welches oberhalb der Hörschwelle liegt, nicht wahrgenommen, da dieser von weiterem Schall maskiert wird, spricht man

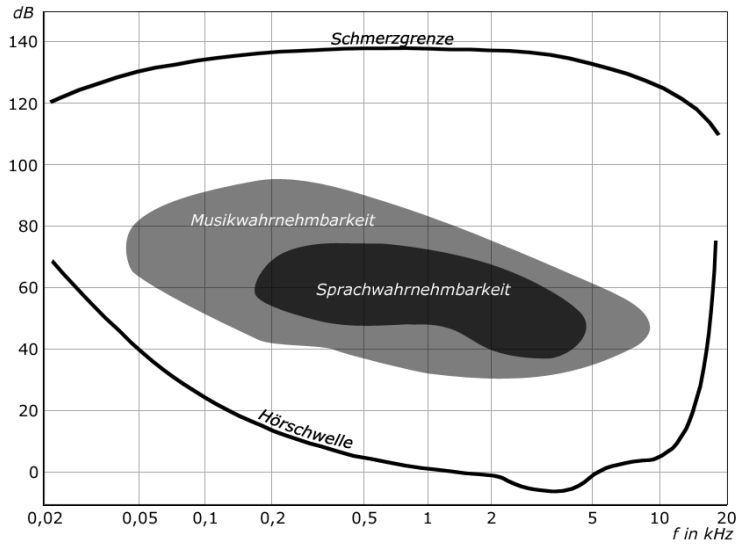

Abbildung 3: Hörfläche, Quelle:
`http://de.wikipedia.org/wiki/Datei:Hoerflaeche.png`

von Maskierung. Man unterscheidet zwischen einer Reihe verschiedener Arten von Maskierung, die zwei wichtigsten sind die Frequenzmaskierung und die Zeitmaskierung.

2.2.1 Maskierung im Frequenzbereich

Die Maskierung eines Signals tritt am ehesten durch Störschall auf, welcher, verglichen mit dem Signal, naheliegende oder gleiche Frequenzanteile enthält.

Abbildung 4 zeigt wie die Hörschwelle angehoben wird, wenn ein 1 kHz Störsignal mit verschiedenen Schalldruckpegeln vorhanden ist. Ist nun ein Störsignal von 1 kHz mit einem Schalldruckpegel von 80 dB präsent, kann man z.B. einen Sinuston, mit 2 kHz und einem Schalldruckpegel von 40 dB nicht wahrnehmen, obwohl er weit über der Ruhehörschwelle liegt.

2.2.2 Maskierung im Zeitbereich

Simultane Maskierung findet statt, wenn das Signal von einem gleichzeitig präsenten Störsignal maskiert wird. Allerdings kann ein Signal auch maskiert werden, wenn es kurze Zeit nach (Nachverdeckung) oder vor (Vorverdeckung) einem Störsignal auftritt.

Abbildung 5 soll diesen Umstand an folgendem Beispiel erläutern: Ein Störsignal (Breitbandrauschen), auch Maskierer genannt, ist über 200 ms präsent und hebt die Hörschwelle, für eine feste Frequenz, in diesem Zeit-

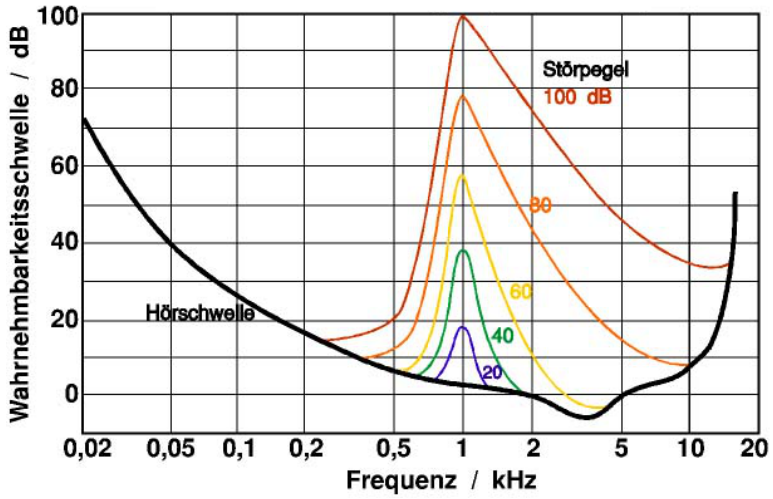

Abbildung 4: Anhebung der Hörschwelle bei Anwesenheit eines Störsignals (Sinuston 1 kHz), Quelle: adaptiert von
`http://de.wikipedia.org/wiki/Datei:Akustik_Mithoerschwelle2.JPG`

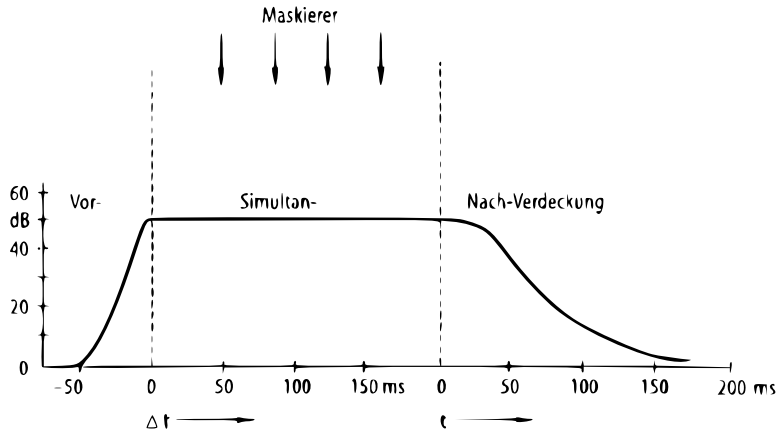

Abbildung 5: Maskierung im Zeitbreich [1, adaptiert von S. 78]

raum um 50 dB an. Nach und vor dem Maskierer ist die Hörschwelle eine Zeit lang angehoben. Diese Anhebung nimmt mit Abstand zum Störsignal ab, bis die Ruhehörschwelle wieder erreicht ist.

3 MP3

In diesem Kapitel wird das verlustbehaftete Audiokompressionsverfahren MP3 im Detail beschrieben und seine Funktionsweise erläutert. MP3 macht sich die im zweiten Kapitel genannten psychoakustischen Effekte zu Nutzte, um die Größe einer Audiodatei mit einem nicht bzw. kaum wahrnehmbaren

Qualitätsunterschied um ein vielfaches zu reduzieren. Es wird eine blockweise Spektralanalyse des Signals durchgeführt, in der festgestellt wird, welche Frequenzen nicht wahrgenommen und damit gelöscht werden können. Anschließend wird das Signal mit so vielen Bits quantisiert, dass das Quantisierungsrauschen möglichst nicht mehr zu hören ist. Die quantisierten Daten werden mithilfe der Huffman-Kodierung entropiekodiert und die Blöcke werden, zusammen mit einem Header, in die Datei geschrieben oder als Datenstrom verschickt.

3.1 MP3-Kodierer

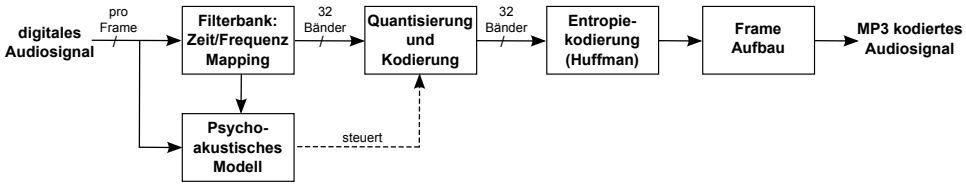

Abbildung 6: MP3 Kodiervorgang

Der MP3-Kodierer zerteilt das Eingangssignal in Blöcke fester Größe (ca. 27 ms), sogenannte Frames, welche dann eine Reihe von Verarbeitungsschritten durchlaufen. Abbildung 6 stellt diese Schritte übersichtlich dar. Die Filterbank überführt das Signal in den Frequenzbereich und zerteilt den Frame wiederum in 32 Subbänder. Die Subbänder durchlaufen eine Spektralanalyse (psychoakustisches Modell), bei der festgestellt wird, welche Frequenzen nicht wahrnehmbar sind. Nach dem Entfernen der überflüssigen Frequenzen werden die Subbänder quantisiert und kodiert. Das Quantisieren stellt die größte Datenreduktion dar. Eine weitere, jedoch geringer ausfallende, dafür verlustfreie Reduktion der Daten geschieht durch die Huffman-Kodierung der Datenblöcke der Frames.

3.1.1 Filterbank

Wie bereits in Kapitel 2 beschrieben, können sich nahegelegene Frequenzen gegenseitig maskieren. Daher wird das Frequenzspektrum in Subbänder unterteilt und einzeln quantisiert und kodiert. Nachdem der Frame mithilfe der Fouriertransformation in den Frequenzbereich überführt wurde, werden die Frequenzkoeffizienten mithilfe einer Reihe von Subbandfilter in 32, sich teilweise überlappende, Subbänder unterteilt. Diese werden nun vom psychoakustischen Modell und vom Quantisierer weiter verarbeitet.

3.1.2 Psychoakustisches Modell

Im zweite Kapitel wurde beschrieben, dass die psychoakustische Wahrnehmung begrenzt ist und dass Anteile des Signals durch Signalkomponenten,

sogenannte Maskierer, verdeckt werden können. Diese Eigenschaften unseres auditiven Systems finden in allen Audiokompressionsverfahren Anwendung, welche ein psychoakustisches Modell nutzten. Ziel des psychoakustischen Modells ist es, ein Signal-Maskierungs-Verhältnis (englisch: signal-to-mask ratio, SMR) für jedes der 32 Subbänder zu ermitteln. Anhand diesen Verhältnisses kann der Quantisierer entscheiden, welche Frequenzkomponenten nicht wahrnehmbar sind und verworfen werden können.

Die Berechnung des SMR geschieht in mehreren Schritten. Im ersten Schritt wird ein spektrales Profil des zu kodierenden Signals extrahiert. Die Schalldruckpegel der Subbänder, welche die Filterbank ausgibt, werden ermittelt. Da tonale und nicht tonale Komponenten die Maskierungs-Schwelle unterschiedlich beeinflussen, werden als zweites die Anwesenheit und Position dieser Komponenten bestimmt. Tonale Komponenten sind lokale Maxima welche zusätzlich folgende Bedingung erfüllen:

$$20 \cdot log_{10} \frac{|X_k|}{|X_{k+j}|} \geq 7$$

wobei X_k dem Schalldruckpegel der Frequenz an Stelle k des spektralen Profils entspricht und j abhängig von der Frequenz ist. Die identifizierten tonalen Maskierer werden vom entsprechenden Subband entfernt, um die nicht tonale Maskierungs-Schwelle zu finden. Sobald alle Maskierer identifiziert worden sind, werden alle Frequenzkomponenten deren Schalldruckpegel unterhalb der Maskierungs-Schwelle liegen entfernt. Zudem werden von zwei sehr nah beieinander liegenden Maskierern der Maskierer mit niedriger Amplitude entfernt. Mithilfe einer Spreizfunktion, welche die Frequenzmaskierung modelliert, wird der Einfluss der verbleibenden Maskierer festgestellt [7, S. 518].

Als letztes wird für jedes der 32 Subbänder das SMR berechnet, welches dem Quotient aus der maximalen Signalintensität innerhalb des Bandes und der minimalen Maskierungs-Intensität des Bandes entspricht. Der Quantisierer nutzt dieses Verhältnis für die Quantisierung und Kodieren des Signals.

3.1.3 Quantisierung

Die Quantisierung stellt die größte Reduktion der Daten dar. Mithilfe der Ausgabe des psychoakustischen Modells kann der Quantisierer in jedem Subband die Frequenzkomponenten entfernen, welche nicht wahrnehmbar sind. Die restlichen Frequenzkomponenten werden, wenn möglich, mit so wenigen Bits quantisiert, dass das Quantisierungsrauschen in jedem Subband gerade nicht mehr wahrnehmbar ist. Dies geschieht in einer Iterationsschleife, in der das Signal zuerst mit einer niedrigen Bitrate quantisiert wird und anschließend überprüft wird, ob das Quantisierungsrauschen unterhalb der Maskierungs-Schwelle liegt.

$$MNR = SNR - SMR$$

Sollte das Maskierungs-Rauschen-Verhältnis (englisch: mask-to-noise ratio, MNR) negativ sein, muss das Signal mit einer höheren Bitrate erneut quantisiert werden. Die Schleife wird so lange durchlaufen, bis das Quantisierungsrauschen nicht mehr wahrnehmbar ist.

3.1.4 Huffman Kodierung

Die quantisierten Frequenzkomponenten des Frames werden zusätzlich mithilfe einer von 18 Huffman-Tabellen entropiekodiert. [4, S. 35]

3.1.5 Format des Datenstroms

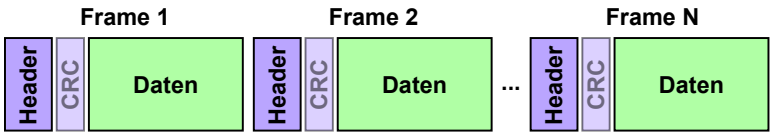

Abbildung 7: MP3 Datenstrom

Der MP3-Datenstrom besteht aus einzelnen Frames, welche aus einem Header, gegebenenfalls einer Fehlerchecksumme (englisch: cyclic redundancy check, CRC) und einem Datenblock bestehen (siehe Abbildung 7). Der Header besteht aus genau 32 Bit, welche wie folgt zusammengesetzt sind [4, S. 22–24]:

- **syncword** – '1111 1111 1111'
- **ID** – ein Bit, welches den verwendeten Algorithmus angibt. '1' für MPEG Audio, '0' ist reserviert
- **Layer** – 2 Bits, welche den verwendeten Layer nach folgender Tabelle angeben:
  '11' Layer I
  '10' Layer II
  '01' Layer III
  '00' reserviert
  Bei MP3 ist der Layer immer '01'
- **protection_bit** – ein Bit, welches angibt, ob dem Frame Redundanz hinzugefügt wurde, um Fehlererkennung und Fehlerverdeckung zu ermöglichen. '1' falls keine Redundanz hinzugefügt wurde, '0' falls Redundanz hinzugefügt wurde.
- **bit_rate_index** – 4 Bits, welche die Bitrate angeben. Die Bitrate bezieht sich auf die Summe der Kanäle. Der Wert ist zusätzlich abhängig

vom Layer. Die Beziehung von Bitrate-Index und Layer sind in Tabelle 1 dargestellt.

	Bitrate		
bit_rate_index	Layer I	Layer II	Layer III
'0000'	free format	free format	free format
'0001'	32 kbps	32 kbps	32 kbps
'0010'	64 kbps	48 kbps	40 kbps
'0011'	96 kbps	56 kbps	48 kbps
'0100'	128 kbps	64 kbps	56 kbps
'0101'	160 kbps	80 kbps	64 kbps
'0110'	192 kbps	96 kbps	80 kbps
'0111'	224 kbps	112 kbps	96 kbps
'1000'	256 kbps	128 kbps	112 kbps
'1001'	288 kbps	160 kbps	128 kbps
'1010'	320 kbps	192 kbps	160 kbps
'1011'	352 kbps	224 kbps	192 kbps
'1100'	384 kbps	256 kbps	224 kbps
'1101'	416 kbps	320 kbps	256 kbps
'1110'	448 kbps	384 kbps	320 kbps

Tabelle 1: Bitrate

- **sampling_frequency** – 2 Bits, welche die Abtastrate nach folgender Tabelle angeben:
  '00' 44.1 kHz
  '01' 48.0 kHz
  '10' 32.0 kHz
  '11' reserviert

- **padding_bit** – ein Bit, welches angibt, '1' dass dem Frame zusätzliche Daten angefügt wurden, um die mittlere Bitrate an die Abtastrate anzupassen oder '0' dass dem Frame keine zusätzlichen Daten angefügt wurden.

- **private_bit** – ein Bit, welches der persönlichen Nutzung frei steht. Es wird auch in Zukunft nicht durch ISO genutzt.

- **mode** – 2 Bit, welche den Kanalmodus nach folgender Tabelle angeben:
  '00' Stereo
  '01' Joint-Stereo
  '10' Zweikanal
  '11' Einzelkanal

- **mode_extension** – 2 Bits, welche anhand folgender Tabelle angeben, welche Joint-Stereo Kodierungsmethode angewendet wurde:

	Intensitäts-Stereo	Mid-Side-Stereo
'00'	off	off
'01'	on	off
'10'	off	on
'11'	on	on

- **copyright** – ein Bit, welches angibt, ob der kodierte Bitstream '1' durch das Urheberrecht geschützt oder '0' nicht geschützt ist.

- **original/home** – ein Bit, welches angibt, ob der Bitstream '0' eine Kopie ist oder '1' das Original.

- **emphasis** – 2 Bits, welche nach folgender Tabelle angeben, welche Emphase verwendet wurde:

'00'	keine Emphase
'01'	50/15 microsek. Emphase
'10'	reserviert
'11'	CCITT J.17

Falls das **protection_bit** auf '0' gesetzt ist, enthält der Bitstream zwischen Header und Datenblock noch einen 16 Bit Paritätscheck. Dieser wird zur Fehlererkennung innerhalb des Frames genutzt [4, S. 33].

3.1.6 Kodierung von Stereosignalen

Anstatt beide Kanäle einzeln zu kodieren, erlaubt der Standard die Mid-Side-Kodierung und die Intensitäts-Kodierung. Beide Stereo-Kodierungs-Verfahren können zur gleichen Zeit für verschiedene Frequenzenbereiche genutzt werden [7, S. 531].

Die Intensitäts-Kodierung nutzt die Tatsache, dass in höheren Frequenzbereichen das Stereo-Signal durch einen einzelnen Kanal plus einer Richtungsinformation kodiert werden kann [7, S. 531]. Diese Methode ist jedoch verlustbehaftet und sollte nur bei niedrigen Bitraten genutzt werden.

Die Mid-Side-Kodierung nutzt die Korrelation zwischen rechtem Kanal R und linken Kanal L, indem ein Mid- und ein Side-Kanal berechnet wird:

$$Mid = \frac{L + R}{2}$$

$$Side = \frac{L - R}{2}$$

Danach werden Mid- und Side-Kanal einzeln kodiert. Bei hoher Korrelation der ursprünglichen Kanäle, ist der Side-Kanal wesentlich weniger komplex

als die ursprünglichen Kanäle. Daher kann dieser mit einer niedrigeren Bitrate kodiert werden. Die ursprünglichen Kanäle können verlustfrei rekonstruiert werden:

$$R = Mid - Side$$

$$L = Mid + Side$$

3.2 Der MP3-Dekodierer

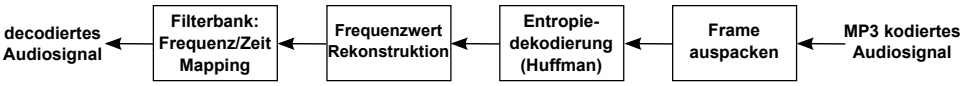

Abbildung 8: MP3 Dekodiervorgang

Eine wichtige Anforderung für multimediale Kompressionsverfahren ist es, eine schnelle und ressourcensparende Dekodierung zu ermöglichen, da die Dateien in der Regel wesentlich öfter dekodiert werden und die Dekodierung auch auf leistungsschwachen Geräten funktionieren sollte. Die Kodierung ist, aufgrund des psychoakustischen Modells und der iterativen Suche nach der richtigen Bitrate, speicher- und rechenaufwendig. Jedoch fallen genau diese Komponenten bei der Dekodierung weg. Der Dekodierer muss lediglich die restlichen Schritte des Kodierungsvorgangs in umgekehrter Reihenfolge durchlaufen, um das verlustbehaftete Zeitsignal zu rekonstruieren. Der Dekodierer erhält die Frames des MP3-kodierte Audiosignals und entpackt den Frame, dekodiert die Daten mithilfe der entsprechenden Huffman-Tabelle und rekonstruiert die Frequenzwerte, indem eine inverse Quantisierung durchgeführt wird. Als letztes wird das Frequenzspektrum wieder in den Zeitbereich überführt. Abbildung 8 zeigt die vom Dekodierer auszuführenden Schritte.

4 Ogg Vorbis

Ogg Vorbis besteht aus dem Containerformat Ogg und dem Audiokompressionsverfahren Vorbis. Beide wurden von der Xiph.Org Foundation entwickelt, sind patentfrei und können somit ohne Lizenzabgaben verwendet werden. In diesem Kapitel geht es vorrangig um das Audiokompressionsverfahren Vorbis, welches 1998 als Alternative zu proprietären Audiokompressionsverfahren entwickelt wurde.

Das Containerformat Ogg wird am Ende des Kapitels kurz erläutert.

4.1 Vorbis-Kodierer

Der Kodierprozess wird in Abbildung 9 dargestellt. Der Vorbis-Kodierer zerteil den Datenstrom in Frames, welche einzeln kodiert werden. Zuerst wird

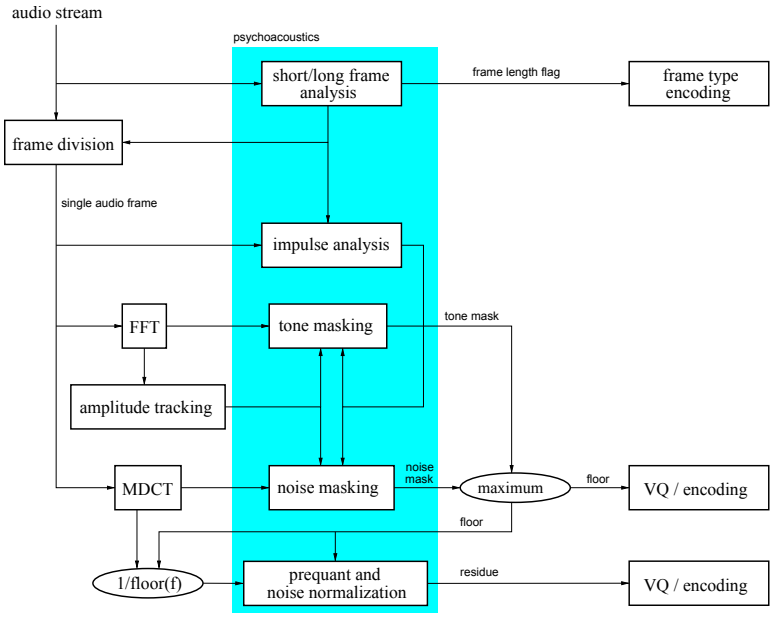

Abbildung 9: Vorbis Kodierer [5, S. 4]

ermittelt, ob das Signal nicht-tonale Ereignisse enthält. Falls ja werden für diese Abschnitte kürzere Frames verwendet, da diese Stellen im dekodierte Signal sonst hörbar verzerrt wären. Dann wird der Frame in den Frequenzbereich überführt und seine psychoakustischen Merkmale extrahiert. Mithilfe dieser Merkmale wird eine niedrig aufgelöste Version des Frequenzspektrums erstellt, der sogenannte Floor-Vektor. Das Spektrum wird mithilfe des Floor-Vektors normalisiert. Als letztes werden der Floor-Vektor und das normalisierte Spektrum vektorquantisiert [2].

4.1.1 Headertypen

Die kontroverseste Designentscheidung in Vorbis ist, dass das gesamte Wahrscheinlichkeitsmodell des Codecs, die Codebooks für Huffman-Kodierung und Vektorquantisierung, in den Bitstream-Header gepackt werden, zusammen mit umfangreichen Codec-Setup-Parametern. Daher ist es nicht möglich bei einem beliebigen Frame den Dekodierprozess zu starten, ohne vorher bereits die Codec-Setup-Header empfangen zu haben [2, S. 5].

Abbildung 10: Vorbis Datenaufbau

Vorbis nutzt drei verschieden Header [2, S. 9]:

- **Identification-Header** – identifiziert den Bitstream als Vorbis, gibt die Vorbis Version an und enthält einfache Audio-Eigenschaften wie Abtastrate und Anzahl der Kanäle.
- **Comment-Header** – enthält benutzerdefinierte Kommentare (englisch: tags) und einen String, der angibt, welche Applikation bzw. Bibliothek diesen Bitstream erzeugt hat.
- **Setup-Header** – enthält umfangreiche Codec-Setup-Parameter sowie die Huffman und VQ Codebooks

Nach den drei Headern können beliebig viele Audio-Frames gespeichert sein bzw. übertragen werden (siehe Abbildung 10).

4.1.2 Floor

Vorbis kodiert einen spektralen Floor-Vektor für jeden Kanal. Dieser Vektor repräsentiert eine niedrige Auflösung des Frequenzspektrums des gegebenen Kanals im aktuellen Audio-Frame [2, S. 8]. Der Floor-Vektor wird genutzt, um das Frequenzspektrum zu normalisieren, sodass eine direkte lineare Quantisierung des normalisierten Frequenzspektrums ausreicht [5, S. 4].

4.1.3 Residue

Das spektrale Residue-Vektor spiegelt die feine Struktur des Frequenzspektrums wieder [2, S. 8]. Es entspricht dem, mit den Floor-Vektor normalisierten Frequenzspektrum und wird berechnet, indem man das ursprünglichen Frequenzspektrum durch die Floor-Kurve teilt [5, S. 4]. Die Floor-Kurve wird durch lineare Interpolation des Floor-Vektors erzeugt.

Das ursprüngliche Frequenzspektrum kann durch das Skalarprodukt der Floor-Kurve und dem Residue-Vektor zurückgewonnen werden.

4.1.4 Codebooks

Vorbis hat im Gegensatz zu MP3 kein statisches Wahrscheinlichkeitsmodell und muss daher die entsprechenden Huffman-Tabellen mit übertragen. Diese werden in den Codebooks gespeichert, welche wiederum im Setup-Header gespeichert sind.

4.2 Vorbis-Dekodierer

Der Dekodierprozess wird in Abbildung 11 dargestellt. Dem Dekodierer müssen alle Codec-Setup-Header vorliegen, bevor er den Dekodierprozess starten kann. Der Floor-Vektor wird aus dem aktuellen Frame gelesen und

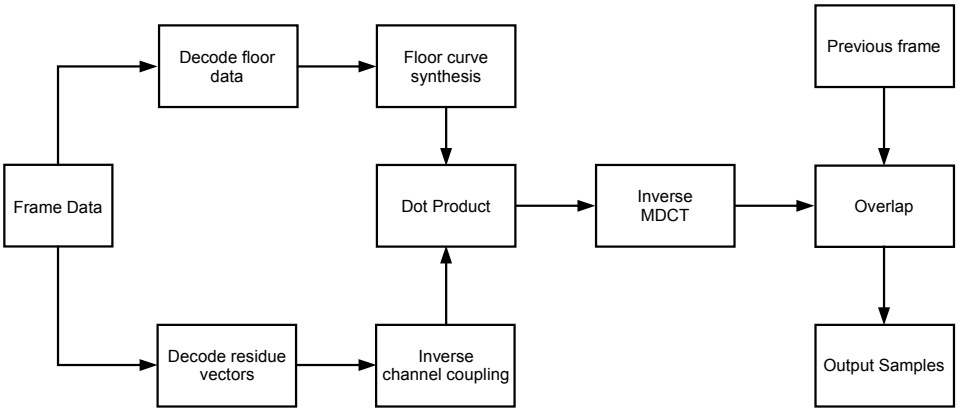

Abbildung 11: Vorbis Dekodierer [3, S. 13]

die Floor-Kurve wird durch lineare Interpolation brechnet. Der Residue-Vektor wird ebenfalls aus dem Frame gelesen und eine inverse Kanalkopplung durchgeführt. Anschließend wird das Skalarprodukt aus dem resultierenden Residue-Vektor und der Floor-Kurve berechnet. Daraus ergibt sich das Frequenzspektrum des aktuellen Frames, welches über die inverse diskrete Kosinustransformation in den Zeitbereich überführt wird. Um den aktuellen Frame zu dekodieren, benötigt der Dekodierer zusätzlich die zeitliche Darstellung des vorherigen Frame. Die beiden zeitlichen Signale werden überlagert und das resultierende Signal ausgegeben [2].

4.3 Ogg-Container

Vorbis ist ein reines Audiokompressionsverfahren, welches sich nicht um Framing, Synchronisierung, Positionierung und Fehlererkennung kümmert. Daher benötigt es ein Containerformat, welches diese Aufgaben übernimmt. Grundsätzlich können eine Reihe von verschiedenen Containerformaten verwendet werden [2, S. 5]. In der Praxis wird jedoch meist Ogg verwendet.

5 Vergleich: MP3 und Ogg Vorbis

Im direkten Vergleich der beiden vorgestellten verlustbehafteten Audiokompressionsverfahren MP3 und Ogg Vorbis stellt man wesentlichen Unterschiede bzw. Vor- und Nachteile fest.

Da Ogg Vorbis (patent-)frei ist, müssen für dessen Verwendung keine Lizenzgebühren gezahlt werden, sowie es bei MP3 der Fall ist. Dafür ist MP3 Teil der ISO Normen MPEG-1 und MPEG-2, ist weiter verbreitet als Ogg Vorbis und wird damit von wesentlich mehr Hard- und Software unterstützt.

MP3 wurde für Audiodaten mit maximal zwei Kanälen und einer geringen Bandbreite von wählbaren Abtastraten (siehe Tabelle 2) entworfen.

	MP3	Ogg Vorbis
Patentbehaftet	− ja	+ nein
Verbreitung Unterstützung	+	−
Bitraten	− 32–320 kb/s	+ 32–500 kb/s
Abtastfrequenz	− 32–48 kHz	+ 8–192 kHz
Kanäle	− 1–2	+ 1–255
Kodierer Speicherverbrauch	+	−
Dekodierer Komplexität	−	+
Soundqualität (niedrige Bitraten)	−	+

Tabelle 2: Vor- und Nachteile von MP3 und Ogg Vorbis [4][2]

Ogg Vorbis hingegen ist weniger restriktiv und gibt dem Benutzer wesentlich mehr Flexibilität, da man bis zu 255 Kanäle nutzen kann und Abtastraten von 8–192 kHz wählen kann.

Bei niedrigen Bitraten ist die Soundqualität bei Ogg Vorbis besser als bei MP3, da das Quantisierungsrauschen bei MP3 dann deutlich zu hören ist, wobei Ogg Vorbis immer noch gute Ergebnisse liefert. Bei hohen Bitraten liefern beide Verfahren vergleichbare gute Ergebnisse.

Tabelle 2 gibt einen Überblick über die wesentlichen Unterschiede und fasst die Vor- und Nachteile zusammen.

Zusammenfassung

Die verlustbehafteten Audiokompressionsverfahren MP3 und Ogg Vorbis nutzen die Einschränkungen unserer auditiven Wahrnehmung um die Audiodaten zu komprimieren. Sie erkennen welche Frequenzen nicht wahrnehmbar sind und entfernen diese. Je mehr die Audiodaten komprimiert werden, desto eher kann man Verzerrungen wahrnehmen, welche durch die Quantisierung entstehen. Diese Verzerrungen werden Quantisierungsrauschen genannt und Ziel der Verfahren ist es, das Rauschen so niedrig wie möglich zu halten. Die Verfahren verfolgen dabei verschiedene Ideen und liefern teils qualitativ stark unterscheidbare Ergebnisse. So erzeugt Ogg Vorbis bei niedrigen Bitraten subjektiv deutlich bessere Audiodateien als MP3. MP3 ist heute das am weitesten verbreitete Audiokompressionsverfahren obwohl es viele modernere Verfahren wie zum Beispiel Ogg Vorbis gibt, welche wesentlich flexibler und nicht durch Patente behaftet sind.

Literatur

[1] H. Fastl and E. Zwicker. *Psychoacoustics*. Springer series in information sciences. Springer, 2007.

[2] Xiph.Org Foundation. Vorbis I specification, 2012. Verfügbar unter `http://xiph.org/vorbis/doc/Vorbis_I_spec.pdf`.

[3] Robert Harsan, Thomas Loch, and Jocelyn Ratac. VP Wissenschaftliche Arbeitstechniken und Präsentation –Vorbis. `http://www.cosy.sbg.ac.at/~held/teaching/wiss_arbeiten/slides_09-10/Vorbis.pdf`, 2010.

[4] ISO/IEC. 11172-3: Coding of moving pictures and associated audio for digital storage media at up to about 1.5 mbit/s part 3 audio, 1991.

[5] Christopher Montgomery and Jean-Marc Valin. Improved noise weighting in celp coding of speech - applying the vorbis psychoacoustic model to speex. In *Audio Engineering Society Convention 120*, 5 2006.

[6] B.C.J. Moore. *An Introduction to the Psychology of Hearing*. Academic Press, 2003.

[7] K. Sayood. *Introduction to Data Compression*. The Morgan Kaufmann Series in Multimedia Information and Systems. Elsevier Science, 2005.