Konzeption und Implementierung eines MATLAB-basierten MPEG-Audiokodierers unter Berücksichtigung didaktischer Aspekte

Per Loenicker[1], Dieter Leckschat[2], Christian Epe[3]

[1] *FH Düsseldorf, 40225 Düsseldorf, E-Mail: perhanloe@hotmail.de*
[2] *FH Düsseldorf, 40225 Düsseldorf, E-Mail: dieter.leckschat@fh-duesseldorf.de*
[3] *FH Düsseldorf, 40225 Düsseldorf, E-Mail: christian.epe@fh-duesseldorf.de*

Einleitung

Die Entwicklung des Audiokodierers *MPEG-1 Layer III* (mp3) Anfang der 90er Jahre hat bis heute die Art und Weise wie Tonaufnahmen konsumiert werden so nachhaltig verändert, dass man von einer Revolution sprechen kann. Digitale Musikvertriebe wie iTunes oder Napster wären ohne verlustbehaftete Datenkompression nicht denkbar und die Musikindustrie war und ist als Folge gezwungen, sich zum Teil neu zu erfinden.

Doch auch aus technischer Sicht ist der Algorithmus sehr interessant. Allen voran die Erkenntnisse aus dem Forschungsfeld der Psychoakustik, aber auch Themen der digitalen Signalverarbeitung und der Informatik werden hier zusammengeführt.

Das Prinzip der verlustbehafteten Datenkompression, durch Ausnutzung von Verdeckungseffekten bestimmte Signalanteile nicht zu übertragen, ist vielen bekannt. Dies ist aber nur die halbe Wahrheit, der tatsächliche Prozess ist weitaus komplexer.

Die MATLAB-basierte Experimentierumgebung *MPER-3* bietet Studierenden die Möglichkeit, die komplexen Abläufe innerhalb des Kodierers zu verstehen. Wichtige berechnete Werte des mp3-Standards sind an Graphen ablesbar. Außerdem können Parameter verändert und verschiedene Aufgaben bearbeitet werden, die sich näher mit den vielen Teilprozessen des Kodierers beschäftigen. Ziel der Anwendung ist es, dem Nutzer ein tiefgreifendes Verständnis von der verlustbehafteten Audiokodierung zu vermitteln.

MPER-3 richtet sich vorzugsweise an Studierende der Ingenieurswissenschaften mit Grundkenntnissen in digitaler Signalverarbeitung.

Konzeption und Implementierung von MPER-3

Bei der Konzeption von MPER-3 stand die Vermittlung der verwendeten Algorithmen im Vordergrund, die Qualität des Kodierers spielte dabei eine untergeordnete Rolle. Daraus ergaben sich zwei Vorüberlegungen. Zum einen sollten die Entwicklungsumgebung und die Programmiersprache möglichst benutzerfreundlich sein und zum anderen sollte der Kodierer lediglich über die nötigsten Funktionen verfügen, um die Komplexität möglichst gering zu halten und auf das Wesentliche zu reduzieren.

So fiel die Wahl auf MATLAB, da hier das Kriterium der Benutzerfreundlichkeit zutrifft und es darüber hinaus eine Menge Funktionen mitbringt, die das Analysieren der Werte und damit den Lernerfolg fördern. Allerdings bringt MATLAB auch einen Nachteil mit sich, die Berechnung des Algorithmus' erfolgt vergleichsweise langsam. Dies ist allerdings aus den oben genannten Gründen vertretbar. Weitere Einschränkungen gegenüber der MPEG-1 Layer III Vorlage aus der ISO/IEC 11172-3 [1], die zugunsten der Performance und Reduzierung der Komplexität in MPER-3 nicht enthalten sind:

1. der Encoder arbeitet nur einkanalig (mono)
2. der Encoder arbeitet nur mit langen Analysefenstern
3. die optionalen Funktionen ‚Error Check' und ‚Preemphasis' werden nicht unterstützt

Die Implementierung orientierte sich stark an der ISO/IEC 11172-3. Sämtliche Formeln zur Berechnung des Algorithmus sind diesem Standard entnommen. Um die praktischen Versuchsaufgaben in sinnvolle Abschnitte zu gliedern, ist MPER-3 in drei Unterprogramme unterteilt, die sich am allgemeinen Aufbau eines verlustbehafteten Audiokodierers orientieren.

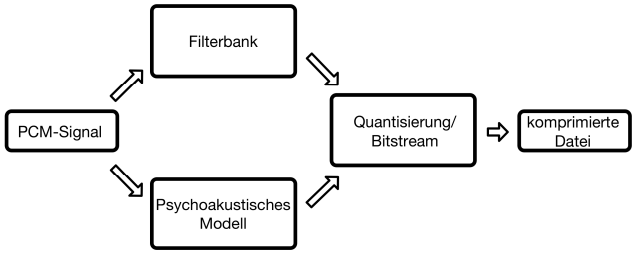

Abbildung 1: Allgemeiner Aufbau eines verlustbehafteten Audiokodierers.

Die Filterbank bildet den ersten Part (MPER-3 Pt1) und unterteilt das breitbandige PCM-Signal in eine Reihe von Subbändern. Parallel dazu findet im zweiten Part (MPER-3 Pt2) die Berechnung der Maskierungsschwelle mit Hilfe des in der ISO/IEC 11172-3 vorgeschlagenen psychoakustischen Modells statt. Im letzten Part (MPER-3 Pt3) werden schließlich die Subbänder unter Einbeziehung der Maskierungsschwelle quantisiert sowie ein Datenstrom zur Speicherung in einer mp3-Datei generiert.

Begleitend zum Quellcode in MATLAB gibt es eine Versuchsanleitung die in zwei Abschnitte gegliedert ist. In einem theoretischen Teil werden Grundlagen der mp3-Kodierung vermittelt sowie wichtige Formeln zur

Bearbeitung der Aufgaben eingeführt. Anschließend gibt es zu jedem Programmpart Aufgaben zu bearbeiten, bevorzugt direkt im Quellcode. Einige dieser Aufgaben können auch mit den jeweils zugehörigen MATLAB Apps mit Hilfe einer grafischen Benutzeroberfläche durchgeführt werden.

Die einzelnen Parts im Detail

MPER3 Pt1

Der erste Part behandelt die Filterbank des Encoders. Der mp3-Algorithmus besitzt eine zweistufige hybride Filterbank, in der das Ausgangsmaterial in bis zu 576 Spektrallinien unterteilt wird. Die erste Stufe, die *Pseudo Quadratur Mirror Filterbank* (PQMF), teilt das Spektrum in 32 Subbänder und die anschließende *Modifizierte Diskrete Cosinus Transformation* (MDCT) erzeugt jeweils 18 weitere Spektrallinien pro Subband.

Interessanterweise kann gezeigt werden, dass beide Verfahren mathematisch identisch sind [2]. Daher geht es in MPER-3 Pt1 lediglich um die PQMF, da diese Variante besser nachzuvollziehen ist.

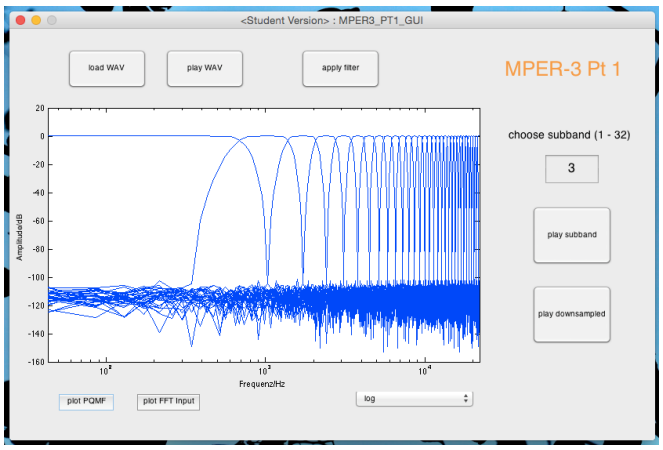

Abbildung 2: MPER-3 Pt1 GUI mit logarithmisch aufgetragenem Amplitudenverlauf der PQMF.

Die Aufgaben behandeln die Modulation eines Prototyp FIR Filters zur Erzeugung der PQMF, die anschließende Filterung des Signals und das Downsampling der einzelnen Subbänder zur Erhaltung der Datenrate. Um das Downsampling nachvollziehen zu können, kann nach der Filterung in sämtliche Subbänder jeweils vor und nach dem Downsampling hineingehört werden. Dadurch erhält man ein äußerst eindrucksvolles Bild von Aliaseffekten.

MPER-3 Pt2

Im zweiten Teil geht es um die Berechnung des *Signal to Mask Ratio* (SMR), welches zur Quantisierung der Subbänder benötigt wird. Dabei handelt es sich um das Verhältnis zwischen der berechneten *Maskierungsschwelle* und der Signalleistung pro *Bark-Band[z]*. Maskierungsschwelle und Barkskala[z] sind Gegenstand der Psychoakustik [3]. Dabei wird das hörbare Frequenzspektrum in einzelne Gruppen, die Bark-Bänder[z] aufgeteilt. Innerhalb einer solchen Gruppe kommt es zu spektralen Verdeckungseffekten. Diese treten auf, wenn ein Spektralanteil um einen bestimmten Wert lauter ist als ein anderer. Man spricht dann vom Maskierer, der andere Spektralanteile verdeckt (maskiert), wodurch diese nicht mehr wahrnehmbar sind. Die Schwelle, unterhalb derer ein Spektralanteil vom Maskierer verdeckt wird, heißt Maskierungsschwelle.

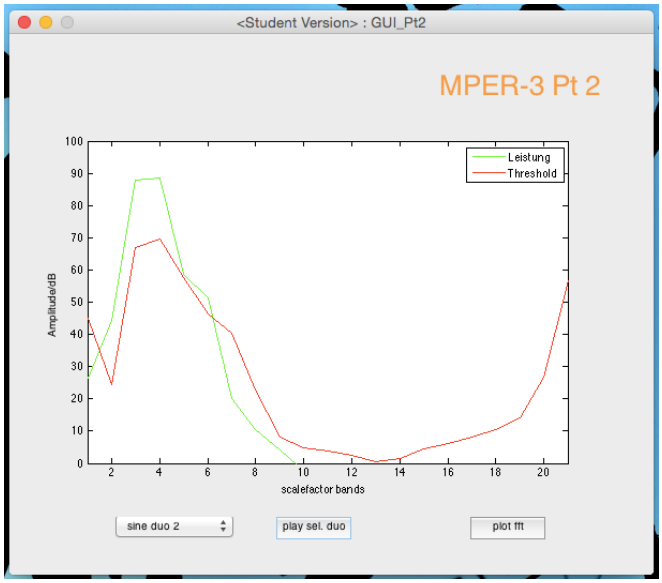

Abbildung 3: MPER-3 Pt2 GUI mit der Signalleistung pro Bark-Band (grün) und der Maskierungsschwelle (rot) für zwei Sinustöne mit unterschiedlicher Frequenz und Amplitude.

Das Ziel dieses Abschnitts ist es, die Berechnung der Maskierungsschwelle im Detail nachzuvollziehen, indem die einzelnen Schritte dazu erläutert werden. Zusätzlich soll der letzte Schritt, die Einbeziehung der Ruhehörschwelle, selbst implementiert werden.

Um das Phänomen der Maskierung kennen zu lernen, kann mit Hilfe der GUI ein kleines Experiment durchgeführt werden: in fünf Tonbeispielen sind zwei Sinustöne zu hören, wobei der zweite Ton immer leiser wird, bis er im letzten Beispiel komplett verschwunden ist. Wahrnehmbar ist er allerdings schon vorher nicht mehr, nämlich dann, wenn die Signalleistung des zweiten Sinustons unterhalb der Maskierungsschwelle gesunken ist. Zum Vergleich kann auch eine FFT der einzelnen Tonbeispiele angezeigt werden.

MPER-3 Pt3

Im dritten und letzten Teil geht es um die Quantisierung der Spektrallinien unter Berücksichtigung des SMR und die anschließende Erzeugung eines Datenstroms. Diese Prozesse sind sehr komplex, sie im Detail zu kennen ist für ein Verständnis des Prinzips der verlustbehafteten Datenkompression nicht unbedingt notwendig. Es reicht zu wissen, dass die zur Verfügung stehenden Bits in einem iterativen Prozess zur Quantisierung der Spektrallinien genutzt werden. Dabei spielt der SMR eine Rolle, je höher dieser Wert ist, desto mehr Bits werden benötigt um das Quantisierungsrauschen unterhalb der Maskierungsschwelle zu halten. Wenn alle Bits verteilt sind wird geprüft, ob noch Bänder über hörbare Quantisierungsverzerrungen verfügen. Ist dies der Fall, werden diese Bänder mit einem Skalierungsfaktor versehen und die Quantisierung wird

wiederholt. Nach der Quantisierung wird ein Datenstrom erzeugt, der die Audiodaten in Form einzelner Frames enthält.

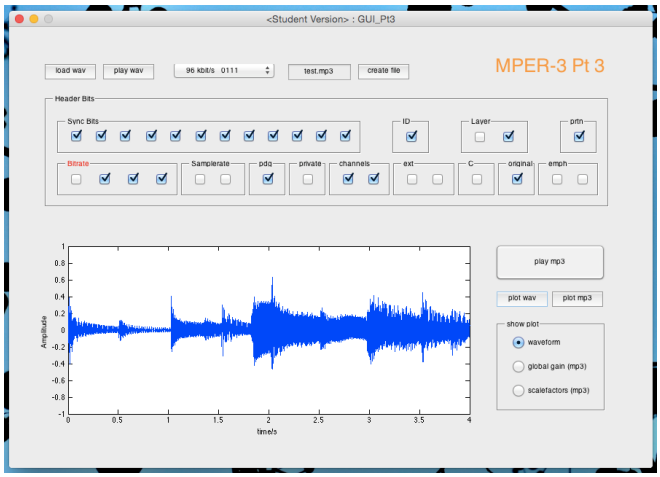

Abbildung 4: MPER-3 Pt3 GUI mit Waveform des Eingangssignals.

Eine Aufgabe dazu beschäftigt sich mit der Quantisierung, in einer anderen Aufgabe soll das aus Nullen und Einsen zusammengesetzte Array in eine Datei geschrieben werden.

Abbildung 5: mp3-Frame.

Zusätzlich soll untersucht werden, wie sich verschiedene Datenraten auf den Parameter *global gain* und auf die *Skalierungsfaktoren* der einzelnen Bänder auswirken.

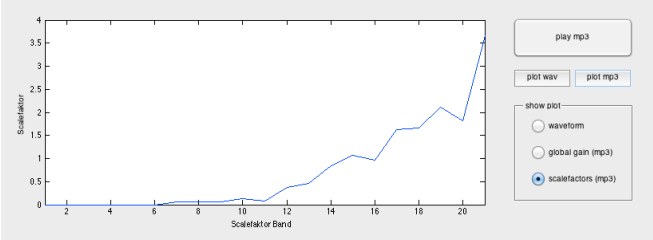

Abbildung 6: mittlere Skalierungsfaktoren der einzelnen Bark-Bänder einer fertig gerechneten mp3-Datei.

Die grafische Benutzeroberfläche von MPER-3 Pt3 bietet zudem die Möglichkeit, die 32 Header-Bits der einzelnen Frames von Hand zu setzen (siehe Abb. 4). Dadurch soll der Benutzer ein Gefühl für die Sensibilität digitaler Datenformate bekommen. Wird beispielsweise eines der Sync-Bits auf 0 gesetzt, kommt am Ende nur Chaos heraus. Oder werden die vier Bits für die Bitrate nicht richtig gesetzt, wird das mp3-File unter anderem über regelmäßige Lücken verfügen, weil der Decoder die Sync-Bits an anderer Stelle erwartet (siehe Abb. 7).

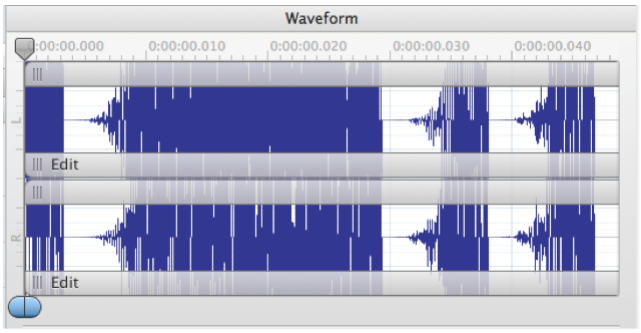

Abbildung 7: Waveform einer mp3-Datei, bei der die Header-Bits für die Datenrate falsch gesetzt wurden. Der Decoder erwartet das Sync-Bitmuster zu einem anderen Zeitpunkt und wartet dann so lange, bis es kommt. So entstehen die Lücken im Audio.

Fazit

Auch wenn der mp3-Standard inzwischen etwas in die Jahre gekommen ist und weiterentwickelte Formate wie *Advanced Audio Coding* (AAC) zu besseren Resultaten kommen, ist das Grundprinzip, Subbänder unter Berücksichtigung psychoakustischer Modelle neu zu quantisieren, erhalten geblieben. Aufgrund der im Gegensatz zu neueren Formaten geringeren Komplexität eignet sich die mp3 hervorragend als Einstieg in die Welt der Datenkompression.

MPER-3 bietet fortgeschrittenen Studierenden der Ingenieurswissenschaften einen praktischen Einblick in die Funktionsweise dieses verlustbehafteten Kopressionsalgorithmus in Form eines umfangreichen Laborversuchs. Mit Hilfe der grafischen Benutzeroberfläche können auch ohne große Vorbereitung kleinere Experimente durchgeführt werden.

Literatur

[1] International Standard ISO/IEC 11172-3: Information technology – Coding of moving pictures and associated audio for digital storage media at up to 1,5 Mbit/s – Part 3: Audio, 1993

[2] Bosi, M.; Goldberg, R. E.: Introduction to Digital Audio Coding and Standards. Kluwer Academic Publishers, Dordrecht, 2003

[3] Weinzierl, S. (Hrsg.): Handbuch der Audiotechnik. Springer-Verlag, Berlin Heidelberg, 2008