

# Vermeidung von Generationseffekten in der Audiocodierung

**Dissertation**

zur

Erlangung des Doktorgrades (Dr. rer. nat.)

der

Mathematisch-Naturwissenschaftlichen Fakultät

der

Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von

Frank Kurth

aus

Bad Honnef am Rhein

Bonn, 9. März 1999



# Vermeidung von Generationseffekten in der Audiocodierung

**Dissertation**

zur

Erlangung des Doktorgrades (Dr. rer. nat.)

der

Mathematisch-Naturwissenschaftlichen Fakultät

der

Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von

Frank Kurth

aus

Bad Honnef am Rhein

Bonn, 9. März 1999

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Rheinischen Friedrich-Wilhelms-Universität Bonn

1. Referent: Prof. Dr. Michael Clausen
2. Referent: Prof. Dr. Joachim M. Buhmann

Tag der Promotion:

# Danksagung

Die vorliegende Arbeit ist im Projektbereich *Kommunikationssysteme und Algorithmen* am Institut für Informatik V der Universität Bonn entstanden. Sie fügt sich in die Thematik des von der Deutschen Forschungsgemeinschaft (DFG) geförderten Projekts *Inhaltsbasierte Indexierung, Retrieval und Kompression von Daten in digitalen Musikbibliotheken* im Rahmen des Schwerpunktprogramms *Verteilte Verarbeitung und Vermittlung digitaler Dokumente* ein und behandelt insbesondere den Kompressionsaspekt.

Ich möchte mich bei den zahlreichen am Gelingen dieser Arbeit direkt oder indirekt beteiligten Personen an dieser Stelle herzlich bedanken, insbesondere

- bei Herrn Prof. Dr. Michael Clausen, der mir nicht nur als Leiter der Arbeitsgruppe und wissenschaftlicher Betreuer eine unschätzbare Unterstützung hat zuteil werden lassen, und ohne dessen Einsatz diese Arbeit gar nicht erst entstanden wäre,
- bei Herrn Prof. Dr. Joachim Buhmann für seine hilfreichen Anmerkungen und die Bereitschaft, trotz einer Vielzahl anderweitiger Verpflichtungen als Referent dieser Arbeit tätig zu werden,
- bei allen Kollegen und Freunden, die stets für ein besonders angenehmes Arbeitsklima gesorgt haben, insbesondere bei Frau Vlora Arifi und den Herren Heiko Goeman, Patrick Hamilton, Dirk Meyer und Meinard Müller,
- bei Frau Esther Neus für vieles,
- bei meinen Eltern, ohne deren Unterstützung die Anfertigung dieser Arbeit nicht möglich gewesen wäre,
- bei allen Diplomanden, insbesondere für das ständige hartnäckige Nachfragen,
- bei allen Testhörern und Teilnehmern der Projektgruppen *Audiosignalverarbeitung* und *Digitale Musikbibliotheken* und
- bei Donald E. Knuth und Leslie Lamport dafür, daß sie mit T<sub>E</sub>X und L<sup>A</sup>T<sub>E</sub>X die Möglichkeit geschaffen haben, Texte aller Art in herausragender Qualität zu setzen.

Bonn, den 29. April 1999



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Grundlagen der Audiokompression</b>	<b>5</b>
2.1	Mathematische und signaltheoretische Grundlagen . . . . .	5
2.2	Psychoakustik . . . . .	8
2.2.1	Maskierung und kritische Bänder . . . . .	9
2.2.2	Globale Maskierungsschwelle . . . . .	13
2.3	Audiokompression . . . . .	13
2.3.1	Kompression und Multimedia: MPEG Standards . . . . .	14
2.3.2	Forschungsaktivitäten und Trends . . . . .	16
<b>3</b>	<b>Digitale Generationseffekte</b>	<b>19</b>
3.1	Fehleranalyse am Codec-Modell . . . . .	20
3.1.1	Fehlerquellen . . . . .	21
3.1.2	Fehlquantisierung durch Aliasing . . . . .	22
3.2	Fallstudie: MPEG-1 Codec . . . . .	23
3.2.1	Subjektive Tests . . . . .	23
3.2.2	Auswirkung von Translationen . . . . .	27
3.2.3	Signal- und Parameterdegeneration . . . . .	30
3.2.4	Hörtests . . . . .	32
3.3	Lösungsansätze . . . . .	35
<b>4</b>	<b>Dateneinbettungsverfahren</b>	<b>37</b>
4.1	Grundlagen und Steganographie . . . . .	37
4.2	Fehler- und Einbittungsregionen . . . . .	42
4.3	Datenangepaßte Rückquantisierung . . . . .	47
4.4	Übergang zu diskreten Gittern quantisierter Werte . . . . .	52
4.4.1	Allgemeine Betrachtung . . . . .	52
4.4.2	Lineare Quantisierung . . . . .	53
4.5	Auswirkung von Skalierungen . . . . .	54
4.6	Fehlertolerante Einbettungsverfahren . . . . .	55
4.6.1	Arithmetik- und Rekonstruktionsfehler . . . . .	56
4.6.2	Direkte Biteinbettung und Arithmetische Codes . . . . .	58
4.6.3	Stabilität gegenüber sonstigen Fehlern . . . . .	60
4.7	Variable Einbettungsbreiten und Blockbildung . . . . .	61
4.7.1	Variable Bitraten und Einbettungsbreiten . . . . .	62

4.7.2	Erkennung einer Einbettung . . . . .	63
4.7.3	Blockbildung und -organisation . . . . .	64
<b>5</b>	<b>Codecs zur Vermeidung von Generationseffekten</b>	<b>67</b>
5.1	Das Codec-Modell . . . . .	67
5.2	Theoretische Überlegungen . . . . .	70
5.3	Codec ohne Generationseffekte . . . . .	74
5.3.1	Grundaufbau . . . . .	74
5.3.2	Direkteinbettungscodec . . . . .	75
5.3.3	Codec basierend auf Einbettungsregionen . . . . .	79
5.3.4	Implementierungsfragen . . . . .	79
5.3.4.1	Synchronisation . . . . .	80
5.3.4.2	Einbettungsregionen und Kompressionsraten . . . . .	80
5.3.4.3	Bitreservoirtechniken . . . . .	80
5.4	Mehrstufige Codecs . . . . .	81
<b>6</b>	<b>Codec-Realisierung: MPEG-1/Audio, Layer II</b>	<b>85</b>
6.1	Codecs nach ISO/IEC 11172-3, Layer II . . . . .	85
6.1.1	Aufbau des Codecs . . . . .	86
6.1.2	Bitdatenstrom, globale und rahmenbezogene Parameter . . . . .	88
6.1.3	Leistungsmerkmale . . . . .	89
6.2	Direkteinbettungscodec: Decoder . . . . .	89
6.2.1	Strukturierung und Modifikation des Originalcodecs . . . . .	89
6.2.2	Blockeinteilung und Bitdatenstrom . . . . .	90
6.2.3	Verfügbare Einbettungsbitbreite . . . . .	90
6.2.4	Bitallokieralgorithmus . . . . .	90
6.2.5	Einbettung und Einbettungsformat . . . . .	93
6.2.6	NPR-Problematik und Fehlerkorrektur . . . . .	96
6.3	Direkteinbettungscodec: Encoder . . . . .	97
6.3.1	Strukturierung . . . . .	97
6.3.2	Einbettungsdetektion und -decodierung . . . . .	97
6.3.3	Hybridencoder . . . . .	97
6.4	Fehlerregionencodec . . . . .	99
6.4.1	Erweiterung des MPEG-Bitdatenstroms . . . . .	99
6.4.2	Einbettung, Detektion und Extraktion . . . . .	100
6.5	Codec-Tests . . . . .	101
6.5.1	Testkriterien und Anforderungen . . . . .	101
6.5.2	Testmaterial und -konfiguration . . . . .	102
6.5.3	Ausgewählte Teststücke . . . . .	103
6.5.4	Einbettungskapazitäten . . . . .	103
6.5.5	Testläufe und Resultate . . . . .	108
6.5.6	Hörtests und Befragung von Testpersonen . . . . .	111
6.5.7	Objektive Messungen . . . . .	117

<b>7</b>	<b>MPEG-1 Layer III - Codec</b>	<b>123</b>
7.1	Funktionsweise von Layer III-Codecs . . . . .	123
7.1.1	Encoder . . . . .	123
7.1.1.1	Analysetransformation und Blockgrößen . . . . .	124
7.1.1.2	Skalierungsfaktoren . . . . .	126
7.1.1.3	Bitallokierung . . . . .	126
7.1.1.4	Huffman-Codierung . . . . .	127
7.2	Verfügbare Einbettungskapazität . . . . .	127
7.3	Einzubettende Coderinformationen . . . . .	128
7.4	Mehrstufige Einbettung . . . . .	129
7.5	Problematiken der Direkteinbettung . . . . .	131
<b>8</b>	<b>Anwendungen und Erweiterungen</b>	<b>133</b>
8.1	Erweiterungen der Generationen-Codecs . . . . .	133
8.1.1	Bitreservoitechnik und Fehlerregionencodex . . . . .	133
8.1.2	Reduktion von Arithmetik- und NPR-Fehlern . . . . .	134
8.1.3	Stabilität gegenüber Signalverarbeitungsoperationen . . . . .	134
8.1.4	Vorkomprimierung . . . . .	135
8.2	Adaptation auf andere Audiocodecs . . . . .	135
8.2.1	ATRAC . . . . .	135
8.2.2	PASC . . . . .	136
8.2.3	AC-2/AC-3 . . . . .	136
8.2.4	MPEG-2 AAC . . . . .	137
8.3	Editiersysteme . . . . .	137
8.4	Audio-Vertextung und Partiturintegration . . . . .	138
8.5	Weitere Anwendungsgebiete . . . . .	139
8.5.1	Bildkompression . . . . .	139
8.5.2	Sprachverarbeitung und prädiktive Einbettung . . . . .	140
<b>9</b>	<b>Zusammenfassung und Diskussion</b>	<b>143</b>



# Kapitel 1

## Einleitung

Noch vor wenigen Jahren war die Verbreitung von qualitativ hochwertigem Audiomaterial auf einige wenige Quellen wie den Musikfach- oder Versandhandel beschränkt. Die Qualität sonstiger Mitschnitte, z.B. von Rundfunkaufnahmen oder Analogkopien, war meist von minderer Qualität. Heutzutage sorgen weitverbreitete digitale Medien wie CD (Compact Disk), DAT (Digital Audio Tape), DVD (Digital Versatile Disk) oder MiniDisk für kostengünstige, persistente Speichermöglichkeiten bei ausgezeichneter Wiedergabequalität. Weiterhin sorgt die zunehmende globale Vernetzung für schnelle Verbreitungsmöglichkeiten beliebiger digitaler Daten. So ist es bereits teilweise möglich, neu produziertes Audiomaterial über das Internet probenzuhören, zu bestellen oder sogar sofort zu beschaffen. Harddiskrecording- und Editiersysteme, sowie die mittlerweile sehr kostengünstigen CD-Recorder ermöglichen einem breiten Personenkreis die Erstellung hochqualitativen Audiomaterials.

Die globale Vernetzung schafft mit steigender Verfügbarkeit multimedialer Dokumente eine Vielzahl neuer Fragestellungen und Probleme. Sowohl bezüglich der Geschwindigkeit und Bandbreite von Netzwerken als auch bezüglich lokaler Speicherkapazität und Rechnerleistung stößt man hier immer wieder an Grenzen des momentan physikalisch Machbaren. Da die Kapazitätsanforderungen meist über die Ressourcen vorhandener Netzwerke hinausgehen, werden viele Datensätze hochkomprimiert übertragen. Im Falle von Bilddaten werden unter Ausnutzung psychovisualer Phänomene Kompressionsraten von mehr als 1:100 unter Einhaltung sehr guter Darstellungsqualität erreicht. Im Falle von Audiodaten liefern psychoakustische Phänomene eine Datenreduktion von 1:20 und mehr. Bei entsprechender Netzkapazität sind somit Echtzeit- oder Onlineübertragungen audiovisueller Daten möglich.

Die Verbreitung und Vermittlung digitaler Dokumente schafft weiterhin eine Reihe logistischer und juristischer Probleme, die in Zusammenhang mit der Verwendung in Netzwerken für zahlreiche technische Fragestellungen sorgen. Einige Beispiele sind:

- Netzbasierte *Abrechnungs- und Zahlungssysteme*. Hier existieren bereits einige Ansätze, z.B. Electronic Cash. Problematisch ist allerdings häufig die sichere Übertragung von Informationen über das Netzwerk sowie die Authentifizierung.
- Digitale *Copyright- und Wasserzeichen*. Da sich Standardformate für digitale Dokumente sowie Verbreitungsmechanismen im Internet schon deutlich vor der Entwicklung der kommerziellen Nutzbarkeit etabliert haben, sind in Standardformaten selten oder nur unzureichend Mechanismen für Copyright- oder Wasserzeichen vorgesehen.
- Umgang mit *Netzkriminalität* oder kriminellen Handlungen in Verbindung mit dem

Internet.

Vollständig neuartige Fragestellungen werden ebenfalls aufgeworfen. Eines der größten Probleme in diesem Zusammenhang ist die durch die Vielzahl an verfügbaren Dokumenten und Verweisen (z.B. Hyperlinks) entstehende Informationsflut. Neben der Tatsache, daß oft nicht einmal eine sinnvolle *textuelle* Suche im Internet möglich ist, ergibt sich die technisch wesentlich anspruchsvollere Fragestellung einer *inhaltsbasierten* Suche in allgemeinen multimedialen Dokumenten. Hierbei kommt noch erschwerend hinzu, daß neben einer fehlenden standardisierten textuellen Schnittstelle zur Beschreibung multimedialer Inhalte auch grundlegende Techniken zur automatischen Extraktion und Indexierung nach dem heutigen Stand der Technik nicht verfügbar sind. Solche Techniken wären z.B. wichtig, um Text aus gesprochenen Datensätzen oder Noten aus Musikstücken zu extrahieren. Der geplante MPEG-7 Standard beschäftigt sich mit dem Schnittstellenproblem [24].

Verschiedenste Aspekte in diesem Zusammenhang werden in dem im Jahre 1998 begonnenen DFG-Schwerpunktprogramm *Verteilte Verarbeitung und Vermittlung digitaler Dokumente* ( $V^3D^2$ ) behandelt.

### Audiodatenbanken

Neben Datenübertragungsaspekten wurde in den letzten Jahren verstärkt die Frage nach Archivierung von und Suche in digitalen Audiodaten aufgeworfen. Die komprimierte Speicherung der Daten kann dabei aus vielerlei Hinsicht attraktiv sein. Einerseits kann so eine erhebliche Platzeinsparung erfolgen, andererseits jedoch auch der Zugriff auf die Daten erheblich beschleunigt werden. Werden Daten über das Internet ausgetauscht, ist die komprimierte Form sowieso meist die zu bevorzugende. Ein großer Nachteil der komprimierten Speicherung ist allerdings der damit einhergehende Informations- und mögliche Qualitätsverlust der Audiodaten, da psychoakustische Codierverfahren mit hohen Kompressionsraten informationstheoretisch *verlustbehaftet* sind. Zwar ist nach einmaliger Kompression und Dekompression mit gängigen Verfahren jeweils eine Reproduktion der Audioqualität des Originals möglich, jedoch können weitere Kompressions-Dekompressionsschritte die Qualität des Signals erheblich beeinträchtigen. Eine Verarbeitung komprimiert vorliegender Audiostücke innerhalb einer

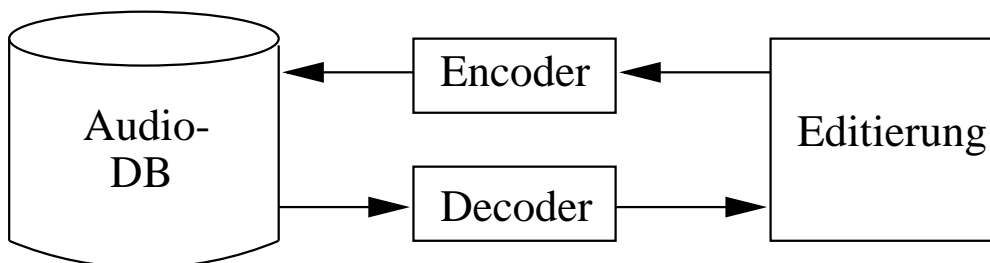


Abbildung 1.1: Audiodatenbank/Editiersystem mit komprimierter Speicherung.

Audiodatenbank ist unter diesen Bedingungen für viele Anwendungen wie z.B. im Rundfunkbereich, bei Plattenfirmen oder Produzenten nicht akzeptabel. Abb. 1.1 zeigt das Szenario einer Audiodatenbank mit angebundenem Editiersystem und komprimierter Speicherung der Audiosignale.

## Beiträge der Arbeit

Den Schwerpunkt der vorliegenden Arbeit bildet die Entwicklung eines Verfahrens zur Vermeidung von Qualitätsverlusten bei wiederholter Kompression und Dekompression (Tandemkompression) von Audiosignalen. Hauptsächlich geht es dabei um die oben genannten psychoakustischen Codierer, die ein Signal unter Einhaltung einer sehr hohen perzeptuellen Qualität mit hohen Kompressionsraten codieren können. Das Augenmerk bezüglich der Tandemkompression wurde bisher stets auf einen möglichst geringen Qualitätsverlust bei der wiederholten Kompression und Dekompression gelegt. So wird innerhalb des G.728-Standards [14] ein Postfilter verwendet, um bestimmte Frequenzen des Signals bei der Dekomprimierung zu verstärken. Auf diese Weise wird bei dreifachem Tandemcoding noch eine akzeptable Qualität erreicht. Der MPEG-2 AAC schneidet ebenfalls bei mehrmaliger Tandemcodierung relativ gut ab [4].

Im Gegensatz dazu erlaubt der hier verfolgte Ansatz, daß ein Audiostück beliebig oft komprimiert und dekomprimiert werden kann, ohne eine schlechtere Qualität als die erste komprimierte Version aufzuweisen. Hierzu wird systematisch der Ansatz verfolgt, alle benötigten Coderinformationen von der ersten zu allen folgenden Codecstufen weiterzugeben. Aufgrund der ähnlichen Strukturen ist der vorgestellte Ansatz auf die meisten der bekannten Audiocoder anwendbar.

Zur Vermeidung neuer Datenformate und eventueller Sekundärdatenströme zusätzlich zum decodierten Audiosignal wird eine Dateneinbettungstechnik entwickelt, die es erlaubt, benötigte Informationen von Codecstufe zu Codecstufe zu übertragen. Die Informationen werden mit Hilfe dieser Technik so im decodierten Audiosignal untergebracht, daß keine Qualitätsverluste wahrnehmbar sind.

Die Integration der formal zugänglichen Einbettungsverfahren in den psychoakustischen Audiocodec ermöglicht es insbesondere, Aussagen über die Funktionalität des Verfahrens zur Vermeidung von Generationseffekten zu *beweisen*. Solche Beweise sind im Zusammenhang mit psychoakustischen Audiocodecs naturgemäß nicht oft möglich, und viele Resultate müssen mit Hilfe aufwendiger Testverfahren verifiziert werden.

Die vorgestellten Dateneinbettungsverfahren sind auch von obiger Anwendung losgelöst auf beliebige Vektorquantisierer anwendbar und liefern eine Reihe interessanter Möglichkeiten im Zusammenhang mit neuen Ansätzen zur Audiokompression. Als Nebeneffekt sind z.B. alle der erstmaligen Kompression nachfolgenden Kompressionsschritte mit einer höheren Geschwindigkeit möglich. Da die vorgestellte Art der Dateneinbettung sehr hohe Datenübertragungsraten erlaubt, ist hiermit weiterhin eine formatunabhängige Alternative zu CD+TEXT oder CD+MIDI bzw. CD+G gegeben [48].

## Netzwerkbasierete Anwendungen

Die Tandemcodierung ist insbesondere im Bereich der Kommunikationsnetzwerke interessant. Hier kommt es häufig zu kaskadierten Codierungen und Decodierungen, z.B. bei wechselnden Netzbetreibern oder reduzierter Netzkapazität. In einer Erweiterung auf heterogene Codec-Kaskaden kann das vorgestellte Verfahren hier einer möglichen Verschlechterung der Signalqualität vorbeugen oder diese verhindern.

Eine wichtige Anwendung steht im Zusammenhang mit dem internetbasierten Vertrieb von Audio- bzw. CD-Material. Da die bestehenden Übertragungskapazitäten bei weitem nicht ausreichen, werden Audiodateien heutzutage in komprimierter Form zum Kauf angeboten. Eine

dekomprimierte Version des gekauften Audiomaterials auf CD ersetzt somit die herkömmliche Audio-CD. Der Nachteil hierbei ist jedoch, daß eine wiederholte Komprimierung bereits deutliche Qualitätsverluste mit sich bringen kann. Es besteht die Notwendigkeit, über bestehende *Generationen* Buch zu führen. Das in dieser Arbeit vorgestellte Verfahren kann in diesem Punkt Abhilfe schaffen.

### Weitere Anwendungen

Im Hinblick auf das obige Beispiel der Audiodatenbank ist es attraktiv, die in dieser Arbeit entwickelten Grundideen zur Realisierung eines digitalen Editiersystems zu verwenden. Die dazu noch zu leistende Arbeit liegt dabei insbesondere in der Berücksichtigung möglicher Editieroperationen und deren Auswirkung auf die Coderparameter. Besonders im Bereich des Home-Recordings und im Falle digitaler Schnittplatzsystemen mit nur begrenzter Speicherkapazität oder langsamer Netzanbindung könnte so die Bereitstellung umfangreichen Audiomaterials ermöglicht werden.

### Gliederung

Die Arbeit gliedert sich wie folgt. Im zweiten Kapitel wird eine kurze Einführung in mathematische, signaltheoretische und psychoakustische Aspekte der Audiocodierung gegeben. Anhand der Standardisierungsaktivitäten der letzten Jahre werden Entwicklungen und Anforderungen im Zusammenhang mit dem Einsatz von Audiocodern in multimedialen Systemen erläutert. Abschließend wird auf aktuelle Forschungsaktivitäten eingegangen. Im dritten Kapitel werden digitale Generationseffekte behandelt. Anhand von Hörtests im Falle einer konkreten Codecrealisierung wird untersucht, wie und ab welcher digitalen Generation die Qualität eines Audiosignals beeinträchtigt wird. Als Grundtechnik zur Vermeidung von Generationseffekten wird im vierten Kapitel die Dateneinbettung behandelt. Dabei wird für allgemeine Vektorquantisierer das Konzept der Fehler- und Einbettungsregionen eingeführt und bezüglich verschiedener Aspekte, wie z.B. Fehlerkorrekturmechanismen, untersucht. Darauf aufbauend wird in Kapitel fünf zunächst theoretisch untersucht, welche Klassen von decodierten Signalen zu einem gegebenen codierten Signal bezüglich einer Vermeidung von Generationseffekten „sinnvoll“ sind. Mit Hilfe der Einbettungsverfahren werden zwei Grundalgorithmen für Audiocodierer zur Vermeidung von Generationseffekten entworfen. Kapitel sechs beschreibt die konkrete Realisierung solcher Audiocodecs am Beispiel des MPEG-1 Standards. Anhand von Hörtests wird die Funktionsfähigkeit des realisierten Codecs überprüft. Kapitel sieben beschreibt eine Erweiterung auf den Layer III des MPEG-1 Codecs. Abschließend wird in Kapitel acht auf mögliche Anwendungen von Dateneinbettungsverfahren und Erweiterungen sowie Verbesserungen der vorgestellten Codecs eingegangen. Anwendungen in der Sprach- und Bildverarbeitung werden kurz untersucht.

## Kapitel 2

# Grundlagen der Audiokompression

### 2.1 Mathematische und signaltheoretische Grundlagen

Die Fourieranalysis bildet die mathematische Grundlage der in dieser Arbeit verwendeten Bereiche der Signalverarbeitung. Durch sie wird die Darstellbarkeit von Funktionen des Signalraums  $L^2(\mathbb{R})$  der quadratintegrierbaren Funktionen [10] als eine Überlagerung von Schwingungen,

$$\hat{f}(\omega) := \mathcal{F}[f](\omega) := \lim_{N \rightarrow \infty} \int_{|t| < N} f(t) e^{-2\pi i \omega t} dt,$$

und

$$f(t) = \int_{-\infty}^{\infty} \hat{f}(\omega) e^{2\pi i \omega t} d\omega, \quad (2.1)$$

garantiert. Hierbei ist  $\hat{f}$  die (kontinuierliche) *Fouriertransformierte von  $f$*  und die Darstellung (2.1) von  $f \in L^2(\mathbb{R})$  ein nichttriviales Ergebnis der Fourieranalysis. Die in Anwendungen auftretenden Signale werden im allgemeinen als  $L^2(\mathbb{R})$ -Funktionen betrachtet. Ist  $\hat{f}(\omega) = 0$  für  $|\omega| > \Omega$ , so heißt  $\hat{f}$   $\Omega$ -bandbegrenzt. Ist lediglich die Existenz eines solchen  $\Omega$  gesichert, heißt  $\hat{f}$  einfach nur *bandbegrenzt*. Bilden  $[\Omega_1, \Omega_2]$  und  $[-\Omega_2, -\Omega_1]$  den Träger von  $\hat{f}$ , so heißt  $\Omega_2 - \Omega_1$  die *Bandbreite* von  $f$ . Eine ausführliche Behandlung der kontinuierlichen Fourieranalysis findet sich in [10].

Wir werden es im folgenden ausschließlich mit diskreten Signalen zu tun haben. Diskrete Signale werden mathematisch durch Abbildungen  $\mathbb{Z} \rightarrow \mathbb{R}$  oder  $\mathbb{Z} \rightarrow \mathbb{C}$  modelliert, manchmal werden *endliche* Signale  $I \rightarrow \mathbb{R}$ ,  $I \subset \mathbb{Z}$  endlich, benötigt. Diskrete Signale entstehen aus kontinuierlichen durch *Abtastung*. Das Abtasttheorem nach Shannon besagt dabei, daß bandbegrenzte Signale schon durch ihre Werte an einer diskreten Menge von Punkten darstellbar sind [1].

Gebäuchliche Operatoren, die im Zusammenhang mit Filterbänken wichtig sind, arbeiten auf geeigneten Teilmengen der diskreten Signale. Wir verwenden die Signalkräume  $\ell^1(\mathbb{Z})$  bzw.  $\ell^2(\mathbb{Z})$  der absolut summierbaren bzw. quadratsummierbaren Signale, wo

$$\ell^p(\mathbb{Z}) := \left\{ f : \mathbb{Z} \rightarrow \mathbb{C} \mid \sum_{i \in \mathbb{Z}} |f(i)|^p < \infty \right\}$$

für  $p \in [1, \infty)$ .

Auf  $\ell^2(\mathbb{Z})$  werden einige lineare Operatoren definiert. Hierfür sei  $g \in \ell^2(\mathbb{Z})$ :

- *Translation*  $T_\ell : g \mapsto g(\cdot + \ell)$ .
- Die Translation ist ein Spezialfall der *Faltung*. Sei zusätzlich  $h \in \ell^1(\mathbb{Z})$ , dann definiert

$$c_h : g \mapsto \sum_{i \in \mathbb{Z}} h(i)g(\cdot - i)$$

eine lineare Abbildung auf  $\ell^2(\mathbb{Z})$ . Es gilt  $T_\ell = c_{\delta_{-\ell}}$ , wobei  $\delta_\ell$  der um  $\ell$  verschobene Einheitsimpuls mit  $\delta_\ell(k) = 1$ , falls  $\ell = k$ , und  $\delta_\ell(k) = 0$  sonst, ist. In der Signalverarbeitung entspricht die Faltung der (linearen) *Filterung*. In diesem Zusammenhang wird  $h$  als *Filter* bezeichnet.

- *Downsampling* ( $\downarrow M$ ) für  $M \in \mathbb{N}$ , wo  $g \mapsto g(\cdot M)$ .
- *Upsampling* ( $\uparrow M$ ) für  $M \in \mathbb{N}$ , wo  $g \mapsto (\uparrow M)[g]$  und

$$(\uparrow M)[g](n) = \begin{cases} g(n/M) & \text{falls } M|n, \\ 0 & \text{sonst.} \end{cases}$$

Die Operatoren des Up- und Downsamplings erlauben eine „Erhöhung“ bzw. „Reduzierung“ der Abtastrate. Man spricht in diesem Zusammenhang von *Multiratensignalverarbeitung*. Mathematisch wurde der Abtastratenänderung keine Rechnung getragen. Dies kann durch eine Änderung des Definitions- oder Bildbereichs geschehen. Der Downsamplingoperator würde somit zu einer Abbildung  $\mathbb{Z} \rightarrow M\mathbb{Z}$  und nach dem Downsampling wird nur noch  $\mathbb{Z} \cong M\mathbb{Z}$  betrachtet. Einen Überblick über die Multiratensignalverarbeitung findet sich in [59], eine detaillierte Einführung in [60].

Zur Separierung verschiedener Frequenzbereiche eines Signals wird häufig eine *Filterbank* verwendet, wobei das Signal mit mehreren verschiedenen Filtern  $h_0, \dots, h_{M-1} \in \ell^1(\mathbb{Z})$  gefaltet wird. In unseren Anwendungen wird für  $0 \leq i < M$  stets  $h_i \in \mathbb{R}^n \hookrightarrow \ell^1(\mathbb{Z})$  gelten. Für ein  $x \in \ell^2(\mathbb{Z})$  liefert eine dezimierte Filterung

$$s_i := (\downarrow L_i) \circ c_{h_i} \circ T_i[x], \text{ für } 0 \leq i < M,$$

*Subbandsignale*  $s_i$ . Gilt  $\forall i : L_i = M$ , spricht man von *kritischer Abtastung* oder *kritischer Dezimierung*. Da auf diese Weise insgesamt die ursprüngliche Datenrate erhalten bleibt und auch theoretisch das Originalsignal durch die Wahl geeigneter Filter wieder rekonstruiert werden kann, wird im folgenden stets von kritischer Dezimierung ausgegangen.

Zur Rekonstruktion eines Signals aus dezimierten Subbandsignalen definiert man eine Rücktransformationsoperation

$$\tilde{x} := \sum_{i=0}^{M-1} T_{-i} \circ c_{f_i} \circ (\uparrow M)[s_i]$$

für Filter  $f_i \in \ell^1(\mathbb{Z})$ .

**Definition 2.1.1** *Ein Operator*  $S : \ell^2(\mathbb{Z}) \rightarrow \ell^2(\mathbb{Z})$  *liefert perfekte Rekonstruktion falls*  $S = T_\ell$  *für ein geeignetes*  $\ell \in \mathbb{Z}$  *gilt.*

Eine Filterbank obiger Struktur liefert somit perfekte Rekonstruktion, falls

$$\sum_{i=0}^{M-1} T_{-i} \circ c_{f_i} \circ (\uparrow M) \circ (\downarrow M) \circ c_{h_i} \circ T_i = T_\ell$$

für ein  $\ell \in \mathbb{Z}$  gilt. Filterbänke werden häufig *kaskadiert*, d.h. hintereinandergeschaltet. Dies erlaubt eine genauere Aufteilung der einzelnen Frequenzbereiche in Subbänder der Breiten  $f/M$ , wobei  $f$  die ursprüngliche Bandbreite des diskreten Signals und  $M$  ein geeigneter Gesamtdezimierungsfaktor ist [35]. Aufteilungen in Subbänder mit Bandbreiten  $fN/M$ ,  $N > 1$ , werden in [32] behandelt.

Lineare Faltungen können als Matrix-Vektor Multiplikation geschrieben werden. Ein Faltungsoperator entspricht einer Matrix  $(a_{i,j})$  mit konstanten Diagonalelementen, also  $\forall i, j \in \mathbb{Z} : a_{i,j} = a_{i+1,j+1}$ . Solche Matrizen heißen *Toeplitzmatrizen*. Sind anstelle der  $a_{ij}$  Einträge in Form von quadratischen Matrizen  $A_{i,j} \in M_n(\mathbb{C})$  gegeben, sprechen wir von einer *Block-Toeplitzmatrix*. Die Zahl  $n$  soll als *Toeplitz-Blocklänge* bezeichnet werden. Hier gilt ebenfalls  $\forall i, j : A_{i,j} = A_{i+1,j+1}$ . Für endliche Matrizen wird der Begriff der Toeplitzmatrix geeignet modifiziert. Im Falle von Filtern endlicher Länge werden wir es immer mit Filterlängen  $\ell = kn$ , also Vielfachen einer Toeplitz-Blocklänge, zu tun haben.

Eine *Blocktransformation* ist durch eine Block-Toeplitzmatrix gegeben, in der nur die Diagonalblöcke  $A_{ii}$  von Null verschieden sind. Für die Signalverarbeitung wichtige Beispiele sind orthogonale oder unitäre Matrizen, wie die diskreten Cosinustransformationen [49] oder diskreten Fouriertransformationen der zyklischen Gruppe mit  $n$  Elementen,

$$\text{DFT}_n(\omega) := (\omega^{k\ell})_{0 \leq k, \ell < n}, \quad \omega \text{ primitive } n\text{-te Einheitswurzel, z.B. } \omega \in \left\{ e^{\frac{2\pi i}{n}}, e^{-\frac{2\pi i}{n}} \right\}.$$

Manchmal werden auch „überlappende“ Formen noch als Blocktransformationen bezeichnet. Von einer doppelten oder 50%-Überlappung spricht man, wenn noch einer der Nebendiagonalblöcke  $A_{i,i+1}$  oder  $A_{i,i-1}$  von Null verschieden ist. Eine  $k$ -fache Überlappung erhält man, falls  $A_{i,i}, \dots, A_{i,i+k} \neq 0$  und  $A_{i,i+\ell} = 0$  für alle  $\ell < -k$  und  $\ell > k$ . Man spricht hier auch von *Band-Toeplitzmatrizen*. Die Darstellung der Überlappung ist hier o.B.d.A. so normalisiert, daß die Hauptdiagonale die erste nicht-Null Matrix enthält. Die *Blockgröße* ist durch die Zahl  $\ell = kn$  gegeben.

Charakteristisch für eine Blocktransformation ist die Interpretation der Transformationskoeffizienten als Spektral- oder Frequenzinformation. Die Blocklänge  $n$  ist dabei meist relativ groß, z.B.  $n > 100$ ,  $n = 512$  oder  $n = 1024$ . Bei einer Abtastrate des diskreten Signals von 44100 Werten pro Sekunde entspricht eine Blockgröße von 1024 einer zeitlichen Auflösung von ca. 44 Millisekunden (mS). Der Begriff *Auflösung* wird dabei durch die Tatsache, daß alle anderen Blöcke der Block-Toeplitz Matrix verschwinden, gerechtfertigt. Die fortlaufenden Blöcke der Matrix beschreiben somit in Verbindung mit der Matrix-Vektor Multiplikation eine blockweise oder fensterweise Verarbeitung oder Analyse des Signals.

Mit steigender Blockgröße  $\ell$  (und kleiner werdenden  $n \times n$  Blöcken der Block-Toeplitzmatrix), ändert sich die Blocktransformationssicht in eine Subbandtransformationssicht. Im Falle der Subbandtransformationen ist die Anzahl  $M$  der Filter gleich der Toeplitz-Blocklänge  $n$ . Im allgemeinen ist dann  $M$  klein, etwa 4, 8, 16 oder 32. Eine kleine Toeplitz-Blocklänge  $M$  liefert hier eine verbesserte zeitliche Auflösung.

Die Signalverarbeitungsschritte finden bei Block- und Subbandtransformation im allgemeinen unter Verwendung von Vielfachen der Blocklängen bzw. Filterlängen,  $\ell\lambda$ ,  $\lambda \in \mathbb{N}$ , statt. Für Anwendungen in psychoakustischen Codern wird diese Verarbeitungslänge entsprechend dem menschlichen Wahrnehmungsverhalten in der Größenordnung von einer bis ca. 80 mS, also einer zeitlichen Auflösung von 44 bis 2000 Werten bei einer Abtastrate von 44100 Hz, gewählt. Abschließend sollen noch einige Schreibweisen erklärt werden:

- Für Matrizen  $A = (a_{ij}) \in \mathbb{C}^{r \times r}$ ,  $B = (b_{ij}) \in \mathbb{C}^{s \times s}$ , sei das *Kroneckerprodukt* durch

$A \otimes B := (a_{ij}B)_{ij} \in \mathbb{C}^{rs \times rs}$  und die direkte Summe  $A \oplus B$  durch

$$A \oplus B := \begin{pmatrix} A & 0_{r \times s} \\ 0_{s \times r} & B \end{pmatrix}$$

definiert, wobei  $0_{a \times b}$  die  $a \times b$  Nullmatrix ist.

- Intervalle  $a, a + 1, \dots, a + k =: b$  in  $\mathbb{Z}$  werden abkürzend als  $[a : b]$  geschrieben.
- Die Symbole  $\cup$  bzw.  $\sqcup$  bezeichnen die Vereinigung bzw. die disjunkte Vereinigung von Mengen.
- Pegelangaben für Signale oder Rauschsignale werden unter der Annahme, daß die betreffenden Signalwerte als reellwertige Zufallsvariablen  $S$  bzw.  $N$  gegeben sind, abhängig von der Varianz der Zufallsvariablen  $\sigma_S$  bzw.  $\sigma_N$  in logarithmischer Form angegeben. Als Darstellungsform hat sich die Verwendung einer Pseudoeinheit, Dezibel (dB), eingebürgert. In dieser Schreibweise sind  $10 \log_{10}(\sigma_S)$  bzw.  $10 \log_{10}(\sigma_N)$  der Signal- bzw. Rauschpegel in dB. Der Abstand beider Pegel,  $10 \log_{10}(\sigma_S) - 10 \log_{10}(\sigma_N) = d(S, N)$  wird vereinfachend als SNR, *Signal-to-Noise Ratio*, geschrieben. Diese Schreibweise wird häufig auch auf andere Abstände erweitert, z.B. auf die im folgenden eingeführten Maskierungsschwellen.

## 2.2 Psychoakustik

Die von modernen Audiocodierern erzielten hohen Kompressionsraten sind auf die systematische Ausnutzung psychoakustischer Phänomene zurückzuführen. Im folgenden soll auf diese Phänomene selbst, deren Anwendungsbereiche in Audiocodern und die Grenzen des momentan Möglichen kurz eingegangen werden. Eine sehr ausführliche Abhandlung über Psychoakustik findet sich in [65]. Eine Diskussion über Anwendungen und Anwendbarkeit der Psychoakustik in der Audiosignalverarbeitung findet sich z.B. in [61].

Das wichtigste in Audiocodern ausgenutzte psychoakustische Phänomen ist der *Maskierungseffekt*. Hiermit wird die Verdeckung eines bestimmten *Zieltons* durch einen *Maskierungston* oder einfach *Maskierer* bezeichnet. Alltägliche Beispiele für den Maskierungseffekt stellen Unterhaltungen zweier oder mehrerer Personen, gestört von lauten Nebengeräuschen, wie z.B. Verkehrslärm, dar. In diesen Fällen muß zur Fortsetzung der Unterhaltung die Lautstärke kurzzeitig angehoben werden.

Ein weiteres Phänomen ist die *absolute Hörschwelle* oder *Ruhehörschwelle* der menschlichen Wahrnehmung, die ausdrückt, wie laut ein Ton in Abhängigkeit seiner Frequenz sein muß, damit er gerade noch wahrgenommen wird. Wie alle anderen Schwellen wird auch dieser Wert als Mittelwert über einer Reihe von Versuchsergebnissen mit verschiedenen Versuchspersonen bestimmt. Daher kann es zu individuellen Abweichungen kommen. Die absolute Hörschwelle wird in Dezibel relativ zu einem Referenzschalldruckpegel von  $2 \cdot 10^{-5}$  Pascal angegeben. Neben der individuellen Variabilität ist die absolute Hörschwelle noch von einigen anderen Parametern abhängig:

- Bei Menschen steigt die Ruhehörschwelle insbesondere in hohen Frequenzbereichen mit wachsendem Alter deutlich an.
- Bei Menschen mit Gehörschäden kann die Ruhehörschwelle an den verschiedensten Stellen stark vom durchschnittlichen Wert abweichen.

- Ein Training auf bestimmte Situationen und Testtöne hin kann bei Menschen dazu führen, daß bestimmte Töne besonders gut oder, andersherum, nicht mehr wahrgenommen werden. Dies ist besonders im Zusammenhang mit subjektiven Hörtests problematisch.
- Ein zu starker Schalldruckpegel, wie z.B. beim Besuch eines lauten Musikkonzerts, kann dafür sorgen, daß die absolute Hörschwelle der Lautstärke angeglichen wird. Dieser Schutzmechanismus, der mit einer Druckangleichung im Innenohr einhergeht, sorgt dafür, daß auch noch einige Stunden nach Ende der lauten Hörwahrnehmung ein taubes Gefühl spürbar ist.

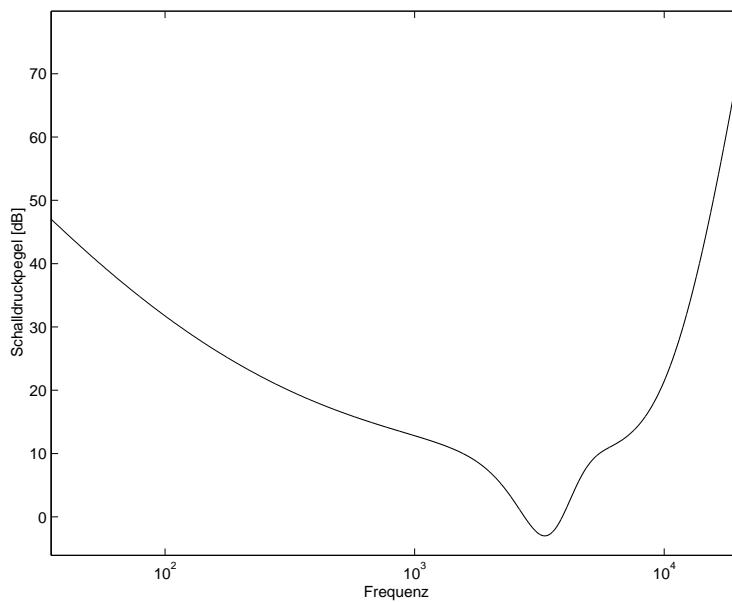


Abbildung 2.1: Menschliche Ruhehörschwelle. Schalldrücke unterhalb dieser Schwelle werden i.a. vom Menschen nicht mehr wahrgenommen.

Aus dieser Betrachtung ist ersichtlich, daß eine Ausnutzung der psychoakustischen Phänomene eine vorsichtige Herangehensweise und die Verwendung gewisser Extremalwerte erfordert. Abb. 2.1 zeigt den ungefähren Verlauf der menschlichen Ruhehörschwelle [64].

### 2.2.1 Maskierung und kritische Bänder

Die Maskierungseffekte sollen ausführlicher beschrieben werden. Da es aufgrund der Heisenbergschen Unschärferelation [29] nicht sinnvoll ist, von *einem* Frequenzereignis an *einem* bestimmten Zeitpunkt (eines Signals) zu sprechen, soll diese Terminologie hier in einer etwas unschärferen Interpretation verwendet werden und bedeuten, daß die Kurzzeitfouriertransformierte

$$\tilde{f}(\omega, t) := \int_{-\infty}^{\infty} \bar{g}(u - t) f(u) e^{-2\pi i u \omega} du \quad (2.2)$$

des Signals  $f \in L^2(\mathbb{R})$  zu diesem Zeitpunkt (entspricht der Variablen  $t$ ) in den jeweiligen Spektralbändern (entspricht der Variablen  $\omega$ ) eine hohe Energie besitzt. In (2.2) ist

$g \in L^2(\mathbb{R})$  eine geeignete Fensterfunktion, die in Anwendungen einen kompakten Träger besitzt. Im Speziellen kann z.B. ein Hann- oder Hammingfenster gewählt werden [29]. Eine

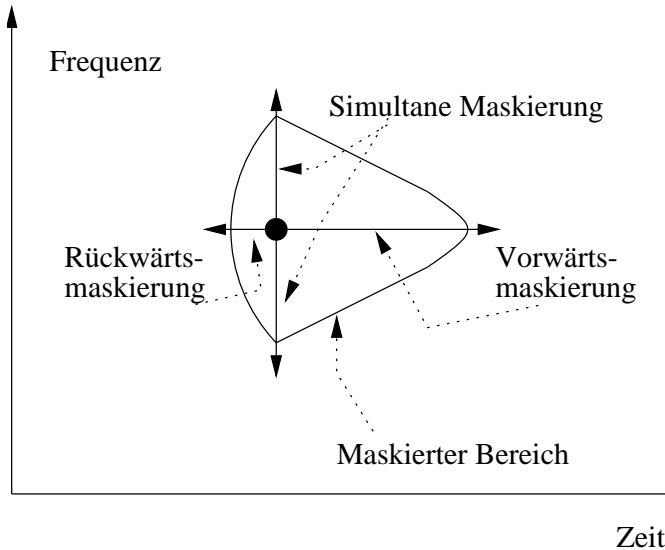


Abbildung 2.2: Wirkungsbereich der Maskierung in der Zeit-Frequenz-Ebene. Temporale Maskierung wird als Vorwärts- bzw. Rückwärtsmaskierung bezeichnet. Eine um einen bestimmten Zeitpunkt auftretende Maskierung nennt man simultan.

Analyse der Maskierung wird meist anhand isolierter Maskierer, also einzelner Tonereignisse in der Zeit-Frequenz-Ebene durchgeführt. Abb. 2.2 zeigt qualitativ den Wirkungsbereich eines Maskierers, der in Zeit und Frequenz an der Stelle des fett wiedergegebenen Punktes lokalisiert ist. Der Effekt der Maskierung, bestimmte Töne durch andere zu überdecken und unhörbar zu machen, kann biologisch grob folgendermaßen erklärt werden: In Anwesenheit eines Maskierers werden bestimmte Stellen der Basilarmembran im Innenohr in Schwingung versetzt. Diese Schwingungen werden durch induzierte Schwebbewegungen der Haarfortsätze der Sinneszellen im Cortischen Organ schließlich in nervöse Reize umgewandelt. Ein in einer gewissen Zeit-Frequenz-Umgebung vorliegender Zielton wird dann überdeckt, wenn er zu wenig Energie besitzt, um die Sinneszellen noch stärker zu erregen [65].

Die Maskierungsbereiche können zeitlich in drei Klassen eingeteilt werden:

1. *Rückwärtsmaskierung* oder *Premasking* beschreibt den Effekt, daß Stimuli, die zeitlich *vor* dem Maskierer präsentiert werden, in gewissem Umfang von diesem verdeckt werden können. Dieser Effekt ist jedoch nur auf kurze Zeitspannen, ca. 10-20 mS, beschränkt.
2. *Simultane Maskierung* beschreibt die Verdeckung anderer Frequenzkomponenten durch den Maskierungston während der Maskierungston existiert. Abb. 2.3 zeigt zur Verdeutlichung dieses Effekts einen vertikalen Schnitt durch Abb. 2.2. Charakteristisch ist der Verlauf der *Maskierungsschwelle*, die angibt, wie laut ein Ton einer anderen Frequenz sein darf, um gerade noch maskiert zu werden.
3. *Vorwärtsmaskierung* oder *Postmasking* beschreibt den Effekt, daß Stimuli, die zeitlich *nach* dem Maskierer präsentiert werden, in gewissem Umfang von diesem verdeckt werden können. Dieser Effekt ist wesentlich stärker ausgeprägt als die Rückwärtsmaskierung und kann sich auf einen Zeitbereich von ca. 200 mS erstrecken.

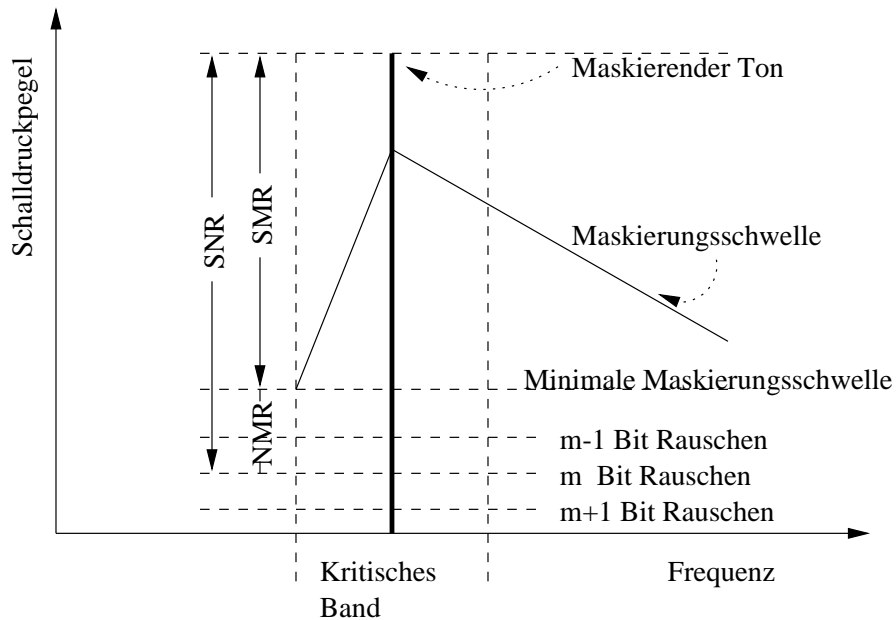


Abbildung 2.3: Wirkungsbereich der simultanen Maskierung. Signal-, Maskierungs- und Rauschschwelle. Eine Erhöhung der Bitrate senkt den Rauschpegel.

Diese drei Bereiche sind natürlich nicht scharf abgegrenzt, sondern eher als Hauptrichtungen des in Abb. 2.2 wiedergegebenen Maskierungsbereichs zu sehen. In der Abbildung fehlt natürlich der Verlauf der Maskierungsintensität. Diese ist, ebenfalls qualitativ, in Abb. 2.3 dargestellt. Die in der Abbildung dargestellte Maskierungsschwelle gibt den Effekt wieder, daß bei der simultanen Maskierung Komponenten mit höheren Frequenzen als der Maskierer stärker überdeckt werden, als niedrigere. Dieser Effekt ist jedoch auch stark abhängig von der absoluten Frequenz.

Eine wichtige Rolle spielt weiterhin die Art des Maskierungstons. Im allgemeinen wird hier zwischen *tonalen* und *nichttonalen* Maskierern unterschieden, da Experimente zwischen diesen Klassen größere Unterschiede im Maskierungsvermögen aufzeigen. Die Begriffe tonal und nichttonal zielen dabei auf das ungefähre Aussehen des Spektrums bzw. der Fouriertransformierten des jeweiligen Tons ab. Nichttonale oder rauschartige Komponenten weisen bekanntlich ein flaches Spektrum auf, wohingegen starke harmonische oder tonale Anteile ausgeprägte Maxima innerhalb des Spektrums induzieren. Die tonal/nichttonal Klassifikation kann entweder über das sogenannte *Spectral Flatness Measure* [28] oder einen Algorithmus zur Suche lokaler Maxima im Spektrum des Signalblocks erfolgen (siehe [18]). Da nichttonale Maskierer simultan an einem bestimmten Punkt der Zeit-Frequenz-Ebene auftretende Töne besser überdecken als tonale Maskierer, wird in Audiocodierern im allgemeinen die stark vergrößernde Annahme gemacht, daß nichttonale Maskierer insgesamt um einen bestimmten (festen) Pegel besser maskieren als tonale. Im ersten der im MPEG-Standard vorgeschlagenen psychoakustischen Modelle wird z.B. die Maskierungsschwelle eines tonalen Maskierers analog zu der eines nichttonalen Maskierers, jedoch unter einer linearen Verschiebung vermindert, berechnet [18]. An dieser Stelle ist jedoch insgesamt noch Platz für Verbesserungen gegeben, da die tatsächliche Maskierung zwischen tonalen und nichttonalen Komponenten zu einem großen Teil noch zu wenig erforscht ist [61]:

Bark	von [Hz]	bis [Hz]	Breite [Hz]	Bark	von [Hz]	bis [Hz]	Breite [Hz]
0	0	100	100	12	1720	2000	280
1	100	200	100	13	2000	2320	320
2	200	300	100	14	2320	2700	380
3	300	400	100	15	2700	3150	450
4	400	510	110	16	3150	3700	550
5	510	630	120	17	3700	4400	700
6	630	770	140	18	4400	5300	900
7	770	920	150	19	5300	6400	1100
8	920	1080	160	20	6400	7700	1300
9	1080	1270	190	21	7700	9500	1800
10	1270	1480	210	22	9500	12000	2500
11	1480	1720	240	23	12000	15500	3500

Abbildung 2.4: Frequenzbänder der Barkskala

- Eigentlich wären alle *vier* wechselseitigen Maskierungseffekte zwischen tonalen und nichttonalen Maskierern zu beachten. Hier sind sogar die zwei besonders interessanten Fälle, in denen das Zielsignal nichttonal ist, noch zu wenig erforscht.
- Experimente zeigen, daß im Hinblick auf eine Maskierungsschwelle wie in Abb. 2.3 ein tonaler Maskierer nicht immer schlechter maskiert, als ein nichttonaler. So zeigt sich, daß nichttonale Maskierer zwar zu niedrigeren Frequenzen hin besser maskieren als tonale, daß sich dieser Effekt jedoch andererseits zu Frequenzen oberhalb des Maskierers hin umwandelt und dort die tonalen Maskierer besser maskieren.

Für die Audiocodierung weiterhin wichtig ist der psychoakustische Gruppierungseffekt, der besagt, daß die menschliche Wahrnehmung Töne benachbarter Frequenzen in bestimmtem Umfang zusammenfaßt oder zusammen analysiert. Da dies so interpretiert werden kann, daß Maskierer sich primär nur innerhalb bestimmter Frequenzgruppen, der sogenannten *kritischen Bänder*, auswirken, wird in vielen Audiocodierern eine Signalanalyse gezielt bezüglich dieser Frequenzgruppen durchgeführt. Abb. 2.3 zeigt beispielhaft die Grenzen eines kritischen Bandes. Die Breiten der kritischen Bänder sind frequenzabhängig und können experimentell ermittelt werden. Bildet man durch Aneinanderreihung von Bändern kritischer Bandbreite (in Abhängigkeit von der jeweiligen Frequenz) eine Einteilung der Frequenzachse, erhält man die *Barkskala*. Die Zählung beginnt bei 0 Hz und Bark 0. Experimente zeigen, daß die kritischen Bänder bis ca. 500 Hz eine Breite von ca. 100 Hz besitzen und sich danach auf ca. 20% der Mittenfrequenz des jeweiligen Bandes verbreitern. In Tabelle 2.4 sind experimentell ermittelte Werte der Barkskala nach Zwicker [65] angegeben.

Eine konservative Wahl der Maskierungsschwelle für ein kritisches Band stellt die in Abb. 2.3 eingezeichnete minimale Maskierungsschwelle dar. Neben dem im letzten Abschnitt eingeführten Signal-Rauschabstand, sind in der Abbildung weiterhin der Signal-Maskierungsschwellenabstand (SMR, *Signal-to-Mask Ratio*) und der Maskierungsschwellen-Rauschabstand (NMR, *Noise-to-Mask Ratio*) eingezeichnet. Der Rauschpegel wird dabei meist in Abhängigkeit von der gewählten Quantisierungsaufösung (in Bits) angegeben. Für einen linearen Quantisierer kann hier bekanntlicherweise eine gegebene Bitrate unmittelbar in einen Rauschpegel umgerechnet werden. Der Rauschpegel steigt dabei um grob 6 dB pro Bit [26].

### 2.2.2 Globale Maskierungsschwelle

Da man es in Musik- oder Sprachsignalen in allgemeinen nie mit isolierten Maskierern zu tun hat, besteht die Notwendigkeit, die Maskierungsschwellen aller relevanten Maskierer zu kombinieren. Psychoakustische Modelle liefern dann insgesamt eine globale Maskierungsschwelle für einen bestimmten Signalblock.

Die Unterteilung der Frequenzachse in feste kritische Bänder ist natürlich in gewisser Weise willkürlich, da die psychoakustischen Phänomene sich hauptsächlich auf die Frequenzbreite und nicht auf die absolute Position des Bandes beziehen. Daher wird sowohl die Maskierungsschwelle eines Maskierers für sein jeweiliges kritisches Band als auch dessen maskierende Wirkung auf die benachbarten Bänder berechnet. Hierzu wird die sogenannte *Spreading-Funktion* verwendet [54]. In erster Näherung wird hier angenommen, daß die maskierende Wirkung nur vom (relativen) Abstand der Bänder abhängt. Die Gesamtmaskierungsschwelle eines Bandes wird durch Addition der Maskierungswirkungen aller Bänder hierauf berechnet. Diese Auswertung ist damit als Matrix-Vektor Multiplikation ausführbar, wobei die Spreading-Werte eine endliche Toeplitz-Matrix bilden [28].

Zusammenfassend kann die Berechnung eines psychoakustischen Modells aus folgenden Schritten bestehen:

1. Spektralanalyse des Signalblocks.
2. Gegebenenfalls Einbeziehung von Informationen vorheriger Blöcke, etwa der perzeptuellen Entropie [27], zur Ausnutzung zeitlicher Effekte.
3. Identifizierung von möglichen Maskierern.
4. Klassifizierung der Kandidaten in tonale und nichttonale Maskierer.
5. Aufteilung in relevante und nicht relevante Maskierer.
6. Berechnung der individuellen Spreading- und Maskierungsfunktionen.
7. Kombination der individuellen Maskierungsfunktionen zur globalen Maskierungsschwelle.
8. Einbeziehung der Ruhehörschwelle in die Maskierungsschwelle. Dies ist besonders bei hohen Frequenzen erforderlich. Die Einbeziehung kann über eine Maximumbildung erfolgen.
9. Berechnung des SMR aus der Maskierungsschwelle und der Signalenergie für eine geeignete Subbandaufteilung.

## 2.3 Audiokompression

In diesem Abschnitt sollen momentane Ergebnisse und Entwicklungen in der Audiocodierung aufgezeigt werden. Hierbei geht es in erster Linie um allgemeine Ziele und verwendete Techniken. Eine detailliertere Beschreibung psychoakustischer Kompressionsverfahren wird in den Kapiteln 5 bis 7 anhand der dort vorgestellten Methoden gegeben.

Zunächst wird kurz auf Standardisierungsaktivitäten im Bereich der Audiocodierung innerhalb der letzten zehn Jahre eingegangen. Da sich inzwischen sowohl Forscher vieler renommierter universitärer als auch industrieller Einrichtungen daran beteiligen, soll dies anhand

der Audiokomponenten der MPEG-Familie (Moving Pictures Experts Group), d.h. den entsprechenden Teilen der MPEG-1, MPEG-2, MPEG-4 und MPEG-7 Standards, geschehen. Im Audibereich haben sich die MPEG Coder bereits in vielen Anwendungsgebieten durchgesetzt [44, 48]. Anschließend werden parallel in der Forschung entwickelte Methoden und entstandene Trends skizziert. Ein Überblick über die Entwicklung der Audiocodierung innerhalb des letzten Jahrzehnts findet sich in [46].

### 2.3.1 Kompression und Multimedia: MPEG Standards

Innerhalb der *International Organization for Standardization* (ISO) und der *International Electrotechnical Commission* (IEC) beschäftigt sich das technische Komitee JTC 1, „Information technology“, und genauer das Unterkomitee SC 29, „Coded representation of audio, picture, multimedia and hypermedia information“, mit der Entwicklung der allgemein hin unter dem Kurznamen *MPEG* bekannten Standards. Bisher wurden hier MPEG-1, MPEG-2 und Teile von MPEG-4 [16, 19, 22] standardisiert. Im Falle von MPEG-4 ist seit Februar 1999 die sogenannte *Version 1* standardisiert. Die abwärtskompatible *Version 2* von MPEG-4 soll ein Jahr später bereitgestellt werden [23]. Insgesamt bildet MPEG mit momentan ca. 300 Mitgliedern aus dem industriellen und universitären Bereich eine sehr große Arbeitsgruppe, die regelmäßig zu Arbeitstreffen zusammenkommt.

Im Jahre 1992 wurde mit MPEG-1/Audio [18] ein Standard zur komprimierten Darstellung von Zweikanal-Audiosignalen mit einer Abtastrate von 32, 44.1 oder 48 kHz bei Bitraten von 64-384 kbps zur Verfügung gestellt. Als Referenz wird hier von einer linearen Quantisierung mit 16 Bit Auflösung ausgegangen. Insbesondere ist hier bei einer Datenrate ab 128 kbps (Kompression von 1:12), je nach gewählter Codierkomplexität, eine Wiedergabe in CD-Qualität möglich. Die anderen Komplexitätsstufen liefern ebenfalls qualitativ sehr hochwertige Hifi-Signale. Technisch wird hier mit einer Subbandtransformation, optional verfeinert durch eine nachgeschaltete Blocktransformation, gearbeitet.

Die Audiokomponenten des MPEG-2 Standards, BCC [20] bzw. ACC [21] (*Backwards Compatible Coding* bzw. *Advanced Audio Coding*), erweitern diese Funktionalität um die Möglichkeit, auch verschiedene niedrigere Abtastraten und maximal 48 Tonkanäle zuzulassen. Bei Gewährleistung einer immer noch sehr guten Qualität der Audiosignale (Kompression von ca. 1:24), finden diese Erweiterungen z.B. bei audiovisuellen Konferenzen oder Internetapplikationen Anwendung. Datenströme nach dem BCC-Teil des Standards sind dabei wechselseitig kompatibel zu MPEG-1, d.h. Decoder des einen Standards können aus dem Datenstrom des anderen Standards stets ein sinnvolles Signal reproduzieren. Da sich die Rückwärtskompatibilität als hinderlich beim Erreichen niedriger Datenraten in der gewünschten Qualität herausstellte und beim gleichzeitigen Ausnutzen von Maskierungseffekten mehrerer Kanäle teilweise Artefakte hörbar sind (sogenannte *unmasking* Effekte), wurde zusätzlich der nicht mehr zu MPEG-1 kompatible AAC entwickelt, und 1997 standardisiert. Die Kompressionsraten lassen sich bei Mehrkanalton nicht mehr so einfach mit den Standardkompressionsraten vergleichen, da durch Ausnutzung von Redundanzen innerhalb der Bänder extreme Einsparungen verzeichnet werden können. Umfangreiche Hörtests haben gezeigt, daß der AAC bei der Codierung von fünf Kanälen voller Bandbreite bei Bitraten von bis zu 256 kbps eine Qualität liefert, die perzeptuell nicht vom Original zu unterscheiden ist [4]. Der AAC arbeitet mit einer adaptiven Blocktransformation, die zur Vermeidung von Präechos von einer „normalen“ Blocklänge auf eine Verarbeitung mit kurzen Blöcken umgestellt werden kann. In einer Konfiguration für skalierbare Bandbreiten ist die Vorschaltung einer 4-Band Subbandtransformation vorgesehen.

Die alleinige Berücksichtigung der unteren Subbänder erlaubt so die Rekonstruktion einer bandbreitenreduzierten, aber immer noch sinnvollen Version des Audiosignals.

Um die Audiokomponente von MPEG-4 zu erläutern, soll zunächst auf das grobe Ziel von MPEG-4 eingegangen werden. Abstrakt geht es um die Darstellung *elementarer Medien* (engl. *Media Objects*), deren Zusammenfassung zu sogenannten *audiovisuellen Szenen* und einer möglichen benutzerseitigen Interaktion hiermit. Elementare Medien können hierbei beliebige visuelle, auditive oder audiovisuelle Daten sein. Diese können wiederum entweder natürlich oder synthetisch erzeugt worden sein. Insgesamt soll hiermit eine einheitliche Darstellungsweise für audiovisuelle Komponenten oder Szenarien in der Multimedialandschaft geschaffen werden. Da eine komplexitäts- oder qualitätsbasierte Skalierung heutzutage äußerst wünschenswert ist, kapselt MPEG-4 verschiedene bekannte und einige noch zu nominierende Standards. Auf der Audioseite soll so zwischen der Verwendung von (natürlich erzeugten) Audiosignalen mit Bitraten von 2 bis über 64 kbps ausgewählt werden können. Darüber hinaus ist für sehr niedrige Bitraten eine Schnittstelle für synthetische Audiosignale vorgesehen.

Im Falle der natürlich erzeugten Audiosignale wird der Bitratenbereich (der natürlich stets

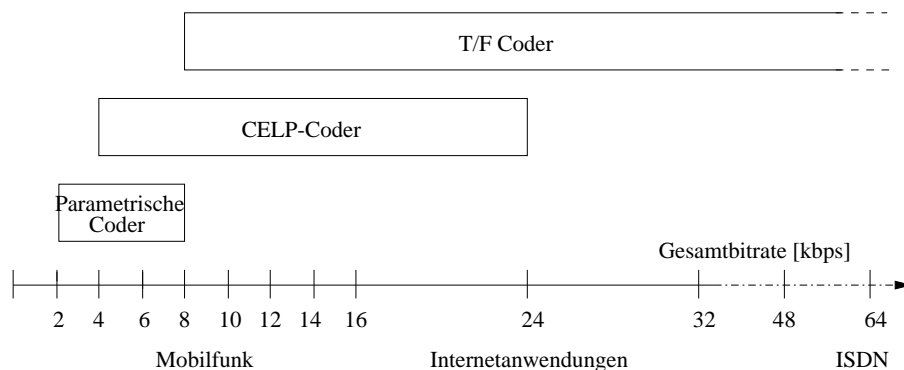


Abbildung 2.5: Die drei in MPEG-4 relevanten Codecklassen für natürliche Audiosignale und mögliche Verwendungsbereiche.

zusätzlich von der ursprünglichen Abtastrate abhängt) von den in Abb. 2.5 dargestellten drei Codeckfamilien der parametrischen-, CELP- (Code Excited Linear Prediction) und T/F- (Time/Frequency) Codecs abgedeckt. Während den ersten beiden Familien das klassische Spracherzeugungsmodell zugrunde liegt, sind die T/F-Coder den Transformations- oder Subbandcodierern zuzurechnen. Im Fall der für uns interessanten T/F-Coder sind zwei Transformationscodierverfahren, MPEG-AAC und TwinVQ [25], zur Verwendung im Standard vorgesehen.

Die synthetischen Audiokomponenten bestehen neben der Text-zu-Sprachsynthese und der Grundfunktionalität einer Wavetable-Synthese im Stile von MIDI [36] aus einem *Structured Audio*-Teil. Hier wird es ermöglicht, mit einer standardisierten Sprache (der *Structured Audio Orchestral Language, SAOL*) Algorithmen zur Erzeugung beliebiger synthetischer Musiksignale anzugeben. Ein MPEG-4-fähiges Wiedergabegerät wird somit in eine Konfiguration versetzt, in der es „Noten“-Befehle, die in einer zweiten Sprache (der *Structured Audio Score Language, SASL*) formuliert werden, zu Audiosignalen verarbeiten kann. Im grundlegenden Unterschied zu MIDI ist hierdurch das erzeugte digitale Audiosignal an jedem MPEG-4 Decoder eindeutig festgelegt und nicht individuellen Synthesevorgängen unterworfen.

Anders als bei MPEG-1 und 2 standardisiert MPEG-4 keine (Codier- oder) Decodier-techni-

ken. So sind z.B. hochwertige Verfahren für das Structured Audio Coding Gegenstand momentaner Forschungsaktivitäten [53, 52]. Analog wird der im Jahr 2001 zur Fertigstellung vorgesehene MPEG-7 Standard keine Datenverarbeitungsverfahren an sich spezifizieren. In MPEG-7 geht es darum, einheitliche Beschreibungsweisen für bestimmte multimediale Informationen festzulegen [24]. Im Beispiel Audio könnten diese Informationen z.B. textuelle Stichworte zu Musikstücken sein, mit denen auf bestimmte Gefühlsausdrücke oder Stimmungen hingewiesen wird, es kann sich aber einfach auch nur um Informationen zu den jeweiligen Komponisten oder Dirigenten handeln. Mit Hilfe einheitlicher Beschreibungsschnittstellen soll eine zielgerichtete Suche in multimedialen Dokumenten signifikant erleichtert oder teilweise erst effizient ermöglicht werden. Wiederum übernimmt MPEG nur die Schnittstellenspezifikation - die Arbeitsweise einer solchen Suche, oder gar die vorab zu leistende automatische Erstellung oder Extraktion einer Beschreibung aus dem audiovisuellen Material, wird nicht betrachtet. Diese Strategie stellt eine insbesondere für die Industrie attraktive Möglichkeit dar, zu einer vorgegebenen einheitlichen Schnittstelle im gegenseitigen Wettkampf proprietäre Lösungen zu entwickeln.

### 2.3.2 Forschungsaktivitäten und Trends

Die Forschungsaktivitäten auf dem Gebiet der Audiocodierung sind weiterhin sehr stark ausgeprägt, da neue Standards wie MPEG-4 und MPEG-7, die zunehmend Modularisierungs- und Erweiterbarkeitskonzepte verfolgen, sowohl Raum für kontinuierliche Verbesserungen als auch für vollständig neue Codiertechniken lassen.

Wünschenswerte Eigenschaften von Audiokompressionsverfahren sind

- *Skalierbarkeit* bezüglich der Bitrate. Hiermit wird allgemein die Veränderbarkeit der Bitrate bei gleichzeitiger proportionaler Änderung der perzeptuellen Qualität bezeichnet. Erste Ansätze sind hier
  - Bandbreitenreduktion durch Aufteilung in mehrere Subbänder (s.o.) oder
  - Verwendung verschiedener auf bestimmte Kompressionsraten abgestimmte Konfigurationen.

Solche Ansätze sind allerdings für viele Anwendungen nicht flexible genug. Für Sprach- und Breitbandssignale wurde von Tang et al. ein auf der CELP-Codierung aufbauendes Verfahren vorgeschlagen [57].

- *Toleranz gegenüber Übertragungsfehlern* verschiedener Art. Hier sind sowohl Bitfehler als auch mögliche Paketfehler (Verlust ganzer Datenblöcke) gemeint. Im Falle von Paketverlusten spielt hier auch die Skalierbarkeit eine Rolle, da diese eine geeignete Paketeinteilung bestimmen kann, so daß selbst bei Paketverlusten noch sinnvolle Signale rekonstruiert werden können.
- *Tandemfähigkeit*. Hiermit ist die im nächsten Kapitel detailliert beschriebene kaskadierte Anwendbarkeit eines Audiocodecs gemeint,

unter gleichzeitiger Gewährleistung der eher „klassischen“ Anforderungen an eine

- niedrige Bitrate,
- geringe Codierverzögerung, und

- geringe Coderkomplexität.

Gerade die simultane Erfüllung dieser letzten drei Anforderungen ist bisher nicht zufriedenstellend möglich.

In der Entwicklung der Audiocoders für hochqualitative perzeptuelle Qualität lassen sich einige deutliche Trends ablesen. Konkurrieren zwar immer noch die Subband- mit den Transformationscodierern, so zeigt sich doch deutlich, daß eine hohe Spektralaufösung sehr von Vorteil ist. Daher verwenden viele Subbandcodierer zusätzlich weitere Transformationsstufen. In vielen Coder haben sich modifizierte Cosinustransformationen bestimmter Blocklängen durchgesetzt (z.B. AC-2, AC-3, MPEG-AAC, TwinVQ etc.). Dabei ist eine adaptive Wahl der Blocklänge zur Vermeidung von Präechos bei hohen Kompressionsraten offenbar unabdingbar. Die Blocklängen werden dann so angepaßt, daß der Präecho-Effekt in jedem Fall durch die Rückwärtsmaskierung verdeckt wird. Psychoakustische Modelle werden häufig in sehr ähnlichen Varianten verwendet. Wie genau diese allerdings jeweils arbeiten und ob vielleicht noch weiterer Spielraum zur Datenreduktion gegeben ist, ist nur selten dokumentiert. Zur Quantisierung wurden zahlreiche Kombinationen verschiedener Strategien (Skalarquantisierer, Vektorquantisierer, prädiktive Quantisierer etc.), oft im Zusammenspiel mit nachfolgender Entropiecodierung, vorgeschlagen. Allerdings hat sich keine eindeutig zu bevorzugende Strategie herauskristallisiert.

Zwei Beispiele zeigen Audiocoders mit einer vom „Üblichen“ etwas abweichenden Struktur: Sinha und Johnston [56] schlagen die Kombination eines Transformations- und eines Waveletcodierers vor, um so deren jeweilige Stärken besser ausnutzen zu können. Tatsächlich eignen sich Transformationen üblicher Transformationscodierer sehr gut zur Codierung relativ stationärer Signalanteile, während Wavelets oder Wavelet Packets gerade bei stark transienten Signalkomponenten Stärken aufweisen. Hörtests [56] bestätigen die Ergebnisse der Autoren bei Bitraten um 64 kbps.

Ein ganz anderer Ansatz wird in dem oben bereits erwähnten TwinVQ-Coder (Transform-Domain Weighted Interleave Vector Quantization) verwendet [25]. Nach einer Blocktransformation wird das Signalspektrum durch ein simultan mittels einer LPC-Analyse [9] gewonnenes Spektrum dividiert. Dies glättet das Spektrum, behält jedoch die Feinstruktur bei. Die LPC-Koeffizienten werden zum Decoder übertragen. Die Feinstruktur wird in einem nächsten Schritt durch einen aus den vorhergehenden Rahmen prädizierten Feinstrukturvektor dividiert. Dieser Schritt ist rückwärtsadaptiv und kann im Decoder ohne eine zusätzliche Übertragung von Daten in derselben Weise ausgeführt werden. Zur Vektorquantisierung werden die zu quantisierenden Komponentenvektoren  $x$  in ihre Phasen  $(\downarrow M)[x], (\downarrow M) \circ T_1[x], \dots, (\downarrow M) \circ T_{M-1}[x]$ , für geeignetes  $M$ , zerlegt. Die Codebuchvektoren werden anhand einer perzeptuellen Gewichtung, ähnlich dem Verfahren innerhalb des G.728-Standards [14], ermittelt. TwinVQ-Codierung wird von Yamaha im SoundVQ-Coder eingesetzt und ist neben den MPEG-Standards der einzige inzwischen weiter verbreitete Audiocodec dieser Kategorie im Internet-/PC-Bereich.

Neuere Fragestellungen in der Audiokompression sind unter anderem

- verstärkte Inhaltsbasiertheit der Kompression. Wie kann Zusatzinformation (Noten, Notendauern etc.) effektiv *und* effizient zur Erreichung sehr hoher Kompressionsraten verwendet werden? Dies ist eine in gewisser Weise mit dem Structured Audio Coding verwandte Fragestellung, die allerdings auf der Seite der natürlich erzeugten Audiosignale anzusiedeln ist.

- Einsatz von komprimiertem Audiomaterial in Audioarchiven und digitalen Musikbibliotheken. Hierbei möglicherweise entstehende Generationseffekte und Fragestellungen im Zusammenhang mit Editieroperationen werfen zahlreiche Fragen auf.

Auf dem Gebiet der psychoakustischen Audiocodierverfahren kommt es häufig auf die exakte Abstimmung der einzelnen möglichen Coderparameter (Filterfrequenz und -impulsantwort, Filterlänge, Blocklänge, Codebuchgröße, Quantisierer, psychoakustisches Modell u.v.a.) an, insbesondere zur Ermöglichung sehr hoher Kompressionsraten bei perzeptuell transparenter Qualität. Daher werden üblicherweise nur die Grobversionen der Codier- und Decodieralgorithmen veröffentlicht. Da sich in vielen Bereichen der Audiokommunikation geeignete Standards erst noch fest etablieren müssen und somit die Konkurrenzsituation in diesem äußerst anwendungsträchtigen Gebiet sehr groß ist, ist dies auch verständlich. Andererseits erschwert die mangelnde Verfügbarkeit entsprechenden Quellcodes somit mögliche Weiterentwicklungen innerhalb verwandter oder resultierender Fragestellungen. Daher wurde als Gerüst für die im Rahmen dieser Arbeit vorgestellten Implementationen eine verfügbare Realisierung eines MPEG-1 Codecs [40] verwendet.

## Kapitel 3

# Digitale Generationseffekte

Im Zusammenhang mit analogen Medien sind Qualitätsverluste, wie sie beim wiederholten Kopieren und Vervielfältigen auftreten, hinreichend bekannt. Beispiele solcher Medien sind Kompaktkassetten, Tonbänder, Videokassetten oder Langspielplatten. Bei der Verwendung nichtprofessioneller Hardware ist hier oft schon die Kopie der Kopie stark in ihrer Qualität beeinträchtigt. Hinzu kommt, daß sich meist sogar das Original mit der Zeit physikalisch abnutzt und somit die Wiedergabequalität sinkt. Der Vorgang des Kopierens soll hier im Sinne einer Signalverarbeitung als Transformation  $T$  betrachtet werden. Die einzelnen Signale und deren Kopien heißen *Generationen*. Das Originalsignal  $x$  ist die 0. Generation, die erste Kopie  $Tx$  die 1. Generation und entsprechend die  $n$ -te Kopie  $T^n x$  die  $n$ . Generation.

Im Zeitalter digitaler Medien haben sich einige Grundvoraussetzungen grundlegend geändert. Insbesondere erlaubt die Möglichkeit, sehr große Mengen diskreter Werte schnell und mit sehr geringer Fehlerwahrscheinlichkeit auf bestimmten Medien zu speichern und wieder auszulesen, prinzipiell eine verlustfreie Datenhaltung. Für die Audiosignalverarbeitung bedeutet dies die Möglichkeit, Audiosignale (fast) beliebig hoher Qualität speichern sowie beliebig oft und ohne Verluste kopieren zu können. Auch Kopien beliebig hoher Generationen entsprechen exakt dem Original. Auch die Verschleißrate der Medien (CDs, DAT-Bänder, Festplatten, DVDs, etc.) ist geringer geworden. Im Gegensatz zu analogen Medien haben Beschädigungen digitaler Medien jedoch meist etwas unangenehmere Auswirkungen. Sind im analogen Fall zunächst nur leicht hörbare Qualitätseinbußen zu verzeichnen, so können fehlerhafte oder beschädigte digitale Datenträger leicht ganze Audiostücke oder gar den gesamten Datenträger unbrauchbar machen. Insgesamt stellt die Einführung digitaler Medien im Audiobereich jedoch einen Quantensprung in Richtung Datenkonservierung, schneller Zugriff und effiziente Verarbeitung dar.

Die Übertragung bzw. Speicherung qualitativ hochwertiger audiovisueller Daten erfordert hierbei eine sehr hohe Kapazität auf Seiten des Übertragungskanals bzw. Speichermediums. So enthält eine handelsübliche Audio-CD bei ca. 70 Minuten Laufzeit ca. 700 Megabyte Daten, im Falle der DVD sind es sogar einige Gigabyte. Zur Laufzeit fallen bei der CD ca. 1400 kBit Daten pro Sekunde an. Im Vergleich hierzu stellt ein ISDN B-Kanal lediglich eine Kapazität von 64 kBit pro Sekunde zur Verfügung, die Datenübertragungskapazität im Internet ist oft noch wesentlich geringer. Weiterhin ist meist keine konstante Übertragungskapazität gewährleistet, hinzukommen gegebenenfalls Datenverluste, die durch nochmaliges Senden oder entsprechende Redundanz korrigiert werden müssen. Die in Kapitel 2 vorgestellten Audiocodierer tragen durch ihre zum Teil sehr hohen Kompressionsraten wesentlich zur

Lösung dieser Kapazitätsprobleme bei. Die bei diesen Verfahren zwangsläufig auftretenden Verluste sind dabei gerade so gewählt, daß sie, zumindest in der ersten Generation, nicht wahrnehmbar sind. Allerdings kann jeder weitere Kompressions/Dekompressions-Schritt die Audioqualität wesentlich beeinträchtigen, wir sprechen von (digitalen) *Generationseffekten*. Bezeichnet man die Encoderoperation mit  $E$  und die Decoderoperation mit  $D$ , so ist in obiger Terminologie  $T = DE$ . Die  $n$ -te Generation ist damit durch

$$T^n x = \underbrace{DE \cdots DE}_{n\text{-mal}} x$$

gegeben. Wir bezeichnen  $DE$  als *Codecestufe* und die Hintereinanderschaltung mehrerer Codecestufen als *Codec-Kaskade*. Da die Rekonstruktion konzeptuell keine verlustbehafteten Schritte mehr enthält, wird manchmal auch das Signal  $E(DE)^{n-1}x$  als  $n$ -te Generation von  $x$  bezeichnet.

In diesem Kapitel wird genauer untersucht, wodurch die Abweichungen höherer Generationen vom Original verursacht werden. Dazu werden zunächst anhand eines allgemeinen Codec-Modells mögliche Teilursachen analysiert. Die Fallstudie eines MPEG-1 Codecs zeigt, daß schon bei kleinem  $n$ , z.B.  $n = 2, 3$  oder  $4$ , die entsprechenden Generationen eines Audiosignals im Gegensatz zum Original stark perzeptuell beeinträchtigt sein können. Abschließend werden Möglichkeiten zur Vermeidung und Abschwächung einiger der aufgezeigten Probleme beschrieben. Hieraus motiviert sich dann das weitere Vorgehen zur Vermeidung von Generationseffekten.

### 3.1 Fehleranalyse am Codec-Modell

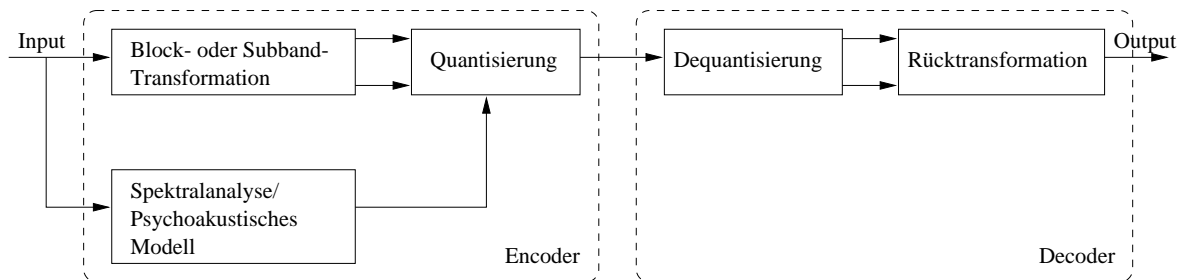


Abbildung 3.1: Vereinfachtes Modell eines Subband- oder Transformationscodierers.

In Abb. 3.1 ist die allgemeine Form eines psychoakustischen Subband- oder Transformationscodierers angegeben. Auf die genauere Form der einzelnen Komponenten wird in Kapitel 5 im Zusammenhang mit der Beschreibung eines Codecs zur Vermeidung von Generationseffekten eingegangen. Anhand des vorliegenden allgemeinen Modells können die wichtigsten Fehlerquellen der gebräuchlichsten Codierer analysiert werden.

Das digitale Signal wird encoderseitig durch eine Block- oder Subbandtransformation (im folgenden einfach als Subbandtransformation bezeichnet) in Spektralkomponenten oder Subbandsignale zerlegt. Synchron werden mit Hilfe von Spektraltransformationen (z.B. DFT, DCT, etc.) Parameter eines psychoakustischen Modells berechnet. Mit deren Hilfe führt der Quantisierer den verlustbehafteten Teil der Codierung durch. Decoderseitig wird nach der Rückquantisierung eine inverse oder Umkehrfilterbank zur Rekonstruktion eines Zeitsignals

verwendet. Mit Rückquantisierung wird hier die Abbildung eines übertragenen Codebuchindex auf ein Datenwort oder einen Datenwortvektor bezeichnet.

### 3.1.1 Fehlerquellen

Bezeichnen  $T$  die Subbandtransformation,  $\tilde{T}$  die entsprechende Umkehrtransformation sowie  $S$  die Spektraltransformation, so sind innerhalb einer Codecstufe folgende Fehlermöglichkeiten gegeben:

1. Quantisierung der Subbandsignale. Dieser Schritt ist i.a. verlustbehaftet. Die Art und das Ausmaß der Quantisierung wird durch die psychoakustischen Parameter gesteuert.
2. Rundungsfehler bei  $T$ ,  $\tilde{T}$ ,  $S$  und der Berechnung des psychoakustischen Modells. Beim psychoakustischen Modell werden oft Matrixoperationen zur Berechnung des Spread-Spektrums verwendet.
3. Aliasing, das durch  $T$  entsteht und bei PR-Filterbänken durch  $\tilde{T}$  neutralisiert wird. Fehler entstehen, wenn dieses Aliasing auf den Subbändern bei der Einbeziehung der Spektralanalyse, die auf dem Originalsignal durchgeführt wird, nicht beachtet wird.
4. Nicht perfekte Rekonstruktion (NPR Filter), z.B.  $\tilde{T}Tx = x + \varepsilon$  wo  $\|\varepsilon\| \ll \|x\|$ .
5. Kanal- oder Übertragungsfehler zwischen Encoder und Decoder. Solche Fehlerquellen sollen hier jedoch nicht weiter betrachtet werden.
6. Fehlerhaftes psychoakustisches Modell. Da die verwendeten Modelle lediglich von Approximationen des menschlichen Wahrnehmungsverhaltens ausgehen, kann es hier von Person zu Person zu Abweichungen kommen. Dieses Problem wird in gebräuchlichen Codierern momentan durch eine konservative Berechnung von Maskierungsschranken bzw. durch eine nicht vollständige Ausschöpfung derselben kontrolliert [61].
7. Fensterung bei der Spektralanalyse. Hier besteht eine gewisse Zeit-Frequenzunschärfe, die durch die Heisenbergsche Unschärferelation zu begründen ist.
8. Fehlende Translationsinvarianz von  $T$  und  $\tilde{T}$ . Da die meisten Filterbank- oder Blocktransformationen Blocktoeplitzmatrizen entsprechen, ist hier lediglich Blocktranslationsinvarianz gegeben. Die erheblichen Auswirkungen dieser Eigenschaft werden in der folgenden Fallstudie besonders deutlich.

Es ergeben sich folgende Abhängigkeiten und Gewichtungen, die insbesondere durch die Fallstudie des nächsten Abschnitts unterstützt werden:

- Der Quantisierungsfehler 1. trägt dominant zur Degenerierung bei. Er ist für den größten Teil der Veränderung des Signals von Generation zu Generation verantwortlich.
- Die Arithmetik- und Rekonstruktionsfehler 2. und 4. sind im Vergleich zu 1. klein, bewirken aber von Generation zu Generation für unsere Anwendungen signifikante Abweichungen, die kontrolliert werden müssen.
- Die Fehler des psychoakustischen Modells 6., der Zeit-Frequenz-Unschärfe 7. und der fehlenden Translationsinvarianz wirken sich indirekt auf den Quantisierungsfehler 1. aus.

Aliasingfehler 3. sind stark abhängig von der verwendeten Filterbank und werden im folgenden kurz behandelt.

### 3.1.2 Fehlquantisierung durch Aliasing

Wir betrachten eine 2-Band Multiratenfilterbank, die ein Signal  $x$  in Subbandsignale  $s_0$  und  $s_1$  zerlegt. Quantisierungsrauschen  $q_0$  und  $q_1$  wird als additiver Zufallsprozeß aus unabhängigen gleichverteilten Zufallsvariablen modelliert. Die Rücktransformation ergibt das rekonstruierte Signal  $\tilde{x}$  mit

$$\tilde{x} = (T_{-1} \circ c_{f_0} \circ (\uparrow 2))(s_0 + q_0) + (c_{f_1} \circ (\uparrow 2))(s_1 + q_1) \quad (3.1)$$

$$= x + (T_{-1} \circ c_{f_0} \circ (\uparrow 2))q_0 + (c_{f_1} \circ (\uparrow 2))q_1, \quad (3.2)$$

wobei die Linearität der Filterbankoperationen ausgenutzt wurde. Außerdem wurde angenommen, daß die Filterbankoperation ansonsten fehlerfrei ist. Anhand eines Beispiels soll der mögliche entstehende Fehler illustriert werden.

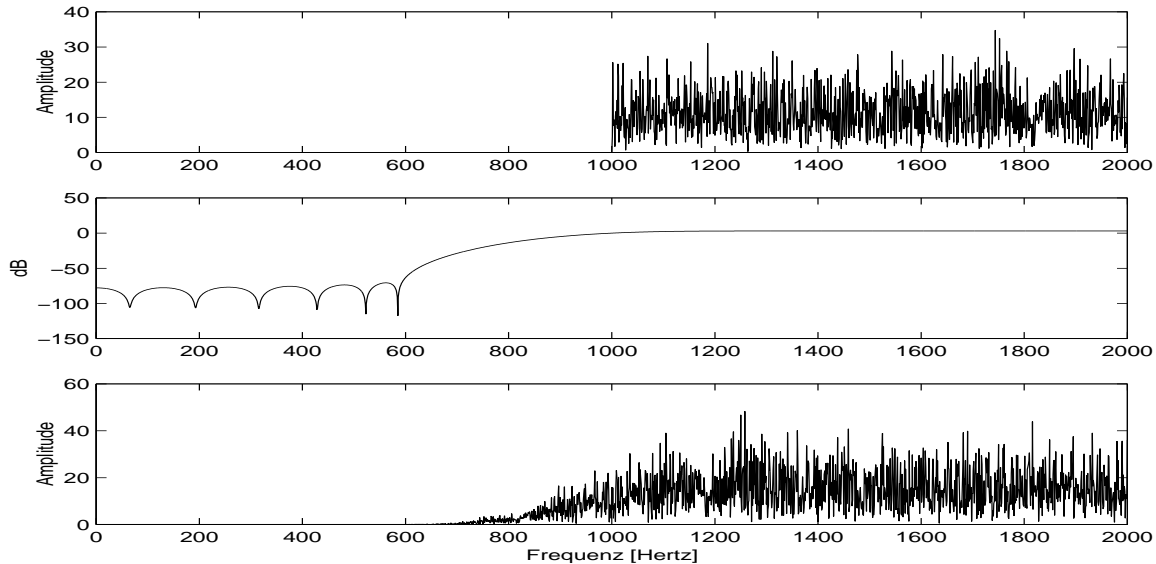


Abbildung 3.2: Oben: Test-Fehlersignal, welches mit der Filterbank zurücktransformiert wurde. Mitte: Frequenzantwort des Hochpaßfilters der 2-Band Filterbank. Unten: Spektrum des rekonstruierten Signals. Entsprechend des Frequenzverhaltens des Hochpaßfilters ist ein Aliasing der hohen Frequenzen in das Tiefpaßband zu erkennen.

Konstruiert man ein 2-Band Subbandsignal, welches im unteren Band konstant Null und im oberen Band das Zufallssignal ist (siehe Abb 3.2 oben), kann die Rücktransformierte unter Anwendung einer inversen 2-Band Filterbank betrachtet werden. In der Abbildung ist im Mittelteil die Frequenzantwort des Hochpaßfilters abgebildet. Dies ist ein Vaidyanathan Filter mit 24 Koeffizienten. Die Filterbank ist so konstruiert, daß sie möglichst gute Stopbanddämpfung besitzt. Im unteren Teil, der das Spektrum des rekonstruierten Signals zeigt, ist deutlich ein Aliasing auf das Tiefpaßband zu erkennen.

Extremer tritt dieses Phänomen bei der Verwendung von Filterbänken auf, die nicht hinsichtlich der Stopbändeigenschaften optimiert sind, wie z.B. die populären Waveletfilter nach Daubechies [8]. Diese Filter liefern zwar perfekte Rekonstruktion, sind sogar orthonormal, haben aber aufgrund der Beschränkung der Nullstellen der Frequenzantwort des Filters auf

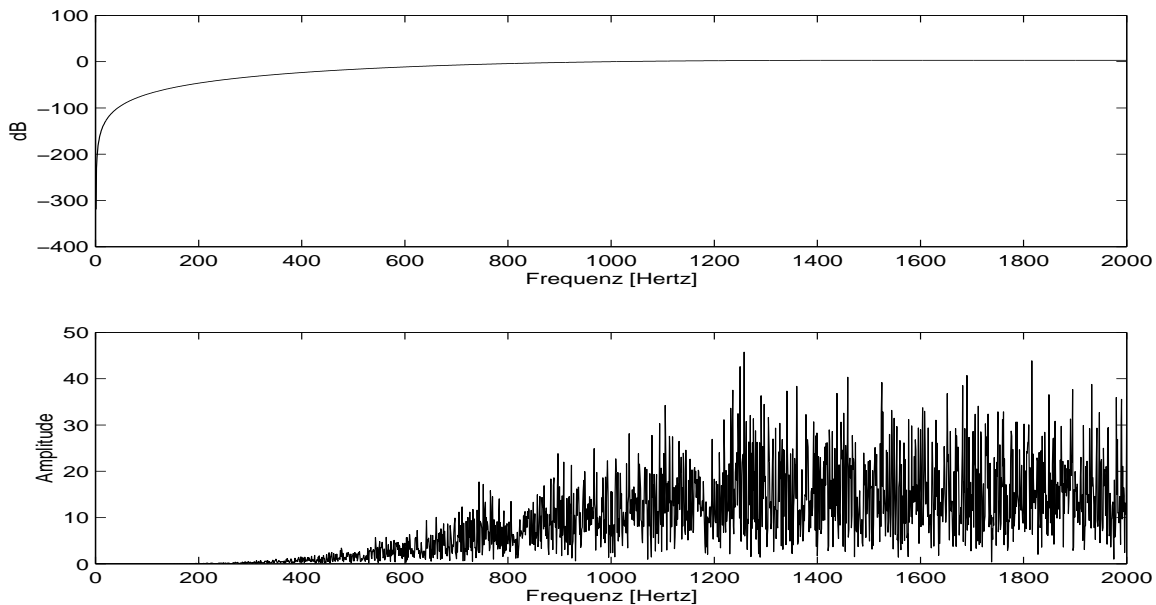


Abbildung 3.3: Der Abb. 3.2 entsprechende Darstellung für das Daubechies Wavelet mit 8 Koeffizienten. In diesem Fall ist das Aliasing wesentlich stärker.

$\pi$  nur schlechte stopbanddämpfende Eigenschaften. Abbildung 3.3 wiederholt den mittleren und unteren Teil von Abb. 3.2 für den Fall eines Daubechies Filters mit 8 Koeffizienten.

## 3.2 Fallstudie: MPEG-1 Codec

In der Praxis auftretende Generationseffekte wurden anhand eines MPEG-1 Codecs untersucht. Die dabei hauptsächlich analysierte Konfiguration liefert eine Kompression mit einer Bitrate von 128 kbps für Stereosignale mit einer Abtastrate von 44.1 kHz (Kompressionsrate 1 : 12). Verwendet wurde ein Layer II-Codec. Ebenfalls untersucht wurde dieselbe Konfiguration auf einem Layer III-Codec, die Verwendung eine Bitrate von 196 kbps mit dem Layer II-Codec, sowie die Layer II-Konfiguration mit einer Abtastrate von 48 kHz. Da hier die beobachtbaren Effekte sehr ähnlich sind, wurde für intensivere Tests nur noch die erstgenannte Konfiguration verwendet. Im Falle des Layer III-Codecs und der Bitrate von 192 kbps treten die Generationseffekte bei einigen Stücken mit einer Verzögerung (um ein bis drei Generationen gegenüber Layer II, 128 kbps) auf, sind aber im allgemeinen ebenso stark zu verzeichnen.

### 3.2.1 Subjektive Tests

Als Testmaterial für die Analyse der Generationseffekte wurden unter anderem die in Tabelle 3.4 aufgelisteten Audiostücke verwendet. Hinzu kommen die in Tabelle 6.6 aufgelisteten Stücke, die im Rahmen der Befragung von Testpersonen zur in Kapitel 5 behandelten Codec-evaluation verwendet wurden. Von diesen Stücken wurden das Kastagnettenstück, die beiden Sprachstücke und das E-Gitarren/Rhythmus-Stück detaillierter untersucht.

Für die Untersuchungen wurden von den Testsignalen jeweils Versionen bis zur 25. Generation, zur Überprüfung asymptotischer bzw. Langzeitphänomene teilweise sogar Versionen

Nr.	Beschreibung	Datei
1	Sophran und Orchester	096sophr
2	Pop (Roger Rabbit)	105rabit
3	Sophran, nur vokal	079sophr
4	Alt, nur vokal, Stück wie 3	080alt
5	Tenor, nur vokal, Stück wie 3	081tenor
6	Baß, nur vokal, Stück wie 3	082bass
7	Quartett aus 3-6	083quart
8	Saxophonstück	sax2

Abbildung 3.4: Auswahl verwendeter Teststücke für die Analyse der Generationseffekte.

bis zur 50. Generation erzeugt. Hierzu wurden die Stücke wiederholt in verschiedenen Räumlichkeiten und unter Verwendung von sowohl Kopfhörer- als auch Lautsprecherausrüstung abgehört. Zur Kontrolle der subjektiven Höreindrücke wurden einzelne Mitglieder der Arbeitsgruppe, Teilnehmer der Projektgruppen *Audiosignalverarbeitung* und *Digitale Musikbibliotheken*, Diplomanden und musikinteressierte Dritte zu verschiedenen Teststücken befragt. Bei den Untersuchungen wurden folgende Fragestellungen betrachtet:

1. Ab welcher Generation ist eine Änderung des Signals im Gegensatz zum Original wahrnehmbar?
2. Ab welcher Generation ist die Änderung des Signals klar vorhanden oder störend?
3. Mit welcher Geschwindigkeit tritt die Qualitätsabnahme auf?
4. Wird die Qualitätsabnahme stationär bzw. ändert sich die perzeptuelle Qualität ab einer bestimmten Generation nicht mehr?
5. Wie sind die Qualitätsbeeinträchtigungen beschreibbar?

Die Testergebnisse lassen sich für den überwiegenden Teil des Testmaterials folgendermaßen zusammenfassen:

1. Eine Beeinträchtigung der Signalqualität ist ab der zweiten bis dritten Generation bei fast allen beobachteten Stücken wahrnehmbar. Jedoch ist diese Beeinträchtigung i.a. nicht störend.
2. Die Klangqualität nimmt so ab, daß in der vierten bis sechsten Generation ein deutlicher Qualitätsverlust bemerkbar ist. Die Signalqualität ist meist störend beeinträchtigt.
3. Signale ab der achten Generation weisen bereits eine meist unzumutbare Qualität auf. Der Anstieg des Qualitätsverlustes verläuft dabei relativ graduell.
4. Ab der 15. bis zur 25. Generation tritt bei allen untersuchten Stücken ein Zustand gleichbleibender Qualität auf. Diese Wahrnehmung wird durch die Auswertung der Signalparameter des nächsten Abschnitts bestätigt.

5. Die Störungen werden als *Rauschen*, *Kratzen* oder *Verzerrung* wahrgenommen. Sie treten besonders stark an energiereichen oder laut ausgesteuerten Stellen der Signale auf. Die Störungen sind an Stellen mit percussiven Elementen oder kurzen energiereichen Übergängen innerhalb des Audiosignals besonders stark ausgeprägt.
6. Bei einigen Stücken konnte die Beobachtung gemacht werden, daß einige leichte Beeinträchtigungen des Originalsignals (leichte Artefakte, z.B. ein „schabendes“ Geräusch innerhalb eines Sprachstücks) in höheren Generationen verstärkt werden.

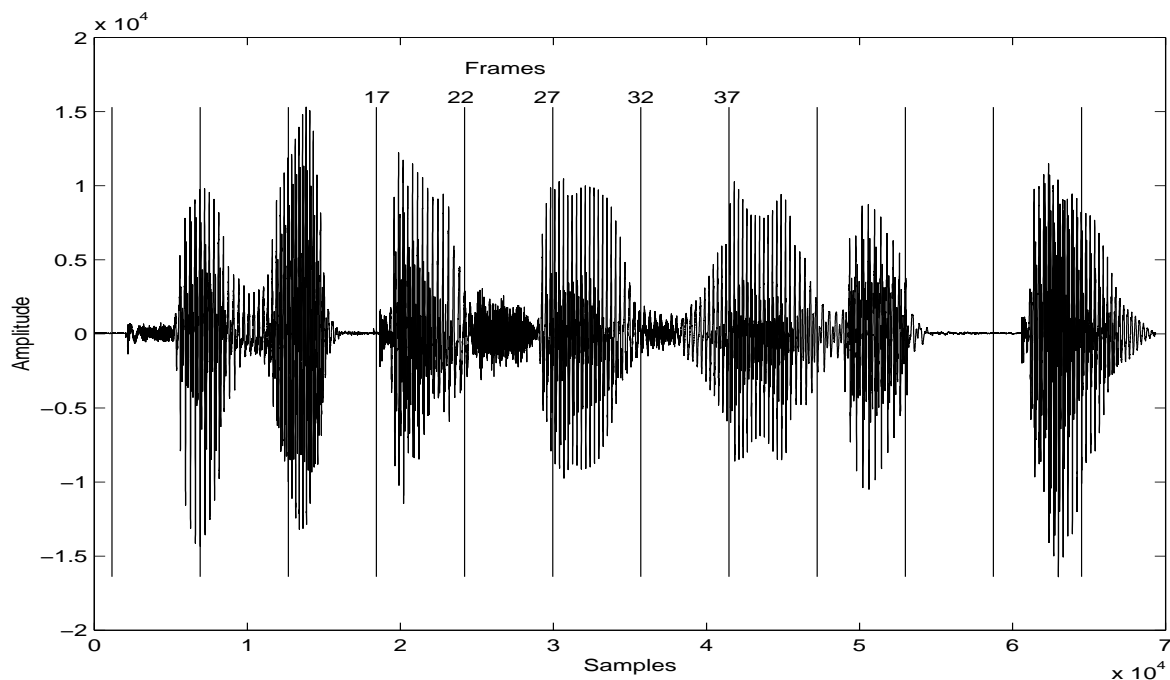


Abbildung 3.5: Ausschnitt des Sprachsignals 087speec mit eingezeichneten Rahmengenrenzen.

Diese Zusammenfassung kann allerdings nur eine grobe Einschätzung der allgemein auftretenden Degeneration geben. Einige wichtige Beobachtungen ergeben sich aus der Einzelbetrachtung einiger Teststücke:

- Einige Klassen von Signalen und Signalkomponenten degenerieren sehr wenig oder überhaupt nicht. Von den obigen Beispielen ist dies besonders stark am Solo-Sophranstück 3 und am Glockenspielstück aus Tabelle 6.6 beobachtbar. Charakteristisch für diese nur wenig degenerierenden Stücke und Signalanteile ist ihr stark tonaler Charakter.
- Ebenfalls sind Passagen sehr geringer Lautstärke nur wenig von der Degeneration betroffen, wie z.B. die Passagen im Sophranstück 1, in denen der Gesang nur leise vom Orchester begleitet wird.
- Mit steigender Generation des Kastagnettenstücks wird die Energie des Kastagnettenschlags von den Schlagzeitpunkten weg „verschmiert“ und damit die Lautstärke zu diesen Zeitpunkten gesenkt. Die Abweichung vom Original ist zwar deutlich zu bemerken, wird jedoch von einigen der befragten Personen als nicht störend empfunden.

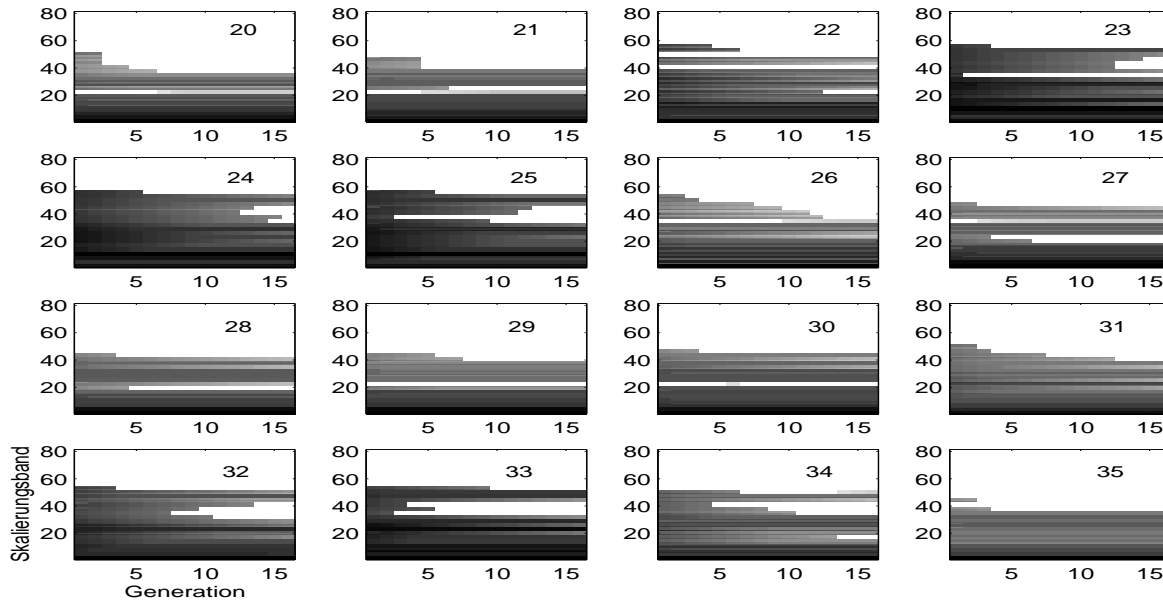


Abbildung 3.6: Degeneration der Signalenergie einzelner Rahmen des Sprachstücks aus Abb. 3.5 mit wachsender Generation. Die Veränderung der Subbandenergien in den nicht tonalen Signalanteilen (Rahmen 22-26 und 32-34) ist deutlich stärker als in den tonalen (Rahmen 20-21, 27-31 und 35).

Die Beobachtungen über das unterschiedliche Verhalten tonaler und nicht tonaler bzw. leiser und lauter Komponenten mit wachsender Generation, läßt sich aus der Funktionsweise des MPEG-Codecs erklären. Ist das Signal in seiner Gesamtheit energiearm (leise), so wird diese Art der Redundanz durch eine entsprechende Skalierung ausgenutzt und so schon indirekt eine wesentliche Reduktion der Bitrate erreicht. Die vorhandenen Signalkomponenten werden dann mit einer relativ hohen Auflösung quantisiert. Die Skalierung wird in Kapitel 4 genauer behandelt. Die Degenerationsanfälligkeit nichttonaler Komponenten läßt sich aus dem verwendeten psychoakustischen Modell erklären, das die Eigenschaft der menschlichen Wahrnehmung, daß rauschartige Signale besser maskieren als tonale, ausnutzt. Somit wird beim Auftreten rauschartiger Komponenten potentiell gröber quantisiert.

Dieser Effekt soll anhand eines konkreten Beispiels verdeutlicht werden. Abb. 3.5 zeigt die Wellenform eines Ausschnitts des Sprachsignals `087spec`. Die eingezeichneten senkrechten Markierungen bezeichnen jeweils die MPEG-Rahmengrenzen in Schritten von fünf Rahmen. Die Beschriftungen geben jeweils die Rahmennummer an. In Abb. 3.6 ist für die Rahmen 20-35 des Stücks das Verhalten der Signalenergie innerhalb der einzelnen Subbänder mit wachsender Generation angegeben. Die Generationen sind aufsteigend von links nach rechts, die Subbänder für aufsteigende Frequenzen von unten nach oben abgetragen. Die Energie ist logarithmisch als Graustufe wiedergegeben. Es ist deutlich zu erkennen, daß die Veränderung der Subbandenergien in den nicht tonalen Signalanteilen (Rahmen 22–26 und 32–34) stärker ist als in den tonalen (Rahmen 20 – 21, 27 – 31 und 35). In den tonalen Rahmen ist ein relativ konstantes Energieverhalten, in den nicht tonalen durch ein „Ausfransen“ erkennbare Energiedegenerationen zu verzeichnen.

### 3.2.2 Auswirkung von Translationen

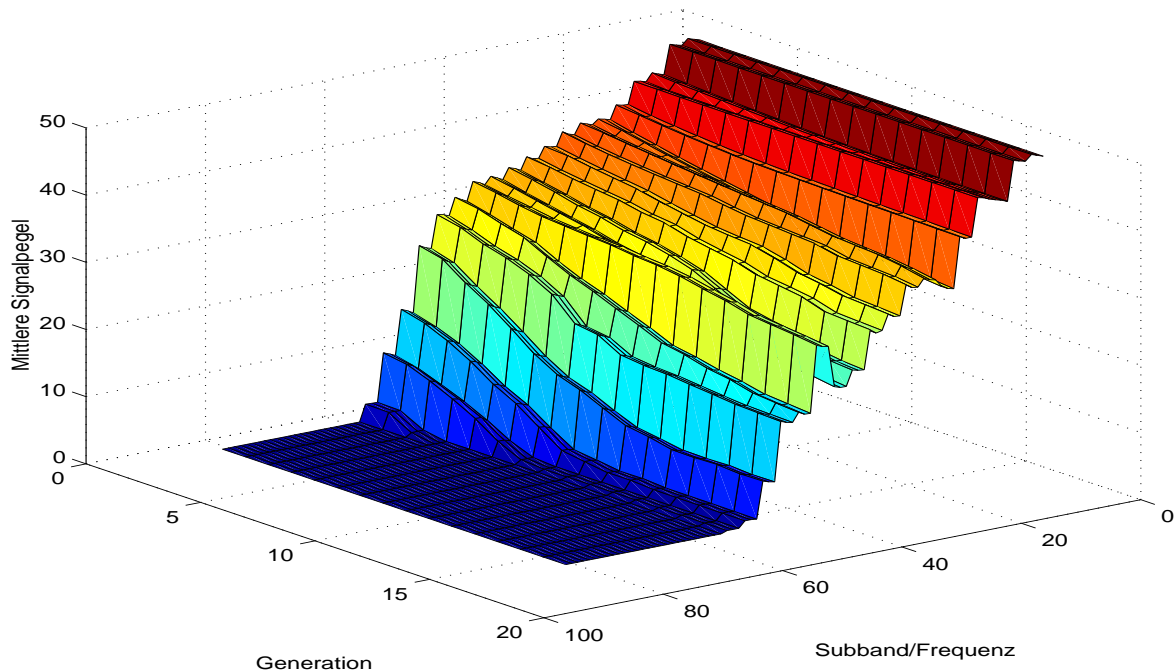


Abbildung 3.7: Entwicklung der mittleren Skalierungsfaktoren der Subbänder über 20 Generationen für das Popstück 104abba.

Offensichtlich ist die der MPEG-Codierung zugeordnete Gesamtoperation nicht translationsinvariant: Die psychoakustische Analyse, aufgrund derer die Quantisierung durchgeführt wird, wird in Layer II für jeden Rahmen nur einmal, mit Hilfe einer gefensterten Fouriertransformation der Länge 1024, ausgewertet. Dabei entsteht ein durch die Rahmenlänge bedingter Offset von 1152 Samples zwischen zwei Analysevorgängen. Wird anstelle des Signals  $x$  das Signal  $T_\ell x$  für  $1 \leq \ell < 1152$  codiert, sind somit andere Kompressionsergebnisse möglich. Weiterhin ist die Analyseoperation der 32-Band Multiratenfilterbank nur translationsinvariant unter Verschiebungen um 32 Samples: Falls für das  $i$ -te Subbandsignal  $s_i = (\downarrow 32) \circ c_{h_i} s$  gilt, so kann im allgemeinen  $T_k s_i = (\downarrow 32) \circ c_{h_i} T_\ell s$  nur geschlossen werden, falls  $\ell = 32 \cdot k$  gilt. Es stellt sich somit die Frage, auf welche Weise Translationen in die Analyse einbezogen werden sollen. Folgende vier Möglichkeiten werden dabei im weiteren untersucht:

- Translationen werden nicht beachtet, die  $n$ -te Generation wird also als  $(DE)^n x$  erzeugt. Dies ist eine realistische Annahme, wenn davon ausgegangen wird, daß Signale immer dateiweise oder als Ganzes verarbeitet werden. Ergebnisse dieser Art der Generationenbildung wurden oben beschrieben.
- Es wird zwischen den Codecstufen eine Synchronisation bezüglich des Filterbankdelays der MPEG-Filterbank  $F$  durchgeführt. Die Filterbank erfüllt (bis auf die NPR Eigenschaft)  $\tilde{F}F = T_{-481}$ . Wird also die  $n$ -te Generation als  $(T_{481}DE)^n x$  erzeugt, so kann jede Codecstufe *synchron* codieren. Tatsächlich schlagen Keyhl et al. [30] ein Verfahren vor, das den *codierten* Datenstrom mit Synchronisationsinformation anreichert, so daß die nächste Codecstufe eine synchrone Codierung vornehmen kann.

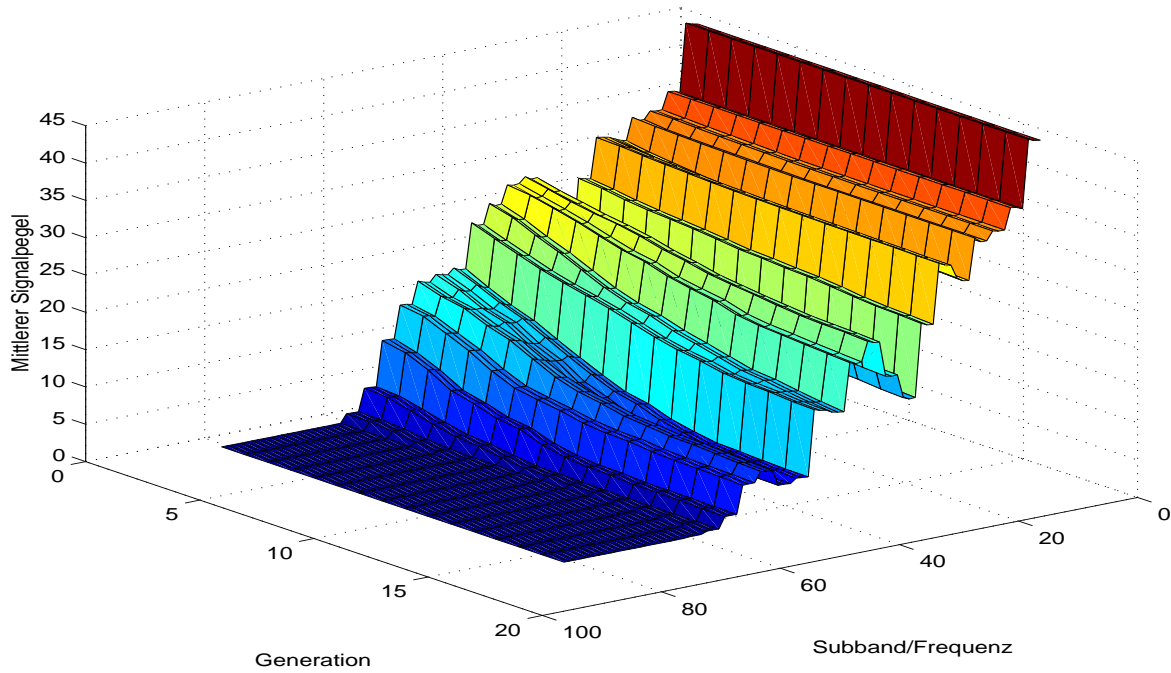


Abbildung 3.8: Entwicklung der mittleren Skalierungsfaktoren der Subbänder über 20 Generationen für das Sprachstück 087speech mit synchronisierender Translation um 481 Samples zwischen jeder Codecstufe.

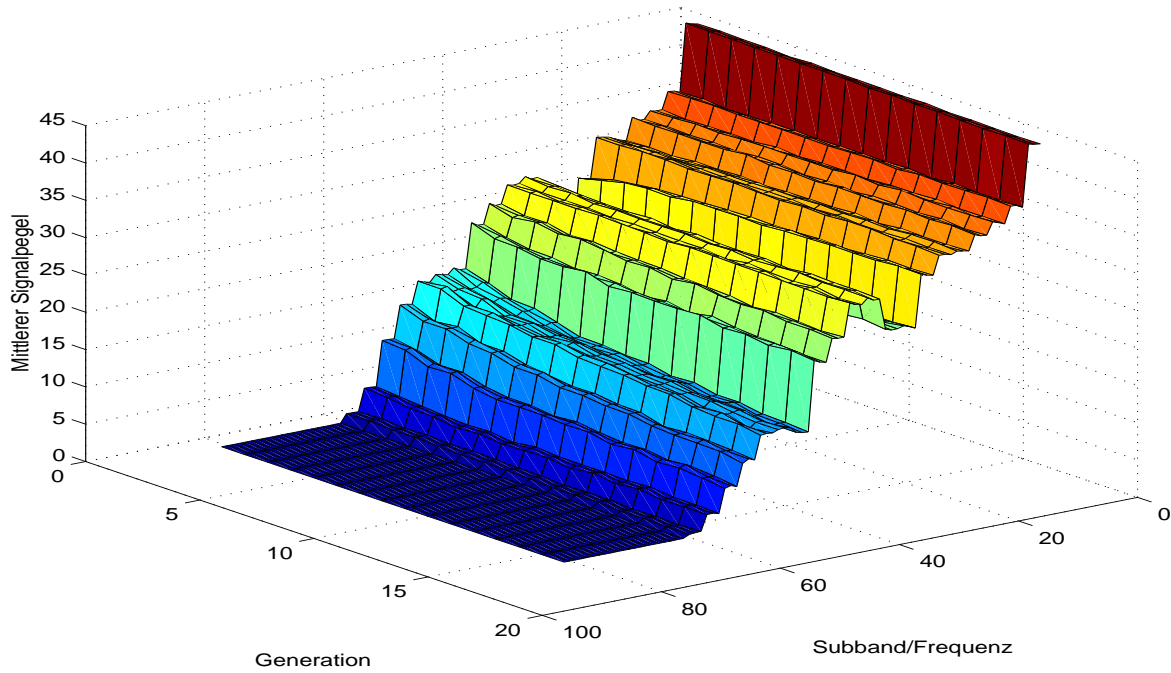


Abbildung 3.9: Entwicklung der mittleren Skalierungsfaktoren der Subbänder über 20 Generationen für das Sprachstück 087speech ohne synchronisierende Translation.

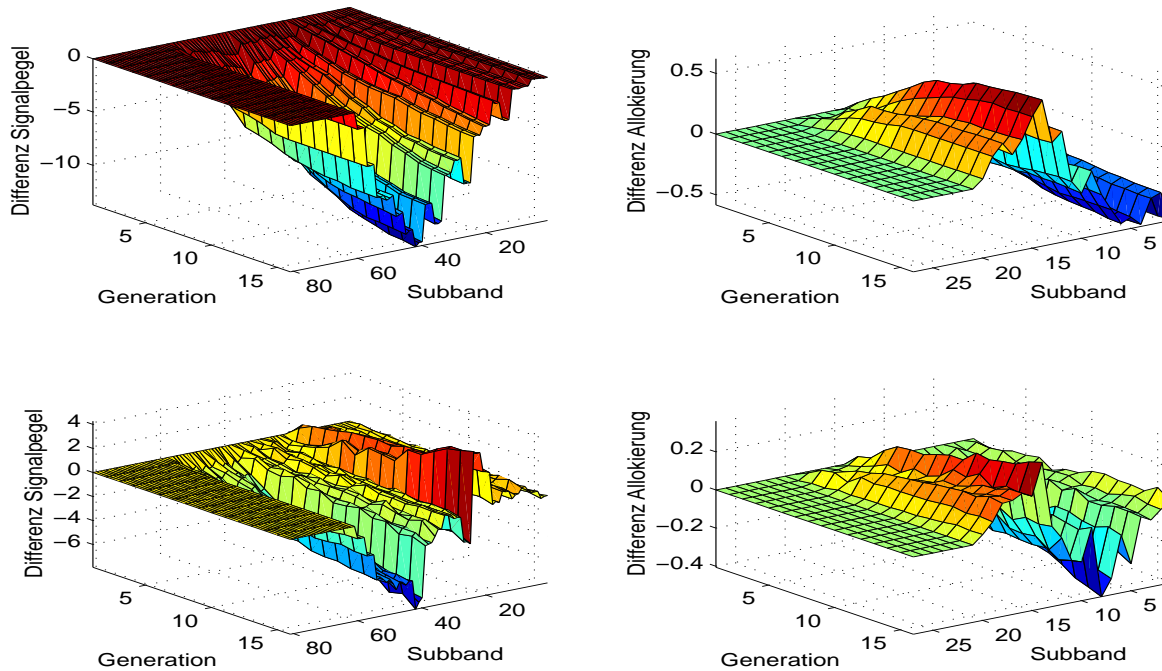


Abbildung 3.10: Entwicklung der mittleren Skalierungsfaktoren und Bitallokierung für 20 Generationen für das Sprachstück 087spec. Dargestellt sind jeweils die Differenzen zwischen der ersten und den folgenden Generationen.

- Im Hinblick auf eine Weiterverarbeitung, z.B. innerhalb eines Audioeditors, erscheint es ebenfalls realistisch, von zufällig gewählten Translationen zwischen den einzelnen Codecstufen auszugehen. Die  $n$ -te Generation wird somit als  $(DE)T_{\ell_{n-1}}(DE) \cdots T_{\ell_1}(DE)x$  erzeugt, wobei  $(\ell_i)_{i=1}^{n-1}$  mit  $0 \leq \ell_i < 1152$  eine Zufallsfolge ist.
- Wahl einer festen Translation  $0 \leq \ell < 1152$ ,  $\ell \neq 481$  und  $(T_\ell DE)^n$  ist die  $n$ -te Generation. Diese Art der Translation wurde einbezogen, um den Grad der Abhängigkeit der Codierung von der 32er Blocksynchronizität zu überprüfen.

Die Ergebnisse zeigen, daß sich die Degeneration durch keine der obigen Translationsarten grundsätzlich verlangsamen läßt. Lediglich bei den oben erwähnten, gegen Degeneration unempfindlichen Passagen ist bei der synchronisierenden Translation eine Verbesserung bemerkbar. Es ergeben sich folgende grundlegenden Ergebnisse:

- Die synchronisierende Translation (um 481 Samples) verursacht im Gegensatz zu den anderen Translationsarten einen starken Filtereffekt, der vor allem hohe Frequenzen betrifft und stark abdämpft. Die Qualität verschlechtert sich aber meist wesentlich deutlicher als bei den anderen Translationsarten. Allerdings sind die kratzenden oder rauschenden Artefakte hier nicht zu beobachten.
- Die Verwendung zufälliger Translationswerte ergibt ähnliche Ergebnisse wie das Standardverfahren ohne Translation.
- Die Wahl einer festen Translation um  $0 \leq \ell < 1152$  Samples,  $\ell \neq 481$ , liefert zwei wichtige Ergebnisse:

- Ist  $\ell = 481 + 32k$  für  $k \in \mathbb{Z} - \{0\}$ , so sind die Degenerationseffekte dieselben wie im Falle der synchronisierenden Translation.
- Schon Abweichungen der Translation um ein bis zwei Samples von den synchronisierenden 481 Samples bewirken einen Wechsel der Degeneration auf das oben beschriebene Kratzen bzw. Verzerren.

Insgesamt zeigt sich somit eine klare Abhängigkeit von der Blocksynchronizität abhängig von der Filterbankordnung von 32 Bändern.

Die Auswirkungen der unterschiedlichen Degenerationsarten sind in den Abb.3.7 bis 3.10 visualisiert. Dargestellt sind jeweils die Signalenergie bzw. die Bitallokierung pro Subband, gemittelt über ca. 200 bzw. ca. 300 Rahmen. In der dritten Dimension ist die Generation (1-20) von links hinten nach rechts vorne abgetragen. Die Subbänder sind zur besseren Visualisierung für aufsteigende Frequenzen von rechts hinten nach links vorne abgetragen. Die Darstellung der Subbandenergien orientiert sich an den Skalierungsfaktoren, ist also logarithmisch. Die Differenz der Bitallokierung in Abb. 3.10 ist in Bits angegeben. Das zu Abb. 3.7 gehörige Signal ist ein Ausschnitt des Popmusikstücks *104abba*, die übrigen Abbildungen beziehen sich auf einen Ausschnitt des Sprachstücks *087speech*. Das Popstück besteht im betrachteten Intervall aus einer gleichförmigen rhythmischen Passage mit Klavierbegleitung, das Sprachstück wird von einem einzigen Sprecher erzeugt. Es liegen somit im gesamten Intervall jeweils sehr ähnliche Klangkomponenten (gleiche percussive Elemente im Rhythmus, Formanten innerhalb der Sprache des Sprechers) vor, was die Mittelwertbildung über die Subbandenergien rechtfertigt.

Abb. 3.7 zeigt die Subbandenergien für den Fall der synchronisierenden Translation zwischen den Codecstufen. Der Abfall der Signalenergien in einigen Subbändern mit wachsender Generation ist deutlich zu erkennen. Für das Sprachstück bestätigt Abb. 3.8 diese Beobachtung. Zu erkennen ist deutlich die höhere Energie der Sprache in den unteren im Vergleich zu den oberen Subbändern. Abb. 3.9 zeigt die Parameter aus Abb. 3.8 für den Fall, daß keine Translation zwischen den Codecstufen vorgenommen werden. In diesem Fall ist das Verhalten der Energien stabiler. Abb. 3.10 zeigt den direkten Vergleich beider Translationsarten. Hier sind links jeweils die Differenzen der durchschnittlichen Signalpegel der einzelnen Generationen und der 1. Generation dargestellt. Die Grafiken der rechten Seite zeigen die entsprechenden Differenzen in der Bitallokierung. Oben wird jeweils die synchronisierende Translation und unten das Verhalten ohne Translation dargestellt.

### 3.2.3 Signal- und Parameterdegeneration

Auf Veränderungen von Generation zu Generation hin wurden insgesamt folgende Coder- und Signalparameter untersucht:

- Skalierungsfaktoren. Diese korrespondieren zum maximalen Signalwert des jeweiligen Subbands und somit grob zur Subbandenergie.
- Skalierungsfaktorauswahl. Es stellt sich heraus, daß aus den Kombinationen der Skalierungsfaktoren nicht so einfach Regelmäßigkeiten ablesbar sind.
- Bitallokierung. Die Anzahl allozierter Bits pro Subband läßt Rückschlüsse auf dessen perzeptuelle Wichtigkeit zu.

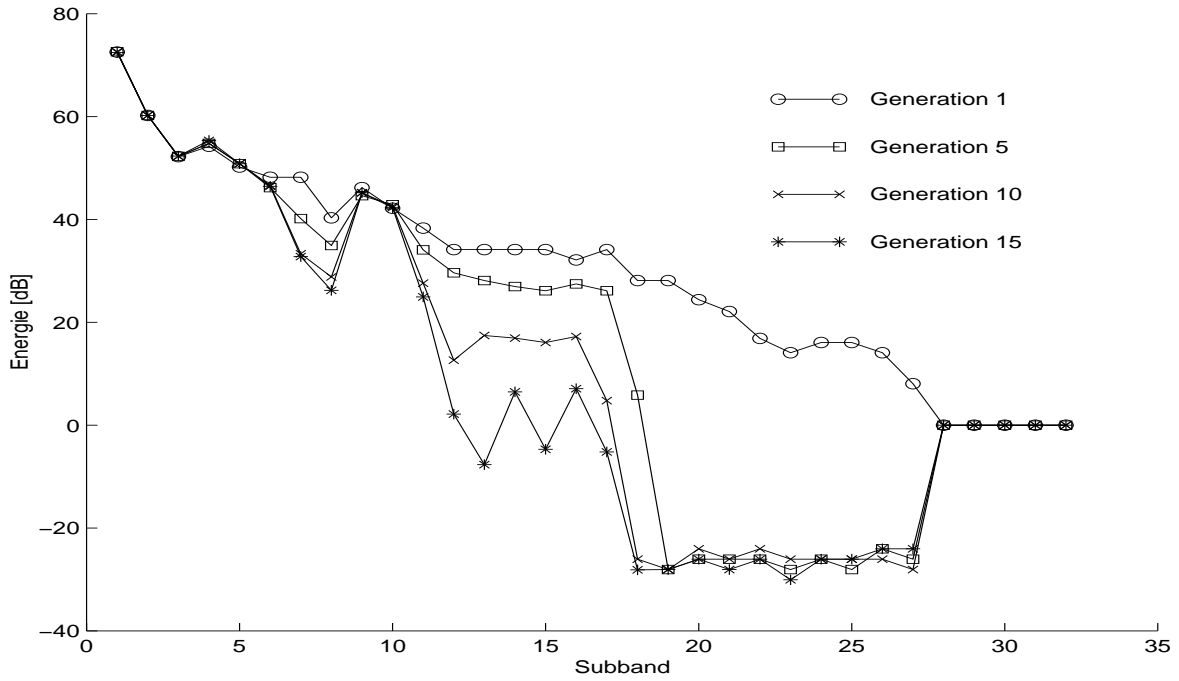


Abbildung 3.11: Degenerierung der Subbandenergien eines Rahmens des E-Gitarrenstücks für die Generationen 1., 5., 10. und 15.

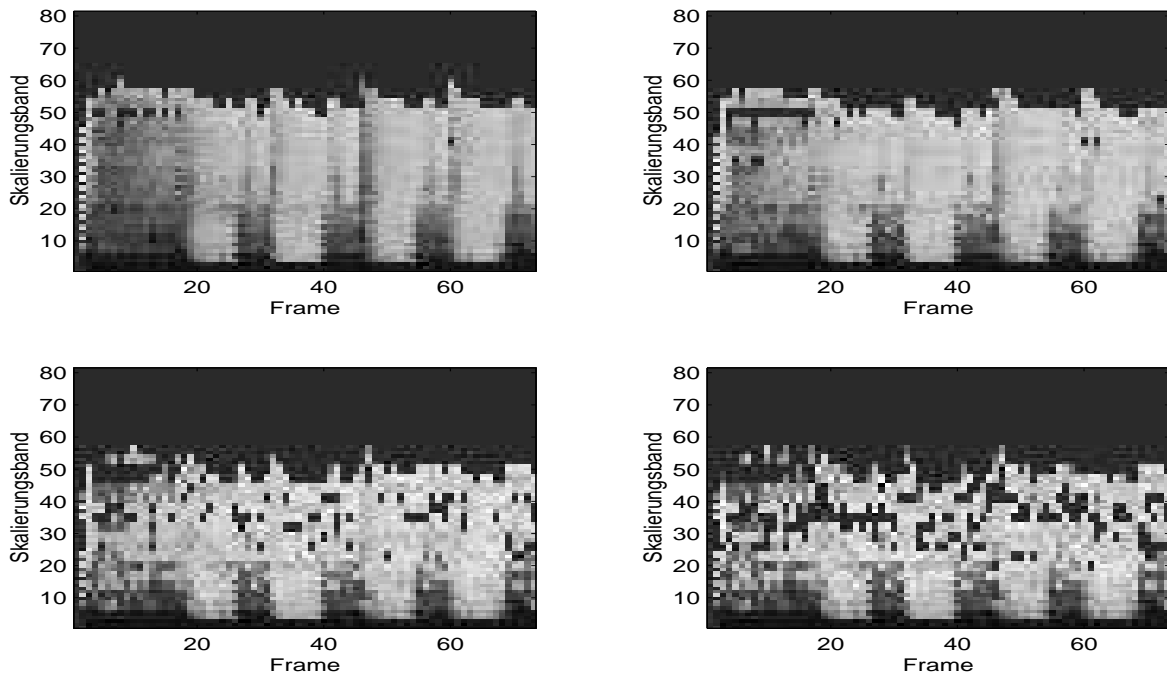


Abbildung 3.12: Degenerierung der Skalierungsfaktoren für das gesamte E-Gitarrenstück. Dargestellt sind wieder die Generationen 1. (links oben), 5. (rechts oben), 10. (links unten) und 15. (rechts unten).

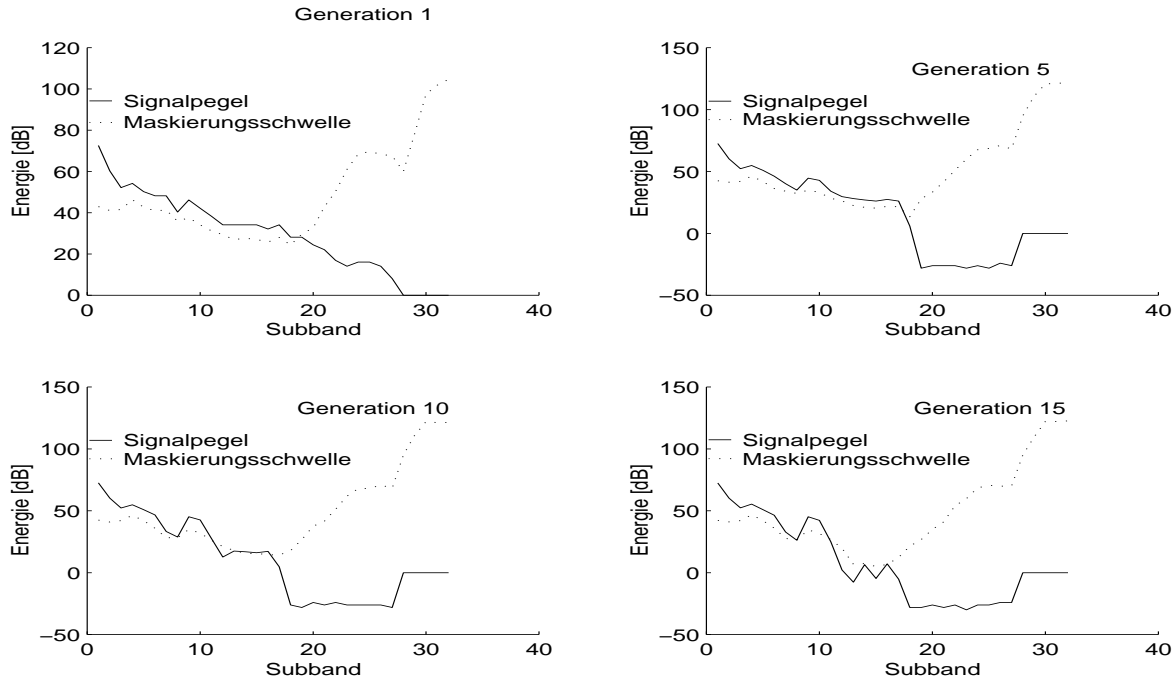


Abbildung 3.13: Vergleich von Signal- und Maskierungsschwelle des Datenrahmens aus den Abbildungen 3.11 und 3.12.

- Signalpegel pro Subband. Dies ist die Gesamtenergie pro Subband, wie sie auch vom psychoakustischen Modell verwendet wird.
- Maskierungsschwelle wie vom psychoakustischen Modell 1 der verwendeten Codecimplementation [40] berechnet.

Abb. 3.11 und 3.13 zeigen für den Fall synchronisierender Translationen die Entwicklung einzelner Parameter für die Generationen 1., 5., 10. und 15. eines Rahmens des E-Gitarrenstücks. Der starke Abfall der Energien zu den sehr hohen Subbändern hin ist dadurch zu erklären, daß der MPEG Codec sowieso nur abhängig von der Bitrate maximal bis zu einem bestimmten Subband Bits allokiert. Bei 128 kbps und 44.1 kHz ist dies Subband 27. Der Abfall der Energien der restlichen Subbänder (6-8 und 12-17) ist allerdings nicht auf diese Eigenschaft des Codecs zurückzuführen und bei den meisten der betrachteten Signalausschnitte in ähnlicher Form beobachtbar. Abb. 3.12 zeigt das Degenerationsverhalten auf den gesamten Signalausschnitt bezogen. Deutlich sind der Signalrhythmus (senkrechte periodische Linien) und die steigende Unregelmäßigkeit der Energieverteilung für wachsende Generationen zu bemerken.

### 3.2.4 Hörtests

Zur Verifikation der angegebenen Analyseergebnisse sollen Teile der Ergebnisse des in Kapitel 6 genauer beschriebenen informellen Hörtests herangezogen werden. Innerhalb des Hörtests wurden 26 Personen zu den zehn markierten Teststücken in Tabelle 6.6 befragt. Bezüglich der Signaldegeneration sind hier folgende Resultate von Bedeutung:

- Absolute Einschätzung der ersten und der fünften Generation bei Verwendung zufälliger

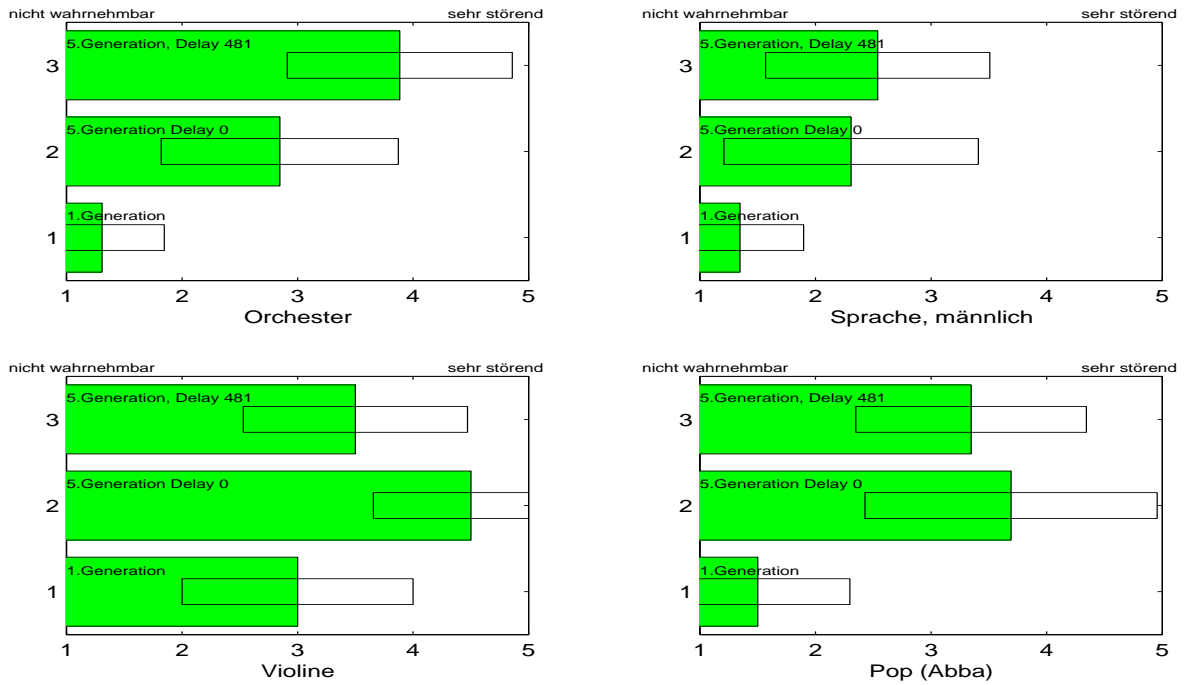


Abbildung 3.14: Bewertung der einzelnen Versionen der Teststücke in einer MOS-Beurteilung mit Standardabweichung.

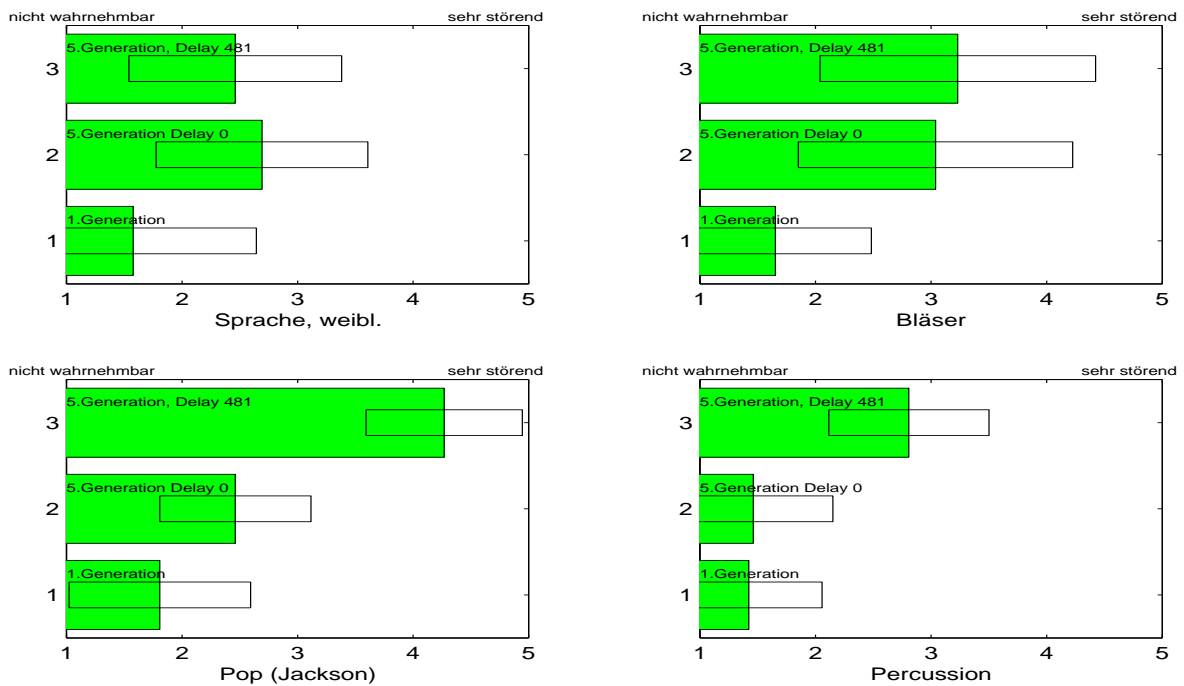


Abbildung 3.15: Bewertung der einzelnen Versionen der Teststücke in einer MOS-Beurteilung mit Standardabweichung.

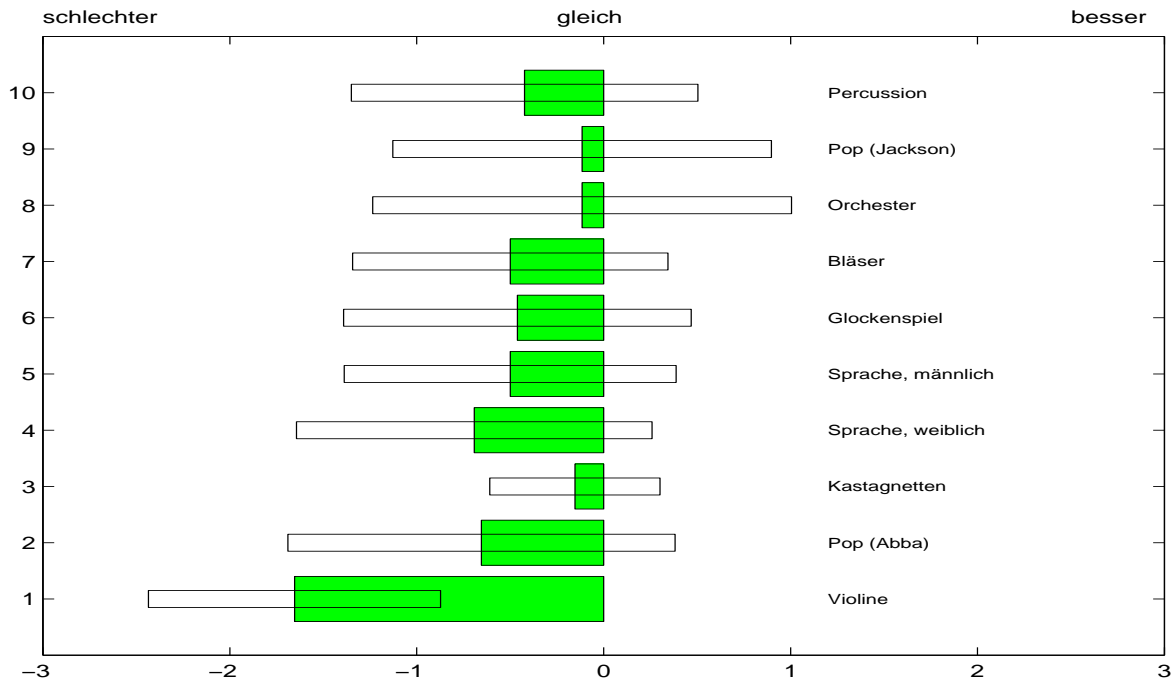


Abbildung 3.16: Bewertung des Vergleichs der dritten Generation mit der ersten Generation auf einer Skala von -3 bis 3 mit Standardabweichung.

Translationen zwischen den Codecstufen, sowie der fünften Generation bei Verwendung synchronisierender Translationen zwischen den Codecstufen.

- Vergleich der ersten mit der dritten Generation bei Verwendung zufälliger Translation zwischen den Codecstufen.

Abb. 3.14 und 3.15 zeigen die Bewertung der einzelnen Versionen der Teststücke in einer MOS (Mean Opinion Score) Bewertung. Die Skala reicht von 1-5 und ist als Klangbeeinträchtigungsskala (engl. *impairment scale*) gewählt. Dementsprechend bedeutet eine Bewertung mit "1", daß *keine* Beeinträchtigungen des Audiostücks wahrnehmbar sind, "2" *leicht wahrnehmbar*, "3" *wahrnehmbar*, "4" *störend* und "5" *sehr störend*.

Abb. 3.16 zeigt die Bewertung des Vergleichs der dritten Generation mit der ersten Generation auf einer Skala von -3 bis 3. Bei "-3" wurde die dritte Generation als *schlechter*, bei "0" als *gleichwertig* und bei "3" als *besser* als die erste Generation eingeschätzt.

Obwohl die Testpersonen nicht speziell geschult und auf den Test vorbereitet wurden, und aus Zeitgründen keine Einzeltests unter Verwendung von Kopfhörern durchgeführt werden konnten, zeigen die Resultate klare Tendenzen. Die absoluten Bewertungen, Abb. 3.14 und 3.15, zeigen, daß beim Signal erster Generation fast keine Beeinträchtigungen wahrgenommen wurden. (Das Violinstück bildet hier einen Ausnahme.) Hingegen werden die Signale 5. Generation in der Regel deutlich schlechter eingestuft. Hierbei schneidet meistens das Signal mit synchronisierender Translation zwischen den Codecstufen noch ein wenig schlechter ab als dasjenige ohne Translation. Abb. 3.16 zeigt, daß bei fast allen Stücken sogar der Unterschied zwischen nur zwei Generationen erkannt wird.

### 3.3 Lösungsansätze

Neben dem oben vorgestellten Verfahren zur Synchronisation der Codierung nachfolgender Generationen mit der früherer Generationen wurden für die aufgezeigten Fehlerquellen weitere Korrekturmechanismen vorgeschlagen bzw. in bestehende Codecs integriert.

Im Falle des Aliasings bietet sich z.B. eine Behandlung im Frequenzbereich an. Hier wäre ein Ansatz, das Spektrum des Rauschens, welches durch die Quantisierung erzeugt wird, direkt so zu wählen, daß die oben beschriebenen Aliasingeffekte benachbarter Bänder nicht auftreten. Dies wird bei MPEG-1 Layer III auf elegante Weise durch der Quantisierung vorgeschaltete Anti-Aliasing Butterflies geregelt. Der Übergang in den Frequenzbereich geschieht mittels diskreter Cosinustransformation.

Eine einfache, aber praktisch nur sehr aufwendig zu realisierende Methode zur Vermeidung von Arithmetikfehlern ist die Erhöhung der Rechengenauigkeit bzw. Bitbreite, die zur Zahldarstellung verwendet wird. Die Abhängigkeit zwischen Bitrate und Rechengenauigkeit beim AC-3 Coder wird in [37] für den Fall der Fixpunktdarstellung untersucht.

Die Behandlung der sehr dominanten Änderung des Signals durch die verlustbehaftete Quantisierung bleibt jedoch problematisch. Da sich hierdurch das Signalspektrum deutlich verändert und somit die Berechnung der psychoakustischen Parameter der folgenden Generationen stark beeinträchtigt wird, ist fraglich, ob die Degeneration des Signals überhaupt ohne Vorwissen über das Ursprungssignal verhindert werden kann. Es erscheint unrealistisch, etwa ein Signal zweiter Generation von einem Originalsignal automatisch „unterscheiden“ zu können. Mit den in den folgenden Kapiteln entwickelten Audiocodierern wird daher das Ziel verfolgt, den Quantisierungsfehler 1. sowie die Fehler 6., 7. und 8. nur *einmal* zu machen. Können gleichzeitig die Arithmetik- und Rekonstruktionsfehler kontrolliert werden, so kann die perzeptuelle Qualität der *ersten* Generation eines Signals in allen folgenden Generationen reproduziert werden.



## Kapitel 4

# Dateneinbettungsverfahren

In diesem Kapitel werden Techniken zur Einbettung von Zusatzinformationen innerhalb digitaler Datensätze entwickelt. Dazu werden zunächst einige größtenteils aus der Steganographie bekannte Techniken erläutert und auf ihre Verwendbarkeit zur Dateneinbettung unter den gegebenen Anforderungen untersucht. Hierbei stellt sich heraus, daß die vorgestellten Verfahren das ursprüngliche Signal auf meist nicht akzeptable Weise in seiner Qualität beeinträchtigen. Zur Behebung dieses Problems wird ein Verfahren entwickelt, das in Verbindung mit Transformationscodierungsverfahren und den bei diesen auftretenden Quantisierungsschritten eine Dateneinbettung ohne subjektive Qualitätsbeeinträchtigung zuläßt.

Zur Entwicklung des Verfahrens wird zunächst der  $\mathbb{R}^n$  als Wertebereich für Vektoren von Signalwerten betrachtet. Hierauf werden Fehler- und Einbettungsregionen eingeführt, mit deren Hilfe die Dateneinbettung durch eine Quantisierung mit datenangepaßter Rückquantisierung durchgeführt wird. Nach einigen Betrachtungen bezüglich der Dateneinbettungskapazität zu gegebenem Quantisierer wird das im Kontinuierlichen eingeführte Modell auf Vektoren digitalisierter Signalwerte übertragen. Zur Anwendung der Techniken in Audiokompressionsverfahren, die Skalierungsfaktoren zur Redundanzverringerung verwenden, wird die notwendige Anpassung der Einbettungskapazität in diesem Fall untersucht. Als alternatives Verfahren wird die direkte Biteinbettung beschrieben und bezüglich auftretender Rekonstruktionsfehler untersucht.

Abschließend wird auf zwei wichtige, in der Praxis auftretende Fragestellungen eingegangen. Einerseits geht es hierbei um die Anpassung der Einbettungsverfahren auf Transformations-Rücktransformationspaare, die Rekonstruktionsfehler verursachen oder arithmetikfehlerbehaftet sind. Andererseits werden die Biteinbettungsverfahren zur Verwendung bei variablen Einbettungsbitbreiten erweitert. Hierzu werden ein Konzept zur blockweisen Einbettung vorgestellt und mögliche Realisierungen vorgeschlagen.

### 4.1 Grundlagen und Steganographie

Psychoakustische Datenkompressionsverfahren für hohe Kompressionsraten reduzieren vorliegende Subbandsignale auf ihre perzeptuell „wesentlichen“ Bestandteile. Wie die hohen Kompressionsraten zeigen, lassen sich diese wesentlichen Bestandteile durch vergleichsweise wenige Daten beschreiben. Die Aufgabe des Decoders besteht in gewisser Weise darin, den vom Encoder erzeugten Code wieder „aufzublasen“. Da die Kompression bei den hier betrachteten Verfahren hochgradig verlustbehaftet ist, aus dem datenreduzierten Code also das Originalsi-

gnal im allgemeinen nicht mehr reproduzierbar ist, erzeugt der Decoder im allgemeinen Daten mit geringerem Informationsgehalt. Dies geschieht sowohl explizit durch die Wahl von groben Gittern für die Rückquantisierung der Subbandwerte, als auch implizit durch eine Technik die unter dem Namen *Noise Shaping* bekannt ist. Letztgenannte Technik bezeichnet das Vorgehen, das durch die Grobquantisierung induzierte Quantisierungsrauschen in eine bestimmte temporale oder spektrale Form zu bringen. Innerhalb der psychoakustischen Kompressionsverfahren wird diese Form gerade so gewählt, daß das Quantisierungsrauschen nach erfolgter Decodierung nicht wahrnehmbar ist. Redundanz ist hier implizit vorhanden, da die genaue Gestalt dieses Rauschens in gewissen Grenzen variieren darf.

Bei gebräuchlichen Audiokompressionsverfahren wird das Quantisierungsrauschen stets in eine bestimmte, von vorneherein festgelegte Form gebracht. Diese Form ist hierbei nur vom Audiosignal und den gewählten Kompressionsparametern, wie z.B. der Bitrate, abhängig. Die in diesem Kapitel beschriebenen Verfahren dienen dazu, den Freiraum für die Wahl eines Quantisierungsrauschens gezielt zur Einbettung von Zusatzinformationen auszunutzen. Da die Einbettung im Prinzip als Spezialfall des Noise Shaping durchgeführt wird, entstehen hierdurch keine hörbaren Beeinträchtigungen der Audioqualität.

Eine andere Sichtweise ist, daß die so eingebetteten Informationen nicht wahrnehmbar, oder durch bloßes Abhören des betreffenden Audiostücks nicht erkennbar sind. Im allgemeinen werden Verfahren, die das Ziel verfolgen, Zusatzinformationen unbemerkt innerhalb bestimmter Datensätze abzulegen, als *Steganographieverfahren* bezeichnet. Die verwendeten Datensätze oder Dateien können dabei die verschiedensten Formate besitzen, wie z.B. reine Text-, Bild- oder, wie in unserem Fall, Audiodateien. Die Motivation, Zusatzinformationen unbemerkt in bestehende Datensätze zu integrieren, ist ebenfalls vielgestaltig und reicht von einer einfachen „Aufmodulierung“ von Sekundärdaten (z.B. textuelle Informationen zu Bildern) über die unbemerkte Übermittlung geheimer Nachrichten bis hin zu digitalen Copyright- und Wasserzeichen (engl. *digital watermarking*). Innerhalb dieses Anwendungsspektrums gibt es eine Reihe von Anforderungen an den Einbettungsmechanismus:

- Kapazität einbettbarer Informationen,
- Robustheit des entstehenden Codes gegen
  - fehlerhafte Übertragung,
  - Attacken gegen die eingebetteten Informationen,
  - Beschädigung oder Entfernung durch Signalverarbeitungsverfahren,
- Erkennbarkeit und Decodierbarkeit durch Dritte. Hierbei kann es z.B. wünschenswert sein, daß eine auditive Erkennbarkeit ausgeschlossen, eine auf die rechnerintere Codierung bezogene Erkennbarkeit jedoch gefordert ist.

Je nach Anwendung kommt jedem Punkt dieses Anforderungskatalogs ein unterschiedliches Gewicht zu. In den in dieser Arbeit entwickelten Methoden wird es dabei weniger um den Schutz gegen Attacken oder die algorithmische Decodier- und Detektierbarkeit der eingebetteten Informationen gehen, sondern vielmehr um eine möglichst hohe Einbettungskapazität, auditiv perzeptuelle Transparenz und effektiven Schutz gegen Störungen wie sie durch einige Signalverarbeitungsverfahren verursacht werden. Insbesondere geht es hier *nicht* in erster Linie um die Entwicklung eines Verfahrens zur Realisierung digitaler Wasserzeichen. Die Hauptanforderung an solche Verfahren ist der Schutz des Wasserzeichens gegen dessen

Entfernung durch alle möglichen Arten von Signalverarbeitungsverfahren (insbesondere Kompression, DA/AD-Wandlung oder Filterung). Daher arbeiten solche Verfahren oft mit einer großen Redundanz, was wiederum unserer Anforderung an eine große Kapazität für einbettbare Informationen entgegensteht.

Da die Watermarkingverfahren jedoch große Beachtung und Anwendung finden, und somit für wichtige Ideen und Konzepte auf dem Gebiet der Steganographie sorgen, seien hier einige wichtige neuere Ansätze im Bereich audiovisueller Daten kurz beschrieben:

- Cox et al. [7] verwenden *Spread Spectrum Techniken* für die Einbettung von Wasserzeichen in audiovisuelle Daten. In ihrer Arbeit wird der wichtige Begriff der *perzeptuellen Kapazität* von Transformationskoeffizienten geprägt. Die Einbettung des Wasserzeichens erfolgt im Frequenzbereich.
- Ein Zeitbereichsverfahren, welches ebenfalls Frequenzmaskierung verwendet, wird von Boney et al. [3] vorgeschlagen. Mit Hilfe der durch ein psychoakustisches Modell bestimmten Maskierungsparameter wird ein Filter im Zeitbereich konstruiert, welches die Dateneinbettung vornimmt.
- Ein Verfahren zur Einbettung von Copyrightvermerken in Videodaten wird von Zhao und Koch [63] am Beispiel des JPEG-Bildkompressionsstandards entwickelt. Bei diesem Verfahren wird die Dateneinbettung nach dem verlustbehafteten Quantisierungsschritt innerhalb bestimmter Datenblöcke des codierten Bildes vorgenommen. Da dies im nicht mehr verlustbehafteten Teil des JPEG Verfahrens geschieht, kann die eingebettete Information später wieder decodiert werden. Die Positionen der Einbettung werden unter anderem so gewählt, daß durch die Einbettung keine zu großen Änderungen innerhalb des Bildes auftreten.
- Ebenfalls mit Spread Spectrum Techniken arbeiten Watermarkingverfahren nach Hartung und Girod [11], die insbesondere im Fall von MPEG-2 Videodatenströmen implementiert wurden. Hier geht es, unter leichter Modifikation der komprimierten Daten unter perzeptuellen Gesichtspunkten, um die Übertragbarkeit des Wasserzeichens sowohl im komprimierten Datensatz als auch im dekomprimierten Videomaterial.

Es soll kurz auf einige steganographische Grundtechniken eingegangen, und somit das weitere Vorgehen motiviert werden. Als Datensatz sei ein digitales Signal bestehend aus Datenworten  $b = \sum_{j=0}^{n-1} b_j 2^j =: (b_{n-1} \dots b_0)$  gegeben. Hierbei ist  $(b_{n-1} \dots b_0)$  die Binärdarstellung von  $b$ . Die rechtsstehenden Bits werden üblicherweise, entsprechend ihrer Signifikanz in der Binärdarstellung von  $b$ , als niederwertige, die linksstehenden als höherwertige Bits bezeichnet. Eine einfache (und gängige) Art der Einbettung von Zusatzinformationen in solch ein Datenwort  $b$  besteht aus der Ersetzung bestimmter niederwertiger Bits durch Bits der Zusatzinformation. Die  $k$ -Bit Einbettung ( $k \leq n$ ) der Zusatzinformation  $z = (z_{k-1} \dots z_0)$

$$E_1 : (b_{n-1} \dots b_k b_{k-1} \dots b_0), (z_{k-1} \dots z_0) \mapsto (b_{n-1} \dots b_k z_{k-1} \dots z_0) \quad (4.1)$$

zerstört die untersten  $k$  Bits des Datenwortes  $b$  und verursacht einen Fehler vom Maximalbetrag  $2^k - 1$ . Eine Einbettung dieser Art in alle Samples eines Signals führt somit zu einem bestimmten Rauschen, welches schon bei sehr niedrigem  $k$  deutlich bemerkbar ist (z.B. bei Wiedergabe als Audiosignal). Diese direkte Einbettung im Zeitbereich ist außerdem leicht erkennbar und decodierbar. Verfahren dieser Art werden trotz dieser erheblichen Nachteile an

vielen Stellen zur Verwendung mit verschiedensten Datensätzen, bei denen die Klassifizierung in niederwertige und höherwertige Bits sinnvoll ist, vorgeschlagen.

Eine Modifikation dieser Verfahren besteht darin, die direkte Biteinbettung in einem bestimmten Transformationsbereich durchzuführen. Dazu wird ein diskretes endliches Signal  $x \in \mathbb{R}^n$  mit einer linearen Abbildung  $T \in \text{GL}(n, \mathbb{C})$  via  $x \mapsto Tx =: y$  transformiert, die direkte Einbettung in  $y$  koeffizientenweise vorgenommen und der so gewonnene Koeffizientenvektor  $\tilde{y}$  via  $\tilde{y} \mapsto T^{-1}\tilde{y}$  zurücktransformiert. Oft ist die Forderung der Invertierbarkeit zu scharf und man fordert eine Invertierbarkeit mit kleinem Fehler, z.B.  $\tilde{T}Tx = x + e$  wo  $\tilde{T}$  die *Rekonstruktionsabbildung* und  $e$  der Rekonstruktionsfehler mit z.B.  $\|e\| \ll \|x\|$ , ist. Ist  $T$  eine Spektral- oder Subbandtransformation, kann die Einbettung frequenzspezifisch vorgenommen werden. Es zeigt sich jedoch, daß eine naive Verwendung solch eines Verfahrens zwar die Decodierbarkeit ohne Kenntnis von  $T$  deutlich erschwert, bezüglich der übrigen Negativeigenschaften der Direkteinbettung jedoch kaum Verbesserungen erbringt.

Eine wesentliche Verbesserung in Richtung perzeptueller Transparenz wird durch ein Verfahren von Wiese [62] erreicht. Ziel des vorgeschlagenen Verfahrens ist die Einbettung der Zusatzinformationen in den komprimierten Datenstrom im Falle von psychoakustisch basierter Transformations- oder Subbandcodierung. Die Idee des Verfahrens begründet sich darin, daß bei solchen Kompressionsverfahren, die zusätzlich mit konstanter Bitrate pro Datenrahmen arbeiten, nicht in allen Datenrahmen die laut psychoakustischem Modell maximal mögliche Datenreduktion vorgenommen wird. Stattdessen werden einige der Subbandwerte mit einer höheren Genauigkeit bzw. einer größeren Anzahl Bits als psychoakustisch benötigt, quantisiert. Dieses „Auffüllen“ ermöglicht somit die konstante Bitrate. Verfahren nach einem Patent von Wiese [62] setzen an dieser Stelle an und ersetzen diese zur erhöhten Genauigkeit verwendeten Bits durch Bits der einzubettenden Information. Zur Beschreibung einer durch dieses Vorgehen induzierten Einbettung wird die Notation der Binärdarstellung durch  $(b_{n-1} \dots b_0)_N := (b_{n-1} \dots b_0)2^N$  erweitert. Hiermit läßt sich über

$$E_2 : (b_{n-1} \dots b_0)_0, (z_{k-1} \dots z_0)_0 \mapsto (b_{n-1} \dots b_0 z_{k-1} \dots z_0)_{-k} \quad (4.2)$$

eine Einbettung durch Konkatenation definieren. Die angefügten Bits werden somit als niederwertige Bits interpretiert.

Der Vorteil einer solchen Einbettung ist die Wahl der Einbettungsstelle und Einbettungsbreite in Bits nach psychoakustischen Gesichtspunkten. Allerdings wird in [62] nicht die gesamte durch das psychoakustische Modell ermöglichte Einbettungskapazität ausgenutzt, da die Einbettung auf den Codewörtern vorgenommen wird. Soll die Einbettung nicht in den komprimierten Datenstrom, sondern in das unkomprimierte Audiosignal erfolgen, kann die gesamte bei der Kompression auftretende Quantisierungsvergrößerung ausgenutzt werden.

Ein wesentlicher Punkt bleibt jedoch bei den bisherigen Steganographieverfahren unberücksichtigt. Bei der Quantisierung wird zur Verringerung des maximalen Rekonstruktionsfehlers i.a. die Centroidbedingung eingehalten. Bei Skalarquantisierung mit konstanter Intervallbreite bedeutet dies, daß für alle  $x \in [A, B]$  die Quantisierung  $(q_n, \dots, q_0) := \tilde{Q}Qx := (A+B)/2$  die Abschätzung  $|x - \tilde{Q}Qx| \leq (B-A)/2$  für den maximalen Rekonstruktionsfehler liefert. Der nach psychoakustischen Gesichtspunkten zulässige Rekonstruktionsfehler wird somit über die Intervallbreite  $|B-A|$  gesteuert. Für eine Biteinbettung  $E_2$  des Datenworts  $0 < z = (z_{k-1}, \dots, z_0)$  gilt aber

$$\begin{aligned} |x - E_2(\tilde{Q}Qx, z)| &= |x - E_2((q_n, \dots, q_0), z)| \\ &= |x - ((q_n, \dots, q_0) + (z_{k-1}, \dots, z_0)2^{-k})| \end{aligned}$$

$$= \left| x - \left( \frac{A+B}{2} + z2^{-k} \right) \right|.$$

Der Rekonstruktionsfehler erhöht sich eventuell dadurch. Im Extremfall  $x = B$  erhält man den Fehler  $\frac{B-A}{2} + z2^{-k}$ , so daß *jedes* eingebettete  $z > 0$  eine Erhöhung des Fehlers bewirkt. Auch Einbettungsfunktionen wie  $E_1$  erhöhen i.a. einen gegebenen Rekonstruktionsfehler: Ist  $x \in \mathbb{N}$  eine Approximation von  $y \in \mathbb{N}$  mit Fehler  $|x - y| = e$ , so kann für eine einzubettende Information  $z \in \mathbb{N}$  ohne weiteres  $|E_1(x, z) - y| > e$  gelten, solange keine weitere Spezifikation von  $z$  und  $e$  gegeben ist. In der Audiokompression und den heutzutage gebräuchlichen hohen Kompressionsraten ist ein solch erhöhter Rekonstruktionsfehler normalerweise nicht akzeptabel. Dies liegt daran, daß bei den erzielten hohen Kompressionsraten der nach psychoakustischen Gesichtspunkten maximal zulässige Rekonstruktionsfehler meist vollständig zur Datenreduktion ausgeschöpft wird. Aus diesen Überlegungen folgt weiterhin, daß Verfahren nach [62] durch die Modifikation der Codeworte mittels Anfügen niederwertiger Bits ebenfalls den Rekonstruktionsfehler erhöhen können.

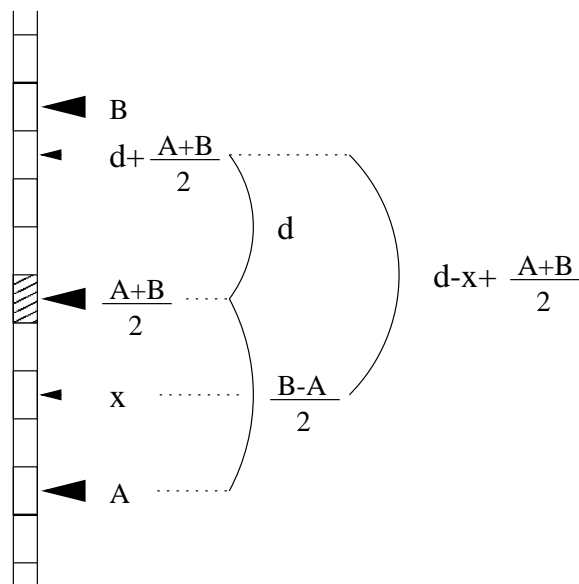


Abbildung 4.1: Einbettung  $E_2(x, d)$ . Es gilt  $|x - E_2(x, d)| = |x - ((A+B)/2 + d)| > (B-A)/2$  für die hier gewählten  $x, d$ .

Das in diesem Kapitel vorgestellte Verfahren basiert auf den folgenden Überlegungen. Abb. 4.1 zeigt eine Einbettung nach  $E_2$  wo  $|x - E_2(x, d)| > (B-A)/2$ , also größer als der maximale Rekonstruktionsfehler, ist. Das Problem liegt darin begründet, daß eine einfache Modifikation bestimmter Bits, wie sie durch die Einbettungen  $E_1$  und  $E_2$  vorgenommen wird, nicht exakt, sondern nur in der Größenordnung zum Rekonstruktionsfehler korrespondiert. Eine Methode, die dies umgeht, arbeitet nicht auf Bitebene sondern auf Abständen: Verwendet man im Beispiel aus Abb. 4.1 neben der Information, daß  $x \in [A, B]$  liegt und  $(A+B)/2$  die Centroide für die Rückquantisierung ist, noch die Zusatzinformation, daß  $x < (A+B)/2$  gilt, so kann man, ohne den Rekonstruktionsfehler zu erhöhen, anstatt der Centroide  $(A+B)/2$  alle beliebigen Werte im Intervall  $[A, (A+B)/2)$  zur Rückquantisierung verwenden. Anders ausgedrückt: Bettet man eine Zusatzinformation  $0 \leq d \leq (A+B)/2$  durch Subtraktion,  $E_3^-(A, B, d) := (A+B)/2 - d$  ein, so gilt für den maximalen Rekonstruktionsfehler

$|x - E_3^-(A, B, d)| < (B - A)/2$ . Für  $x \geq (A + B)/2$  verfährt man entsprechend durch Addition von  $d$  zur Centroide. Die Information, ob  $x < (A + B)/2$  gilt oder nicht, bzw. in welcher sogenannten *Fehlerregion* sich der Wert  $x$  befindet, ist durch nur ein Bit repräsentierbar.

Im folgenden wird das hier skizzierte Konzept formalisiert. Dabei stellt sich heraus, daß die resultierenden *Einbettungsregionen*, im obigen Fall  $[A, (A + B)/2)$  falls  $x < (A + B)/2$  bzw.  $[(A + B)/2, B]$  falls  $x \geq (A + B)/2$ , im Falle einer Vektorquantisierung auch von den Fehlerregionen verschieden sein können.

Insgesamt ergibt sich der in Abb. 4.2 dargestellte Einbettungsalgorithmus für psychoakustische Transformationscodierer. Das Verfahren kann den gesamten durch die Quantisierungsvergrößerung zur Verfügung gestellten Platz ausnutzen und vergrößert den Rekonstruktionsfehler, im Gegensatz zu  $E_1$  und  $E_2$ , in keinem Fall. Im nächsten Kapitel wird sich zeigen, daß den Einbettungsregionen noch weitere Bedeutung innerhalb der theoretischen Betrachtung der Generationseffekte zukommt.

## 4.2 Fehler- und Einbettungsregionen

Die Quantisierungsaufgabe in der Signalverarbeitung besteht einerseits aus einer Abbildung aus einem kontinuierlichen in einen diskreten, meist endlichen Wertebereich (wie z.B. bei einer A/D-Wandlung), andererseits geht es oft um eine Abbildung bereits quantisierter Daten von einem "feinen" auf ein "gröberes" Gitter. Zur Vereinfachung der Darstellung sollen für die nachfolgenden Betrachtungen reellwertige diskrete Signale herangezogen werden. Dabei wird allgemein auf die Vektorquantisierung von Elementen aus  $\mathbb{R}^n$  eingegangen.

Es sei  $(M, d)$  ein metrischer Raum. Ein *Quantisierer*  $\mathcal{P}$  ist eine höchstens abzählbar unendliche Teilmenge von  $M$ . Zu jedem  $P \in \mathcal{P}$  ist die zugehörige *Voronoi-Zelle* definiert durch

$$V(P) = V_{\mathcal{P}}(P) := \{x \in M \mid \forall P' \in \mathcal{P} : d(x, P) \leq d(x, P')\}.$$

Dann gilt  $M = \bigcup_{P \in \mathcal{P}} V(P)$ . Manchmal ist es günstiger,  $\mathcal{P}$  als Folge zu notieren, z.B.  $\mathcal{P} = (P_i)_{i=1}^n$  oder  $\mathcal{P} = (P_i)_{i \in \mathbb{Z}}$ . Zu gegebenem Quantisierer  $\mathcal{P}$  heißt  $(V_s(P))_{P \in \mathcal{P}}$  System *partitionierender Voronoi-Zellen* gdw.

- $\forall P \in \mathcal{P} : V_s(P) \subseteq V(P)$ ,
- $\overline{V_s(P)} = V(P)$  und
- $M = \bigsqcup_{P \in \mathcal{P}} V_s(P)$ .

In allen Beispielen dieses Abschnitts ist  $M = \mathbb{R}^n$  und  $\mathcal{P} = \{P_1, P_2, \dots\}$ . Für  $i \neq j$  ist  $V(P_i) \cap V(P_j)$  entweder leer oder enthalten in einer  $n-1$ -dimensionalen affinen Hyperfläche, die genau eine der Begrenzungsflächen von  $V(P_i)$  und  $V(P_j)$  bildet. Die *Quantisierung*  $Q$  bildet  $x \in \mathbb{R}^n$  auf eines bezüglich der Abstandsfunktion  $d$  nächsten  $P_i$  ab. Für eine abgeschlossene Teilmenge  $P \subset M$  bezeichnet  $\partial P$  den *Rand* und  $P^0 := P - \partial P$  das *Innere* von  $P$ . Falls  $\forall i : x \notin \partial V(P_i)$ , ist das zugehörige  $P_i$  eindeutig bestimmt, ansonsten wird zufällig eines der nächsten  $P_i$ 's bestimmt, wir schreiben  $q : x \mapsto V(P_i)$ . Wie üblich reicht es aus, nur den Index  $i$  bei der Codierung abzuspeichern. Die *Rückquantisierung* ist die Abbildung

$$\tilde{q} : i \mapsto P_i. \tag{4.4}$$

Als Abstandsfunktion soll der euklidische Abstand  $d(x, y) := \|x - y\|_2 = (\sum_{i=1}^n (x_i - y_i)^2)^{1/2}$  betrachtet werden, obwohl grundsätzlich auch andere Abstandsmaße möglich sind.

**Einbettungsalgorithmus****1. Input:**

Signalblock  $x \in \mathbb{R}^n$ , Coderparameter  $P$ ,

Psychoakustisches Modell  $\Psi : x \mapsto \Psi(x)$ ,

Bitallokierungsfunktion  $b : (\Psi(x), P) \mapsto (q, \dots) =: b(\Psi(x), P)$ , wo  $q$  die Quantisierungsart und -auflösung angibt. Das Bild von  $b$  kann weitere Elemente, z.B. Skalierungsfaktoren oder Auswahl eines Codebuchs, enthalten.

Decoderseitig liegt die einzubettende Information  $\mathcal{E}$  vor.

2. Berechne  $s := Tx$  und  $b(\Psi(x), P) = (q, \dots)$ .

3. Bestimme Fehlerregionen  $(s, q) \mapsto f(s, q)$ .

4. Quantisiere  $s \mapsto q(s)$ .

5. Übertrage  $q, q(s)$  und  $f(s, q)$  sowie gegebenenfalls weitere zur Decodierung benötigte Seiteninformationen aus  $b(\Psi(x), P)$ .

6. Rückquantisiere  $q(s) \mapsto \tilde{s}$ .

7. Bestimme aus den Fehlerregionen  $f(s, q)$  die Einbettungsregionen  $e(s, q)$ . Zu  $e(s, q)$  gehören Einbettungsvektoren  $E(s, q)$ .

8. Codiere die einzubettende Information  $\mathcal{E}$  bitblockweise durch Elemente von  $E(s, q)$ :

$$(e(s, q), \mathcal{E}) \mapsto (e_1, e_2, \dots) =: e \quad \text{mit } e_i \in E(s, q). \quad (4.3)$$

9. Bette die  $e_i$  ein:  $(\tilde{s}, e) \mapsto \tilde{s} + e$ , wobei die Addition punktweise auf den Vektoren in  $\mathbb{R}^n$  zu verstehen ist.

10. Rücktransformation via  $T^{-1}(\tilde{s} + e) =: \tilde{x}$ .

11. **Output:** Signalblock  $\tilde{x} \in \mathbb{R}^n$ .

Abbildung 4.2: Einbettungsalgorithmus für psychoakustische Transformations- oder Subbandcodierer mit Transformation  $T$  und Rücktransformation  $T^{-1}$  aus  $\text{GL}(n, \mathbb{R})$ . Es wird die Einbettung für einen Signalblock  $x \in \mathbb{R}^n$  beschrieben.

Der *Abstand* zweier beschränkter Teilmengen  $A, B$  von  $M$  sei definiert durch  $d(A, B) := \sup_{a \in A, b \in B} d(a, b)$ . Es gelten  $d(A, B) = d(B, A)$  sowie  $d(A, C) \leq d(A, B) + d(B, C)$ . Da jedoch der *Durchmesser*  $d(A) := d(A, A)$  typischerweise positiv ist, definiert  $d(\cdot, \cdot)$  keine Metrik. Der *Überdeckungsradius*  $R_P := d(V(P), \{P\})$  bzw.  $R_i := d(V(P_i), \{P_i\})$  gibt den maximalen Rekonstruktionsfehler der Zelle  $V(P)$  bzw.  $V(P_i)$  an.

**Definition 4.2.1** Für einen Quantisierer  $\mathcal{P} = (P_k)_{k \in \mathbb{Z}}$  heißen beschränkte abgeschlossene Teilmengen  $(\mathcal{F}_{i,k})_{0 \leq i \leq m(k), k \in \mathbb{Z}}$  mit  $\mathcal{F}_{j,k} \subset M$  und  $\forall k : m(k) > 0$  Fehlerregionenverbund, falls  $\forall k \in \mathbb{Z}$  und  $m := m(k)$ :

1.  $\mathcal{F}_{0,k} \cup \dots \cup \mathcal{F}_{m,k} = V(P_k)$ ,
2.  $\forall i \neq j : \mathcal{F}_{i,k}^0 \cap \mathcal{F}_{j,k}^0 = \emptyset$ ,
3.  $\forall i \geq 1 : d(\mathcal{F}_{i,k}) \leq R_k$  und
4.  $\mathcal{F}_{0,k} = V(P_k) - \bigcup_{i=1}^m \mathcal{F}_{i,k}^0$ .

Die einzelnen  $\mathcal{F}_{j,k}$  heißen dann Fehlerregionen.

Hierbei wurde O.B.d.A.  $\mathcal{F}_{0,k}$  als Komplement der restlichen Regionen in der Voronoi-Zelle gewählt. Durch  $\mathcal{F}_0 = \emptyset$ ,  $\mathcal{F}_1 = [A, (A+B)/2]$ ,  $\mathcal{F}_2 = [(A+B)/2, B]$  sind z.B. Fehlerregionen für das Beispiel des letzten Abschnitts gegeben.

**Definition 4.2.2** Sei  $\mathcal{P} = (P_k)_{k \in \mathbb{Z}}$  ein Quantisierer und  $(\mathcal{F}_{i,k})_{0 \leq i \leq m(k), k \in \mathbb{Z}}$  ein zugehöriger Fehlerregionenverbund. Die beschränkten abgeschlossenen Teilmengen  $\mathcal{E}_{i,k} \in M$  für  $k \in \mathbb{Z}$  und  $1 \leq i \leq m(k)$  heißen Einbettungsregionenverbund, falls

1. für  $1 \leq i \leq m(k) : d(\mathcal{F}_{i,k}, \mathcal{E}_{i,k}) \leq R_k$ , d.h.  $\mathcal{E}_{i,k}$  ist Einbettungsregion zu  $\mathcal{F}_{i,k}$ , und
2. für  $i \neq j$ , sowie  $k, \ell$  beliebig,  $\mathcal{E}_{i,k}^0 \cap \mathcal{E}_{j,\ell}^0 = \emptyset$ .

Im Beispiel des letzten Abschnitts waren  $\mathcal{E}_1 = \mathcal{F}_1$  und  $\mathcal{E}_2 = \mathcal{F}_2$  gewählt.

Es ist wichtig zu bemerken, daß aufgrund der ‘‘Komplementregionen’’  $\mathcal{F}_{0,k}$  folgende Identitäten im allgemeinen *nicht* gelten müssen:

- $\bigcup_{i \geq 1} \mathcal{E}_{i,k} = V(P_k)$  oder  $\bigcup_{i \geq 1} \mathcal{E}_{i,k} \subset V(P_k)$ ,
- $\bigcup_k \bigcup_{i \geq 1} \mathcal{E}_{i,k} = \mathbb{R}^n$ ,
- $d(\mathcal{E}_{i,k}) \leq R_k$ .

**Beispiel 4.2.3** Wir betrachten  $M = \mathbb{R}^2$  mit dem Quantisierungsgitter  $2\mathbb{Z}^2$ :

$$\mathcal{P} := \{(x, y) | x, y \in 2\mathbb{Z}\}.$$

Es gilt

$$V((x, y)) = [x - 1, x + 1] \times [y - 1, y + 1]. \quad (4.5)$$

$P_0 := (0, 0)$ ,  $V(P_0) = [-1, 1] \times [-1, 1]$ ,  $R_0 = \sqrt{2}$ .

1. Der Fehlerregionenverbund sei gegeben durch

$$\begin{aligned} \mathcal{F}_{1,0} &:= [0, 1] \times [0, 1], \\ \mathcal{F}_{2,0} &:= [-1, 0] \times [0, 1], \\ \mathcal{F}_{3,0} &:= [0, 1] \times [-1, 0], \\ \mathcal{F}_{4,0} &:= [-1, 0] \times [-1, 0]. \end{aligned}$$

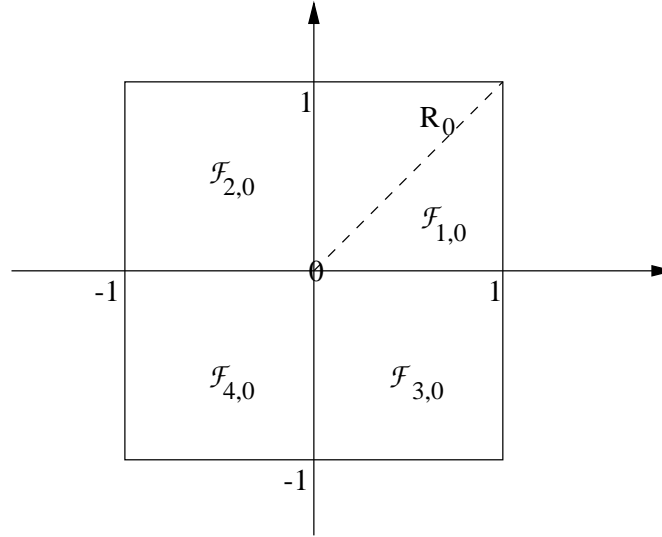


Abbildung 4.3: Quadratische Fehlerregionen.

Die Fehlerregionen sind in Abb.4.3 dargestellt. Der Einbettungsregionenverbund sei durch  $\mathcal{E}_{i,0} := \mathcal{F}_{i,0}$  für  $i = 1, \dots, 4$  gegeben.

2. Der Fehlerregionenverbund sei gegeben durch

$$\begin{aligned} \mathcal{F}_{1,0} &:= [0, 1] \times [0, 1], \\ \mathcal{F}_{2,0} &:= [-1, 0] \times [-1, 0]. \end{aligned}$$

sowie  $\mathcal{F}_{0,0} := V(P_0) - \mathcal{F}_{1,0}^0 - \mathcal{F}_{2,0}^0$ .

Für  $p_1 := \{(0, 0), (0, 1), (1, 0), (1, 1)\}$  und  $p_2 := \{(0, 0), (-1, 0), (0, -1), (-1, -1)\}$  definieren wir jeweils aus dem Schnitt vierer Kreise,  $B_r(x) := \{y \in \mathbb{R}^n | d(x, y) \leq r\}$ , die Einbettungsregionen  $\mathcal{E}_{1,0} := \bigcap_{x \in p_1} B_{\sqrt{2}}(x)$  und  $\mathcal{E}_{2,0} := \bigcap_{x \in p_2} B_{\sqrt{2}}(x)$ . Abb. 4.4 zeigt links die Fehlerregionen und rechts die Einbettungsregionen.

3. Aufgrund seiner Flächenmaximierungseigenschaft betrachten wir noch den Kreis als Fehler- und Einbettungsregion:  $\mathcal{E}_{1,0} := \mathcal{F}_{1,0} := B_{\sqrt{2}/2}(P_0)$ , siehe Abb. 4.6. Das Komplement des Kreises in  $V(P_0) - \partial V(P_0)$  bildet  $\mathcal{F}_{0,0}$ .

In allen drei Beispielen erhält man die Fehler- und Einbettungsregionen  $\mathcal{F}_{i,j}$  und  $\mathcal{E}_{i,j}$  für  $j \neq 0$  durch Verschieben der  $\mathcal{F}_{i,0}$  und  $\mathcal{E}_{i,0}$  um den Vektor  $P_j - P_0$ . So zeigt Abb. 4.5 die Lage der auf das gesamte  $2\mathbb{Z}^2$ -Gitter fortgesetzten Einbettungsregionen.

Sollen allgemein für einen Gitterquantisierer  $\mathcal{P} := \text{span}_{\mathbb{Z}}\{v_1, \dots, v_n\}$  Fehler- oder Einbettungsregionen bestimmter Voronoi-Zellen zu Verbänden fortgesetzt werden, kann dies gegebenenfalls durch Periodisierung bezüglich des Gitters geschehen. Sind  $(\mathcal{F}_{i,k})_{0 \leq i \leq m(k)}$  die Fehlerregionen einer Voronoi-Zelle  $V(P_k)$  des Gitterquantisierers  $\mathcal{P}$ , so ist die *Periodisierung*

$$\{\mathcal{F}_{i,k} + v | v \in \text{span}_{\mathbb{Z}}\{v_1, \dots, v_n\}, 0 \leq i \leq m(k)\}$$

ein Fehlerregionenverbund zu  $\mathcal{P}$ . Für Einbettungsregionen gilt dies aber im allgemeinen nicht, da diese sogar über die Voronoi-Zellen hinausragen können. In unseren vorherigen Beispielen waren jedoch sämtliche Fehler- und Einbettungsregionen einer Voronoi-Zelle periodisierbar.

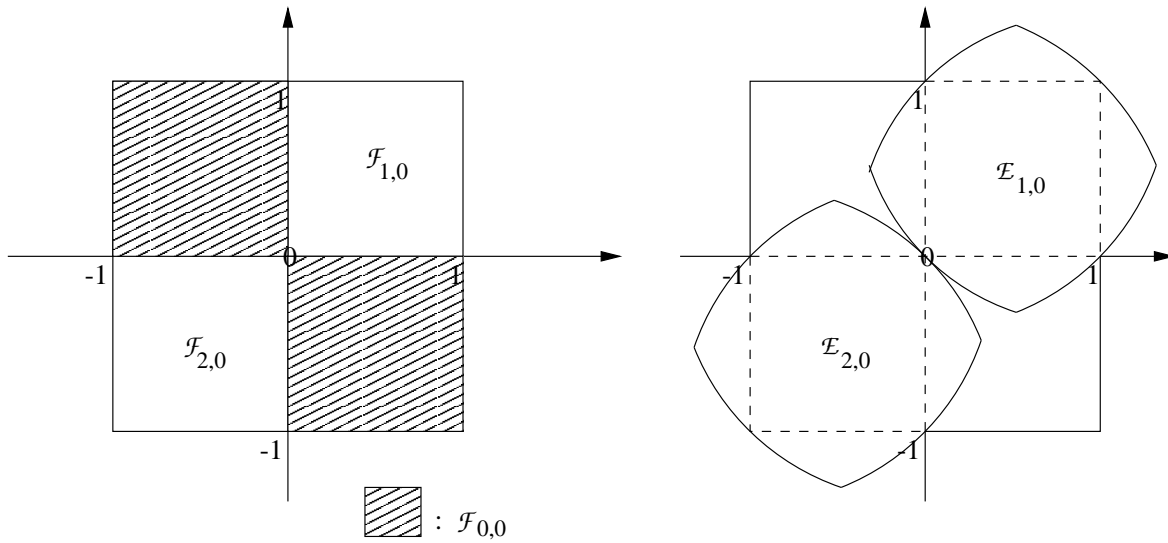


Abbildung 4.4: Fehlerregionen im ersten und vierten Quadranten (links) sowie Vier-Kreisschnitt Einbettungsregionen (rechts).

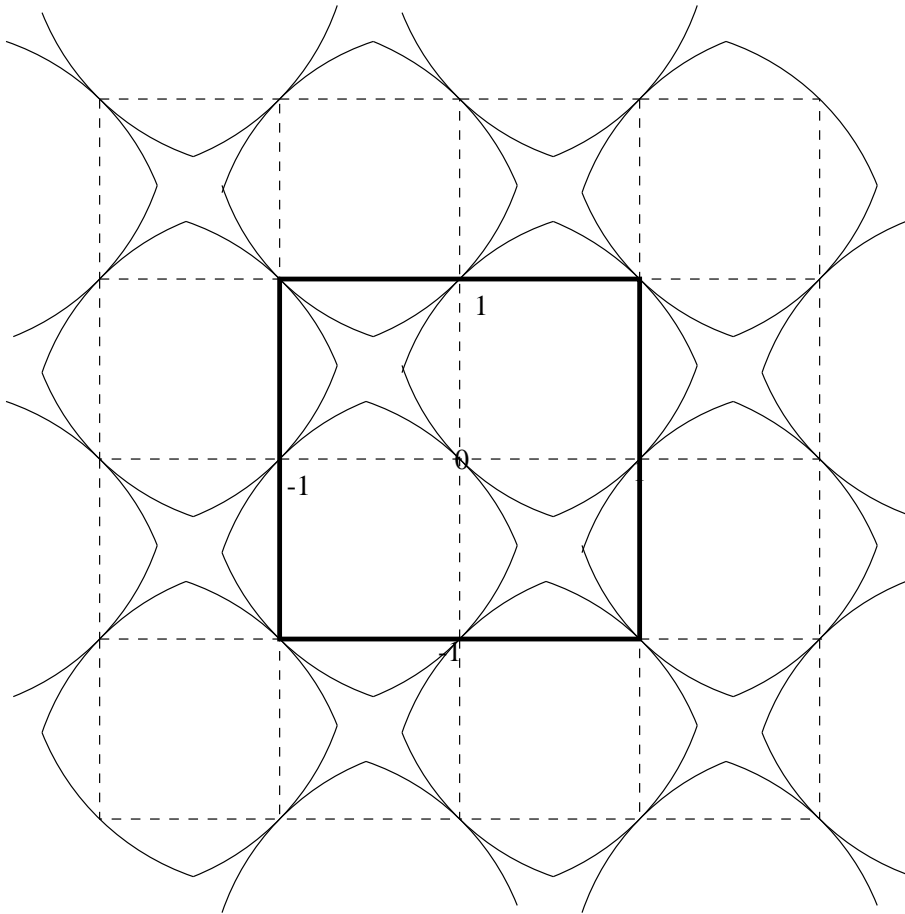


Abbildung 4.5: Periodisierung der Vier-Kreisschnitt Einbettungsregionen der hervorgehobenen Voronoi-Zelle  $V(P_0)$ .

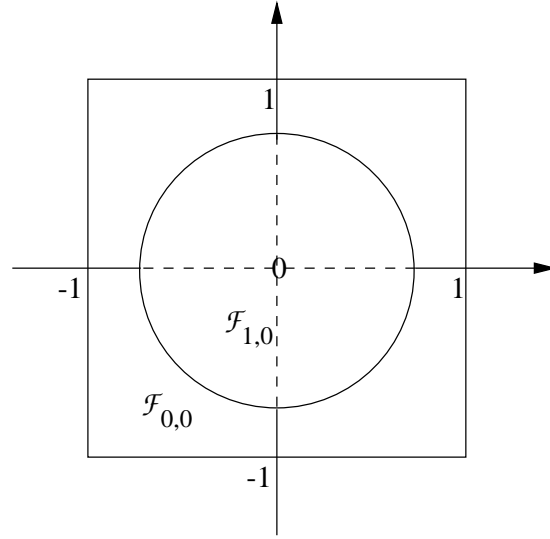


Abbildung 4.6: Fehlerregionen Kreis und Kreiskomplement.

**Beispiel 4.2.4** Zu jedem Fehlerregionenverbund  $(\mathcal{F}_{i,k})_{i,k}$  und Mengen  $\mathcal{E}_{i,k} \subset \mathcal{F}_{i,k}$  ist  $(\mathcal{E}_{i,k})_{i,k}$  Einbettungsregionenverbund, da aus  $d(A) < r$  und  $B \subseteq A$  sofort  $d(A, B) < r$  folgt.

Es ist natürlich weder die Verwendung eines Gitterquantisierers noch die Verwendung kongruenter Fehler- und Einbettungsregionenverbünde für kongruente Voronoi-Zellen generell vorausgesetzt.

### 4.3 Datenangepasste Rückquantisierung

Die einzubettende Zusatzinformation wird durch einen Vektor  $e \in \mathbb{R}^n$  repräsentiert. Zur Einbettung wird vom Rückquantisierer neben dem Codewortindex noch zusätzlich die Information über die Fehlerregion benötigt. Daher wird die Quantisierung  $Q : \mathbb{R}^n \rightarrow \mathbb{Z}^2$  hier definiert durch

$$Q : x \mapsto (i, k), \quad \text{falls } x \in \mathcal{F}_{i,k}. \quad (4.6)$$

Die Einbettung wird mit Hilfe der Menge  $E_{i,k} := \{y - P_k \mid y \in \mathcal{E}_{i,k}^0\}$  der Einbettungsvektoren durch eine modifizierte Rückquantisierung vollzogen. Die Rückquantisierung wird durch

$$\begin{aligned} \tilde{Q} : \text{Im}(Q) \times \bigcup_{j,\ell} E_{j,\ell} &\longrightarrow \bigcup_{i,k} \mathcal{E}_{i,k} \\ (i, k, e) &\mapsto \begin{cases} P_k + e & \text{falls } e \in E_{i,k}, \\ P_k & \text{sonst,} \end{cases} \end{aligned}$$

definiert. Die Regionen  $\mathcal{F}_{0,k}$  haben eine Sonderfunktion. Im Fall  $x \in \mathcal{F}_{0,k}$  wird keine Einbettung vorgenommen. Die Motivation hierzu ist die Vermeidung einer zu feinen Stückelung der Fehlerregionen zu ermöglichen, bzw. die Reduzierung des erwarteten Bitaufwands.

**Satz 4.3.1** Mit diesen Konventionen gilt  $\forall x \in \mathcal{F}_{i,k}$  und  $\forall e \in E_{i,k}$ :  $d(\tilde{Q}(Qx, e), x) \leq R_k$ .

*Beweis.* Für  $Qx = (i, k)$  gilt

$$\begin{aligned}
d(\tilde{Q}(Qx, e), x) &= d(P_k + e, x) && \text{da } e \in E_{i,k}, \\
&= d(P_k + y - P_k, x) && \text{für } y \in \mathcal{E}_{i,k}, \\
&= d(y, x) && \text{mit } x \in \mathcal{F}_{i,k}, \\
&\leq R_k && \text{nach Voraussetzung.}
\end{aligned}$$

□

**Bemerkung 4.3.2** In allen obigen Beispielen gilt  $\forall k$  und  $i \geq 1 : d(\mathcal{F}_{i,k}, \mathcal{E}_{i,k}) = \sqrt{2}$ .

Da für  $i \neq j$  und beliebige  $k, \ell$  nach Definition  $\mathcal{E}_{i,k}^0 \cap \mathcal{E}_{j,\ell}^0 = \emptyset$  gilt, sind für  $x \in \mathcal{F}_{i,k}$  und  $0 \neq e \in E_{i,k}$  aus  $\tilde{x} := \tilde{Q}(Qx, e)$  eindeutig  $i$  und  $k$  ablesbar. Somit kann ein Decoder hieraus  $P_k$  und durch  $e = y - P_k$  den eingebetteten Vektor  $e$  berechnen. Laut Satz 4.3.1 wird der Vektor  $e$  unter Einhaltung des maximalen Rekonstruktionsfehlers  $R_k$  eingebettet bzw. übertragen.

Natürlich wurde bisher nur idealisiert der  $\mathbb{R}^n$  als Wertebereich für die Ausgangswerte  $x$  und die eingebetteten  $\tilde{x}$  betrachtet. Es ist wichtig, die beabsichtigte Anwendung auf endliche Wertebereiche im Auge zu behalten, da sich über kontinuierlichen Wertebereichen wie  $\mathbb{R}^n$  Begriffe wie die Kapazität an übertragbarer Information, die durch eine Einbettungsregion gewonnen wird, nicht in analoger Weise verhalten. So ist die Binärdarstellung eines  $e \in \mathbb{R}$  i.a. durch unendlich viele Bits gegeben. Liegt also z.B. für  $A \neq B$  das Intervall  $[A, B]$  innerhalb einer Einbettungsregion, so sind theoretisch durch geschickte Auswahl eines  $e \in [A, B]$  beliebig viele Bits einbettbar, was natürlich nicht praktikabel ist. Da aber die in der Praxis relevanten Wertebereiche meist durch relativ feine Gitter in  $\mathbb{R}^n$  gegeben sind, korrespondieren zumindest die Größenverhältnisse über  $\mathbb{R}^n$  zu solchen über diskreten oder endlichen in  $\mathbb{R}^n$  liegenden Gittern. Ein Beispiel ist das Verhältnis des Volumens der Einbettungsregionen einer Voronoi-Zelle und der Voronoi-Zelle selbst.

Die Effizienz des Einbettungsmechanismus wird unter den Gesichtspunkten

- benötigte Anzahl Fehlerregionen,
- Größenverhältnis von Einbettungsregionen und Voronoi-Zelle,
- Einbettungskapazität pro aufgewendetem Bit

untersucht.

Zur Einbettung der Zusatzinformationen benötigt der Dequantisierer  $\tilde{Q}$  außer dem Quantisierungscodewort die Fehlerregion des ursprünglichen Datenworts. Bei Anwendung des Einbettungsverfahrens innerhalb des Decoderteils eines Audiokompressionsverfahrens ist diese Information jedoch normalerweise nicht verfügbar, muß also zusätzlich vom Encoder zum Decoder übertragen werden. Also ist eine platzsparende Codierung hierfür zu verwenden. Des weiteren soll von einer bestimmten Wahrscheinlichkeitsdichte  $p_{V(P_k)}$  der Datenworte innerhalb einer Voronoi-Zelle  $V(P_k)$  ausgegangen werden. Eingeschränkt auf die Fehlerregionen ergibt dies Gewichte  $P_{\mathcal{F}_{i,k}} := \int_{\mathcal{F}_{i,k}} p_{V(P_k)}(x) dx$  mit  $\sum_{i=0}^m P_{\mathcal{F}_{i,k}} = 1$ . In den Beispielen soll von der Gleichverteilung ausgegangen werden. Da hier nur der Quantisierungsfehler beschrieben wird, ist dies für die meisten Anwendungen auch ausreichend.

Für die Fehlerregionen  $\mathcal{F}_{0,k}, \dots, \mathcal{F}_{m,k}$  einer Voronoi-Zelle sei  $c : \{0, \dots, m\} \rightarrow \{0, 1\}^*$  eine Binärcodierung der Regionsnummern,  $|c(i)|$  die Länge der Binärdarstellung von  $i$ .

- Der *erwartete Bitaufwand* für die Übertragung der Regionsnummer pro Datenwort  $x \in V(P_k)$  ist definiert durch

$$A := A(\mathcal{F}_{\cdot,k}) := \sum_{i=0}^m |c(i)| P_{\mathcal{F}_{i,k}}. \quad (4.7)$$

- Die *Erfolgswahrscheinlichkeit* für die Einbettung ist definiert durch

$$E := E(\mathcal{F}_{\cdot,k}) := \sum_{i=1}^m P_{\mathcal{F}_{i,k}}. \quad (4.8)$$

- Entsprechend ist der *erwartete Bitaufwand pro Einbettung* definiert durch

$$B := B(\mathcal{F}_{\cdot,k}) := A/E. \quad (4.9)$$

- Die *erwartete Kapazität pro Einbettung* ist definiert durch

$$K := K(\mathcal{F}_{\cdot,k}, \mathcal{E}_{\cdot,k}) := \frac{\sum_{i=1}^m P_{\mathcal{F}_{i,k}} \text{Vol}(\mathcal{E}_{i,k})}{E}. \quad (4.10)$$

- Aus der Kapazität pro erfolgreicher Einbettung ergibt sich die *Kapazität pro Bit* als

$$\frac{K}{B} = \frac{\sum_{i=1}^m P_{\mathcal{F}_{i,k}} \text{Vol}(\mathcal{E}_{i,k})}{\sum_{i=0}^m |c(i)| P_{\mathcal{F}_{i,k}}}. \quad (4.11)$$

Manchmal wird solch ein Volumen noch durch eine Zusatznormierung “dimensionslos” gemacht. Hier soll jedoch darauf verzichtet werden.

Wir betrachten die für unsere bisherigen Beispiele hieraus unter Annahme einer Gleichverteilung der W-Dichten resultierenden Daten.

**Beispiel 4.3.3** 1.  $c : 1 \mapsto 00, 2 \mapsto 01, 3 \mapsto 10, 4 \mapsto 11$  ist eine bezüglich der Entropie optimale Codierung der Fehlerregionen. Wir erhalten  $A = 2, E = 1, B = 2, \text{Vol}(\mathcal{E}_{i,0}) = 1$ , also  $K = 1$  und  $\boxed{K/B = 0,5}$ .

2.  $c : 0 \mapsto 0, 1 \mapsto 10, 2 \mapsto 11$  ist eine bezüglich der Entropie optimale Codierung der Fehlerregionen. Es gilt  $A = P_{\mathcal{F}_{0,0}} + 2P_{\mathcal{F}_{1,0}} + 2P_{\mathcal{F}_{2,0}} = 1,5$  sowie  $E = 0,5$  und damit  $B = 3$ . Eine Nebenrechnung zeigt  $\text{Vol}(\mathcal{E}_{1,0}) = \text{Vol}(\mathcal{E}_{2,0}) = 1,746\dots$  (Hierzu bemerkt man, daß

$$(\text{Vol}(\mathcal{E}_{1,0}) - \text{Vol}(\mathcal{F}_{1,0}))/2 = \text{Vol}(B_{\sqrt{2}}((0,0)) \cap B_{\sqrt{2}}((1,0))) - \text{Vol}(B_{\sqrt{2}})/2.)$$

Daher gelten  $K = 1,746\dots$  und  $\boxed{K/B = 0,582\dots}$ , d.h., es kann mehr Fläche pro aufgewendetem Bit zur Einbettung verwendet werden als im ersten Beispiel. Allerdings kann eine Einbettung in diesem Beispiel nicht so häufig vorgenommen werden, womit ein Trade-Off zwischen Kapazität pro Bit und Datenrate gegeben ist.

3.  $c : 0 \mapsto 0, 1 \mapsto 1$  ist die kanonische Codierung der Regionen. Man erhält  $A = 1, E = \text{Vol}(\mathcal{F}_{1,0})/\text{Vol}(V(P_0)) = 0,39\dots, B = 2,546\dots, K = 1,57\dots$  sowie  $\boxed{K/B = 0,617\dots}$ . Die Fläche pro aufgewendetem Bit ist somit in diesem Beispiel noch höher, die Einbettungswahrscheinlichkeit noch niedriger als zuvor.

Ein Kriterium für eine große Kapazität pro Bit ist offensichtlich eine möglichst großflächige Abdeckung der Voronoi-Zelle mit möglichst wenigen Fehlerregionen. Wir wollen kurz auf die

Frage der minimal notwendigen Anzahl von Fehlerregionen zur *vollständigen* Überdeckung der Voronoi-Zelle eingehen. Formal ist dies die Frage nach

$$m(V(P), P) := \min \left\{ |F| \mid V(P)^o \subseteq \bigsqcup_{Q \in F} Q \subseteq V(P), \forall Q \in F : d(Q) < R_P \right\}, \quad (4.12)$$

wo wie üblich  $F \ni Q \subset \mathbb{R}^n$ .

**Lemma 4.3.4**  $m([-1, 1] \times [-1, 1], (0, 0)) = 4$ .

*Beweis.* Beispiel 4.2.3, Teil 1., zeigt  $m([-1, 1] \times [-1, 1], (0, 0)) \leq 4$ . Da für alle

$$x \neq y \in \{(-1, 1), (-1, -1), (1, 1), (1, -1)\}$$

$|x - y| \geq 2 > \sqrt{2}$  gilt, gehören alle diese vier Punkte zu verschiedenen Fehlerregionen.  $\square$

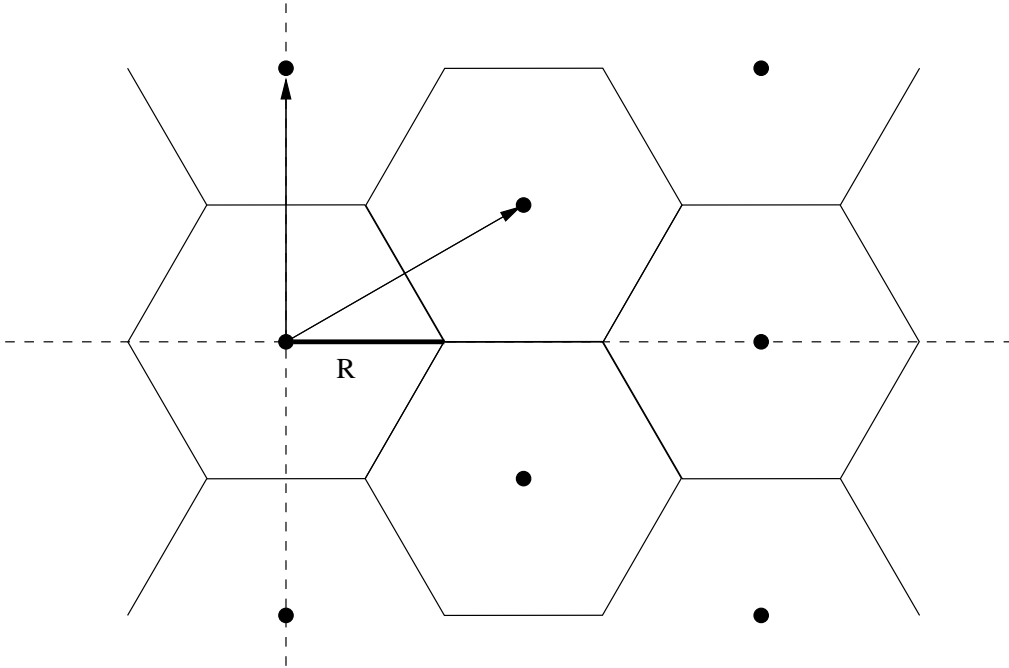


Abbildung 4.7: Hexagonalgitter mit Voronoi-Zellen und Basisvektoren.

Als weiteres Beispiel für einen Gitterquantisierer soll das Hexagonalgitter untersucht werden. Dieses ist bezüglich der Platzausnutzung bei gleichem maximalen Rekonstruktionsfehler effizienter als das  $\mathbb{Z}^2$ -Gitter. Tatsächlich liefert das Hexagonalgitter sogar einen auf  $\mathbb{R}^2$  optimalen Quantisierer. Bei gegebenem Überdeckungsradius  $R$  ist der Quantisierer durch  $\text{span}_{\mathbb{Z}}\{(0, \sqrt{3}R), (3R/2, \sqrt{3}R/2)\}$  gegeben. Abb. 4.7 stellt einen Ausschnitt des Gitters, die hexagonalen Voronoi-Zellen, sowie zwei Vektoren, die eine Basis des Gitters bilden, dar. Der Innenradius der Hexagone bzw. die Höhe eines der einbeschriebenen sechs gleichseitigen Dreiecke ist  $h = \sqrt{3}R/2$ , die Fläche der Dreiecke  $\sqrt{3}R^2/4$  und die Fläche der Hexagone  $H$  somit  $\text{Vol}(H) = 3\sqrt{3}R^2/2$ .

**Beispiel 4.3.5** Zum Vergleich mit dem  $2\mathbb{Z}^2$ -Gitter betrachten wir  $R = \sqrt{2}$  für zwei Einbettungsmethoden. Es gilt  $\text{Vol}(H) = 5,196\dots$

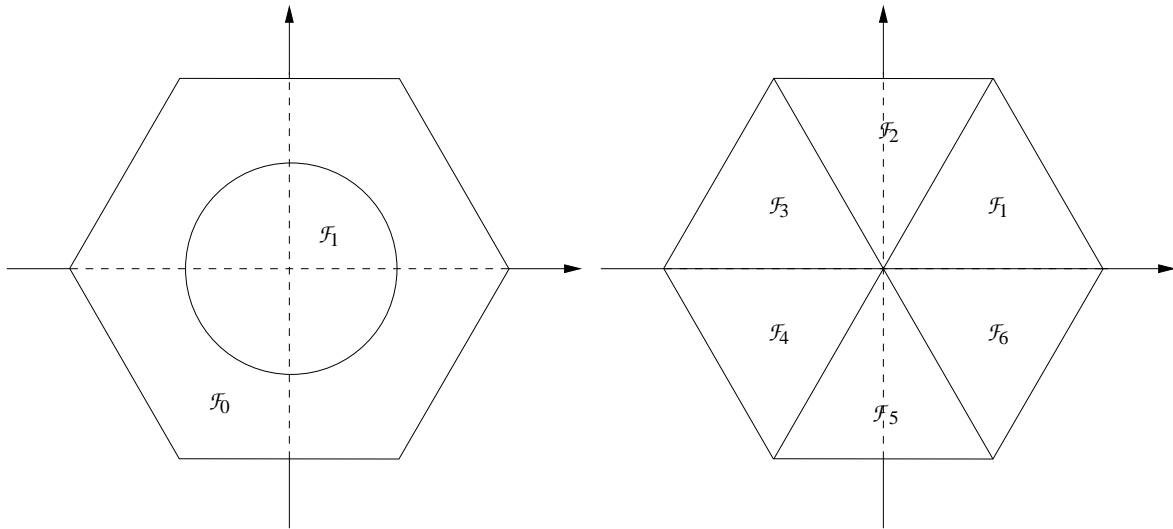


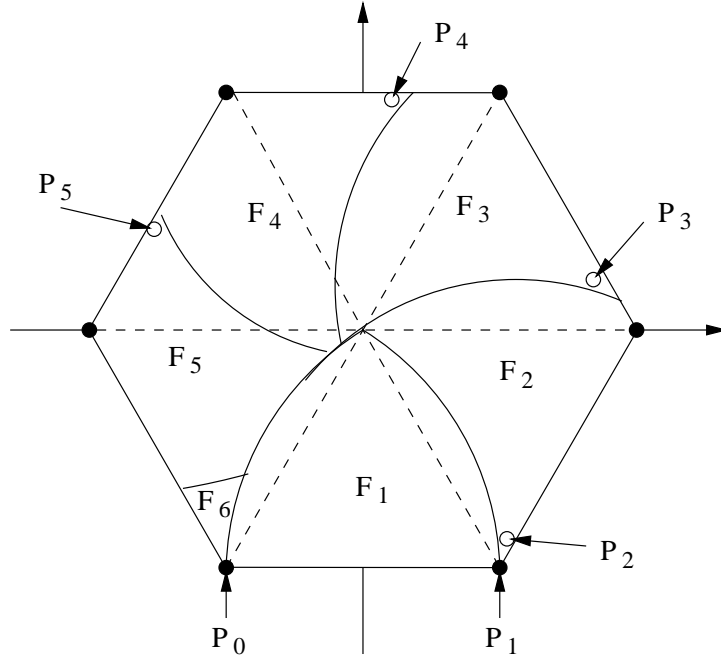
Abbildung 4.8: Kreisförmige Einbettungsregion (links) und sechs Einbettungsregionen aus gleichseitigen Dreiecken (rechts).

1. Die Kreismethode wird in Analogie zum  $2\mathbb{Z}^2$ -Gitter durchgeführt. Die Fehlerregionen sind in Abb. 4.8 (links) dargestellt. Die einzige Einbettungsregion ist  $\mathcal{E}_1 = \mathcal{F}_1$ . Man erhält  $A = 1, E = 0,302\dots$ ,  $B = 3,31\dots$ , Kapazität  $K = \text{Vol}(\mathcal{F}_1) = 1,571\dots$ , sowie  $\boxed{K/B = 0,4748\dots}$ . Die schlechtere Kapazität pro Bit ist aufgrund der besseren Platzausnutzung des Hexagons gegenüber der quadratischen Voronoi-Zelle zu erklären.
2. Eine vollständige Ausschöpfung der Hexagonfläche wird durch eine Überdeckung mit gleichseitigen Dreiecken (siehe Abb. 4.8 rechts) erreicht. Wir schätzen die benötigten Bits zur Codierung der sechs Regionen durch die Entropie ab und erhalten  $A = \log_2 6 = 2,58\dots$ ,  $E = 1$ ,  $B = A$ ,  $K = 0,866\dots$  sowie  $\boxed{K/B = 0,335}$ .

**Lemma 4.3.6** Sei  $H$  eines der Hexagone des obigen Gitters mit Schwerpunkt  $s(H)$ . Dann gilt  $m(H, s(H)) = 6$ .

*Beweisskizze.* Der zweite Teil des Beispiels 4.3.5 gibt eine Überdeckung mit sechs Fehlerregionen an, also  $m(H, s(H)) \leq 6$ .

Abb. 4.9 zeigt ein Hexagon, dessen sechs Eckpunkte hervorgehoben sind. Offensichtlich müssen zwei der Eckpunkte innerhalb einer Fehlerregion liegen um eine Überdeckung mit fünf Regionen zu ermöglichen. Aus Abstandsgründen müssen diese auf dem Rand des Hexagons benachbart sein. O.B.d.A. seien hierfür die Punkte  $P_0$  und  $P_1$  in der Abbildung gewählt. Das maximale Gebiet einer  $P_0$  und  $P_1$  enthaltenden Fehlerregion liegt in  $F_1 := B_{\sqrt{2}}(P_0) \cap B_{\sqrt{2}}(P_1) \cap H$ . Der Punkt  $P_2$  liegt in jedem Fall außerhalb. Eine Fehlerregion maximaler Größe, die unter diesen Voraussetzungen  $P_2$  enthält liegt in  $F_2 := (B_{\sqrt{2}}(P_2) - F_1) \cap H$ . Auf analoge Art und Weise konstruiert man Punkte  $P_3, P_4$  und  $P_5$  mit entsprechenden Regionen  $F_3, F_4$  und  $F_5$ . Offensichtlich wird  $H$  von den resultierenden fünf Regionen nicht vollständig überdeckt, so daß eine weitere Region  $F_6$  benötigt wird.  $\square$

Abbildung 4.9: Zum Beweis von  $m(H, s(H)) = 6$ .

## 4.4 Übergang zu diskreten Gittern quantisierter Werte

### 4.4.1 Allgemeine Betrachtung

Wir konzentrieren uns auf den für die Audiosignalverarbeitung besonders wichtigen Fall der Skalarquantisierung. Gegeben sei ein Quantisierer  $\mathcal{P} = (p_k)_{k=1}^n$  mit  $A < p_1 < \dots < p_n < B$  und  $p_k, A, B \in \mathbb{Z}$ . Der betrachtete Wertebereich ist  $[A : B] \subset \mathbb{Z}$ . Für

$$r_0 := A, \quad 1 \leq k < n : \quad r_k := \frac{p_k + p_{k+1}}{2} \quad \text{und} \quad r_n := B \quad (4.13)$$

ist  $V(p_k) = [[r_k], [r_{k+1}] - 1]$  ein System partitionierender Voronoi-Zellen. Es gilt

$$R_k = \max\{|p_k - [r_{k-1}]|, |[r_k] - 1 - p_k|\}, \quad (4.14)$$

falls für Signalwerte sowie Rückquantisierungsstufen  $p_k$  nur Werte in  $\mathbb{Z}$  zugelassen werden. Führt man jeweils zwei Fehler- und Einbettungsregionen

$$\begin{aligned} \mathcal{E}_{1,k} &:= \mathcal{F}_{1,k} := [[r_{k-1}] : p_k - 1] & \text{und} \\ \mathcal{E}_{2,k} &:= \mathcal{F}_{2,k} := [p_k : [r_k] - 1] \end{aligned} \quad (4.15)$$

ein, so sind die Ergebnisse des letzten Abschnitts mit den Mengen der jeweiligen Einbettungsvektoren

$$\begin{aligned} E_{1,k} &= \{-1, -2, \dots, -(p_k - [r_{k-1}])\} & \text{und} \\ E_{2,k} &= \{1, \dots, [r_k] - 1 - p_k\} \end{aligned} \quad (4.16)$$

anwendbar. Abb. 4.10 zeigt benachbarte Voronoi-Zellen mit den gerade definierten Regionen. Es gibt hier lediglich zwei Fehlerregionen, d.h. der Bitaufwand zur Codierung der Regionsinformation liegt konstant bei einem Bit.

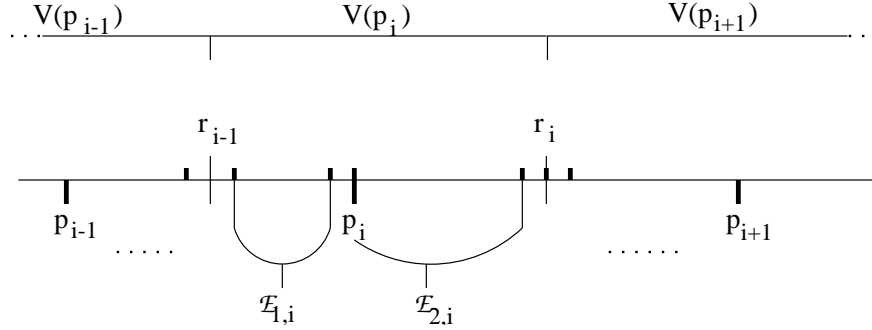


Abbildung 4.10: Benachbarte Voronoi-Zellen mit Fehler- bzw. Einbettungsregionen.

Obige Einteilung in Fehler- und Einbettungsregionen gewährleistet die Einhaltung des ursprünglichen Rekonstruktionsfehlers

$$|\tilde{Q}(Qx, e) - x| \leq \begin{cases} \frac{p_k - p_{k-1}}{2} & \text{falls } x \in \mathcal{F}_{1,k} \\ \frac{p_{k+1} - p_k}{2} & \text{falls } x \in \mathcal{F}_{2,k}, \end{cases} \quad (4.17)$$

der *ohne* Einbettung aufgetreten wäre. Geht es jedoch lediglich (im Sinne unserer Definition von  $R_k$ ) um die Garantie von  $R_k$  als maximalem Rekonstruktionsfehler, so kann z.B.

$$\begin{aligned} \mathcal{E}_{1,k} &:= \mathcal{F}_{1,k} := [\lceil r_{k-1} \rceil : \lceil \frac{r_{k-1} + r_k}{2} \rceil - 1] & \text{und} \\ \mathcal{E}_{2,k} &:= \mathcal{F}_{2,k} := [\lceil \frac{r_k + r_{k+1}}{2} \rceil : \lceil r_k \rceil - 1] \end{aligned} \quad (4.18)$$

gewählt werden. Dies garantiert in jedem Fall die Bitkapazität von  $\log_2 \left\lfloor \frac{r_k - r_{k-1}}{2} \right\rfloor$  Bits pro Einbettung in  $V(p_k)$ .

#### 4.4.2 Lineare Quantisierung

Der Spezialfall der linearen Quantisierung soll an einem für Anwendungen wichtigen Beispiel behandelt werden. Hierbei sind die Intervalllängen durch Zweierpotenzen gegeben. Weiterhin wird auf die Beschränkung  $p_k \in \mathbb{Z}$  verzichtet. Dies hat auf den maximalen Rekonstruktionsfehler keine weiteren Auswirkungen, bei der Rückquantisierung kommt natürlich der Wert  $p_k$  innerhalb der Einbettung niemals vor. Der Wertebereich bestehe aus  $2^N$  Werten, O.B.d.A.  $[0 : 2^N - 1]$ . Wir betrachten die Quantisierung von  $N$  auf  $K < N - 1$  Bits gegeben durch eine Voronoi-Partitionierung

$$V(p_k) := [k2^{N-K} : (k+1)2^{N-K} - 1]. \quad (4.19)$$

Es gelten

$$[0 : 2^N - 1] = \bigsqcup_{k=0}^{2^K - 1} V(p_k) \quad (4.20)$$

sowie

$$p_k = k2^{N-K} + \frac{2^{N-K} - 1}{2}. \quad (4.21)$$

Somit ist ein *Midrise*-Quantisierer, d.h.  $\forall k : p_k \neq 0$ , gegeben. Weiterhin gilt

$$R_k = \max_{x \in V(p_k)} |x - p_k| = \frac{2^{N-K} - 1}{2}. \quad (4.22)$$

Fehler- und Einbettungsregionen seien gegeben durch

$$\begin{aligned} \mathcal{E}_{1,k} &:= \mathcal{F}_{1,k} := [k2^{N-K} : (k + \frac{1}{2})2^{N-K}] \quad \text{und} \\ \mathcal{E}_{2,k} &:= \mathcal{F}_{2,k} := [(k + \frac{1}{2})2^{N-K} + 1 : (k + 1)2^{N-K} - 1]. \end{aligned} \quad (4.23)$$

Die Mengen der Einbettungsvektoren sind in diesem Fall für alle  $0 \leq k < 2^K$  gleich

$$\begin{aligned} E_1 &= E_{1,k} = \{0, -1, \dots, -2^{N-K-1}\} - 0,5 \quad \text{und} \\ E_2 &= E_{2,k} = \{0, 1, \dots, 2^{N-K-1}\} + 0,5 \end{aligned} \quad (4.24)$$

wobei die Summanden (punktweise) dafür sorgen, daß  $\tilde{Q}(Qx, e) \in \mathbb{Z}$  für  $e \in E_1 \cup E_2$ . Unter diesen Voraussetzungen gilt  $\forall e \in E_1 \cup E_2$ :

$$|\tilde{Q}(Qx, e) - x| \leq 2^{N-K-1} - 1. \quad (4.25)$$

Die *Einbettungsbitbreite* ist in diesem Fall  $N - K - 1$  (und somit  $> 0$  nach Voraussetzung). Aus den Ergebnissen des letzten Abschnitts folgt, daß der eingebettete Vektor  $e$  aus  $\tilde{Q}(Qx, e)$  ablesbar ist.

## 4.5 Auswirkung von Skalierungen

Bei Verwendung der Einbettung in Audiocodern ist eine bestimmte Art der Datenvorverarbeitung zu beachten: die Skalierung. Mit Hilfe dieser verlustfreien Technik wird häufig für eine Vorreduktion der Redundanz des Audiosignals gesorgt. Genauer geht es darum, Amplitudenbeschränkungen des Signals systematisch auszunutzen. Dies wird durch folgendes Beispiel motiviert: Ist ein Wertebereich  $Z_N := [-2^{N-1} : 2^{N-1} - 1] \subset \mathbb{Z}$  für die Amplitudenwerte eines Signals gegeben, und wird eine  $N$ -Bit Zahldarstellung verwendet, so lassen sich alle  $x \in R_N := [-2^{N-1}, 2^{N-1} - 1] \subset \mathbb{R}$  unter Einhaltung eines maximalen Quantisierungsfehlers von  $0,5$  linear auf  $Z_N$  quantisieren. Ist bekannt, daß alle Signalwerte eines bestimmten Signalabschnitts sogar in  $R_{N-K}$  liegen, reichen entsprechend  $N - K$  Bits zur Einhaltung dieses maximalen Fehlers. Zusätzlich speichert man in diesen Fällen noch zwecks Decodierbarkeit die Information, daß die Werte in  $R_{N-K}$  liegen. In der Praxis bilden diese Signalabschnitte die sogenannten *Skalierungsblöcke*. Weiterhin werden die möglichen Amplitudenbereiche oft feiner repräsentiert als nur in Zweierpotenzschritten. Der Begriff *Skalierungsfaktor* rührt von folgender Vorgehensweise her:

1. Gegeben sei Skalierungsblock  $A \in [0 : 2^N - 1]^n$  (O.B.d.A.), sowie eine Folge  $0 = s_0 < s_1 < \dots < s_M = 2^N$ .
2. Bestimme minimales  $j$  mit  $A \in [0 : s_j]^n$ .
3. Skalieren  $\forall k : S_j(a_k) := a_k \frac{2^N}{s_j}$ , wo  $S_j(a_k) \in \mathbb{Q}$  i.a. gilt (hier wird plattformabhängig, z.B. mit Fließkommaarithmetik, gearbeitet).
4. Quantisiere  $\forall k : S_j(a_k) \mapsto qS_j(a_k)$ , wo  $q$  ein geeigneter Quantisierer mit zugehörigem Dequantisierer  $\tilde{q}$  ist, siehe (4.4).
5. Nach einer eventuellen Kompression und Übertragung von  $qS_j(a_k)$  sowie  $j$  zu einem Decodierer, findet die Rückquantisierung,

$$qS_j(a_k) \mapsto \tilde{q}qS_j(a_k) =: \tilde{S}_j(a_k) \quad (4.26)$$

statt.

6. Rückskalierung  $\forall k$  :

$$\tilde{S}_j(a_k) \mapsto \tilde{S}_j(a_k) \frac{s_j}{2^N} =: \tilde{a}_k. \quad (4.27)$$

Die Skalierung bewirkt eine verlustfreie Redundanzreduktion. Da hierdurch der Quantisierungsfehler verkleinert wird, steht ein Teil der Kapazität, wie sie unter alleiniger Verwendung der Quantisierungsvergrößerung errechnet wird, zur Dateneinbettung nicht mehr zur Verfügung und muß von der totalen gEinbettungskapazität abgezogen werden. Anschaulich stellt sich dies folgendermaßen dar: Gegeben sei ein Skalierungsfaktor  $s > 1$ . Ein  $a \in [A, B]$  wird durch  $a \mapsto as$  auf das Intervall  $[As, Bs]$  abgebildet und dort mit Hilfe eines geeigneten Quantisierers  $as \mapsto \tilde{a}s$  quantisiert, so daß der Rekonstruktionsfehler  $R$  entsteht. Durch Rückskalierung  $\tilde{a}s \mapsto \tilde{a}s/s$  wird  $\tilde{a}s$  ins Intervall  $[A, B]$  zurücktransportiert. Der Rekonstruktionsfehler liegt insgesamt bei

$$\left| a - \frac{\tilde{a}s}{s} \right| = \frac{1}{s} |as - \tilde{a}s| = \frac{R}{s}. \quad (4.28)$$

Wir betrachten dies genauer im Diskreten am Beispiel des Midtread-Quantisierers

$$\Delta_k : a \mapsto \text{round} \left( \frac{a}{2^k} \right) \cdot 2^k. \quad (4.29)$$

Es gilt

$$\left| a - \Delta_k(a) \right| = \left| \frac{a}{2^k} - \text{round} \left( \frac{a}{2^k} \right) \right| \cdot 2^k \leq \frac{1}{2} \cdot 2^k.$$

Für ein  $a := a_\ell \in A$  aus obigem Skalierungsalgorithmus ist für entsprechendes  $0 \leq j \leq M$

$$\begin{aligned} |a - \tilde{a}| &= \left| a - \Delta_k \left( \frac{a}{s_j} 2^N \right) \frac{s_j}{2^N} \right| \\ &= \left| a \frac{2^N}{s_j} - \Delta_k \left( a \frac{2^N}{s_j} \right) \right| \frac{s_j}{2^N} \leq 2^{k-1} \frac{s_j}{2^N}. \end{aligned}$$

Damit gilt für  $s_j = 2^p$  :  $|a - \tilde{a}| \leq 2^{k-1+p-N}$ . Für die „Extremfälle“ gilt  $s_j = 2^N \Rightarrow |a - \tilde{a}| \leq 2^{k-1}$  und  $s_j = 2^{N-k+1} \Rightarrow |a - \tilde{a}| \leq 1$ . Im letzteren Fall ist die Skalierung optimal auf den Quantisierer abgestimmt, so daß kein Platz zur Einbettung bestehen bleibt.

Im MPEG-Standard [18] sind die Skalierungsfaktoren durch  $(\sqrt[3]{2})^{3-n}$  für  $n \in \{0, \dots, 62\}$  gegeben. Für deren Übertragung werden somit  $\sim 6$  Bit pro Skalierungsblock aufgewendet. Somit kommt den Skalierungsfaktoren in diesem Standard große Bedeutung zu. Skalierungsfaktoren werden im Rahmen der Anwendung des Verfahrens auf MPEG Codecs, wie in Kapitel 6 beschrieben, berücksichtigt.

## 4.6 Fehlertolerante Einbettungsverfahren

Genauso wie im Fall des Watermarkings ist es bei den hier vorgestellten Verfahren wichtig, die eingebettete Information gegenüber Fehlern zu schützen. Es stellt sich heraus, daß eine Unterscheidung in verfahrensinhärente und nicht inhärente Fehler sinnvoll ist. Erstere müssen unbedingt zum Erhalt der eingebetteten Information berücksichtigt werden, letztere betreffen in der Weiterverarbeitung auftretende Störungen.

Unter „Fehlern“ sind allgemein alle Abweichungen eines Signals von seinem ursprünglichen oder durch eine (Signal)verarbeitung angestrebten Zustand zu verstehen. Dies sind z.B. Änderungen ganzer Datenblöcke, einzelner Datenworte oder Bits, sowie Arithmetikfehler oder Unzulänglichkeiten eines Signalverarbeitungsverfahrens. Ursache dieser Fehler können Paketverluste in Übertragungskanälen, Materialfehler auf Datenträgern, unzulängliche oder in der Genauigkeit begrenzte Rechnerarithmetik oder nur approximativ arbeitende Verfahren sein. Für die transformationscoderbasierte Einbettung inhärente Fehler sind in Digitalrechnern natürlich Arithmetikfehler. Weiterhin liefern viele relevante Transformationen wie bereits erwähnt einen gewissen Rekonstruktionsfehler bzw. lediglich Quasi-Invertierbarkeit. Ein prominentes Beispiel ist die in MPEG-1 Audio verwendete NPR-Filterbank [18], die einen Rekonstruktionsfehler von  $\sim 90\text{dB}$  liefert, also ein ursprüngliches Signal „nur“ auf 4 – 5 Dezimalstellen genau approximiert.

Auf die nicht verfahrensinhärenten Fehlerquellen soll im folgenden ebenfalls kurz eingegangen werden.

#### 4.6.1 Arithmetik- und Rekonstruktionsfehler

Die hier betrachteten Arithmetik- und Rekonstruktionsfehler sind in Form eines Abweichungsterms  $e$  vom Originalsignal  $x$  modellierbar. Das fehlerhafte Signal ist dann  $x' = x + e$  wobei  $|e|$  im Vergleich zum Wertebereich von  $x$  klein ist. Beispiele für solche Fehler sind

- Fehler, die aufgrund der begrenzten Darstellungsgenauigkeit von Fließkommazahlen bei arithmetischen Operationen auftreten bzw. Rundungsfehler,
- Rundungsfehler bei Datenformatumwandlungen, z.B. von Fließ- in Festkommadarstellung,
- Approximationsfehler wie z.B. bei obigen NPR-Transformationen  $T$ , so daß es eine Abbildung  $\tilde{T}$  mit  $\tilde{T}Tx = x + e$  bzw.  $T\tilde{T}x = x + e'$  gibt. Eine realistische Forderung an die Rekonstruktionsfehler  $e, e'$  ist z.B. bei einem Wertebereich  $x \in [-2^L, 2^L]$ , daß  $|e|, |e'| < 2^\ell$  für  $\ell \ll L$  gilt.

Zur Illustration der Vorgehensweise betrachten wir wieder kontinuierliche Wertebereiche mit Vektorquantisierung. Für eine gegebene Einbettungsregion  $\mathcal{E}_{i,k}$  mit Einbettungsvektoren  $E_{i,k}$  erreichen wir Redundanz, indem wir aus  $E_{i,k}$  nur bestimmte Teilmengen von Vektoren zur Rekonstruktion zulassen.

**Definition 4.6.1** Eine Menge  $H \subset \mathbb{R}^n$  hat Minimaldistanz größer  $\varepsilon$  gdw.  $\forall x \neq y \in H : d(x, y) > \varepsilon$ .

**Lemma 4.6.2** Ist  $H \subset \mathbb{R}^n$  beschränkt mit Minimaldistanz größer  $\varepsilon$ , so ist  $H$  endlich.

*Beweis.* Legt man um alle  $h \in H$  Kugeln  $B_{\frac{\varepsilon}{2}}(h)$ , so berühren sich laut Definition keine dieser Kugeln. Da  $H$  beschränkt ist, gibt es eine Kugel mit endlichem Radius  $r$  und Mittelpunkt  $g$ , so daß  $H \subseteq B_r(g)$  und daher  $\bigsqcup_{h \in H} B_{\frac{\varepsilon}{2}}(h) \subseteq B_{r+\frac{\varepsilon}{2}}(g)$ . Da keine der Kugeln um die  $h \in H$

einen gemeinsamen Raum überdecken, kann es höchstens  $\left\lfloor \frac{\text{Vol } B_{r+\frac{\varepsilon}{2}}(g)}{\text{Vol } B_{\frac{\varepsilon}{2}}(h)} \right\rfloor$  viele davon geben.

Also ist  $H$  endlich. □

Gegeben sei ein Quantisierer  $\mathcal{P} = \{P_0, P_1, \dots\}$  des  $\mathbb{R}^n$  mit Voronoi-Zellen  $V(P_k)$  und zugehörigen Einbettungsregionen  $\mathcal{E}_{i,k}, 0 \leq i \leq m(k)$ . Für das Einbettungsvolumen

$$G := \bigcup_k \bigcup_{1 \leq i \leq m(k)} \mathcal{E}_{i,k}$$

und maximalen Arithmetik- oder Rekonstruktionsfehler  $\varepsilon/2$  betrachten wir eine beliebige Menge  $H$  mit Minimaldistanz  $\varepsilon$ . Sei  $T$  die den Fehler verursachende Transformation. Da für  $g \neq h \in H$  stets  $B_{\frac{\varepsilon}{2}}(g) \cap B_{\frac{\varepsilon}{2}}(h) = \emptyset$  und  $\forall h \in H : d(Th, h) \leq \varepsilon/2$  gilt, folgt  $Th \in B_{\frac{\varepsilon}{2}}(g) \Rightarrow g = h$ . Arithmetik- und Rekonstruktionsfehler  $\leq \frac{\varepsilon}{2}$  können somit korrigiert werden, falls  $H$  Minimaldistanz größer  $\varepsilon$  besitzt. Die Menge der Einbettungsvektoren ist also mit einer entsprechenden Minimaldistanz zu wählen. Als Beispiel betrachten wir nochmals den

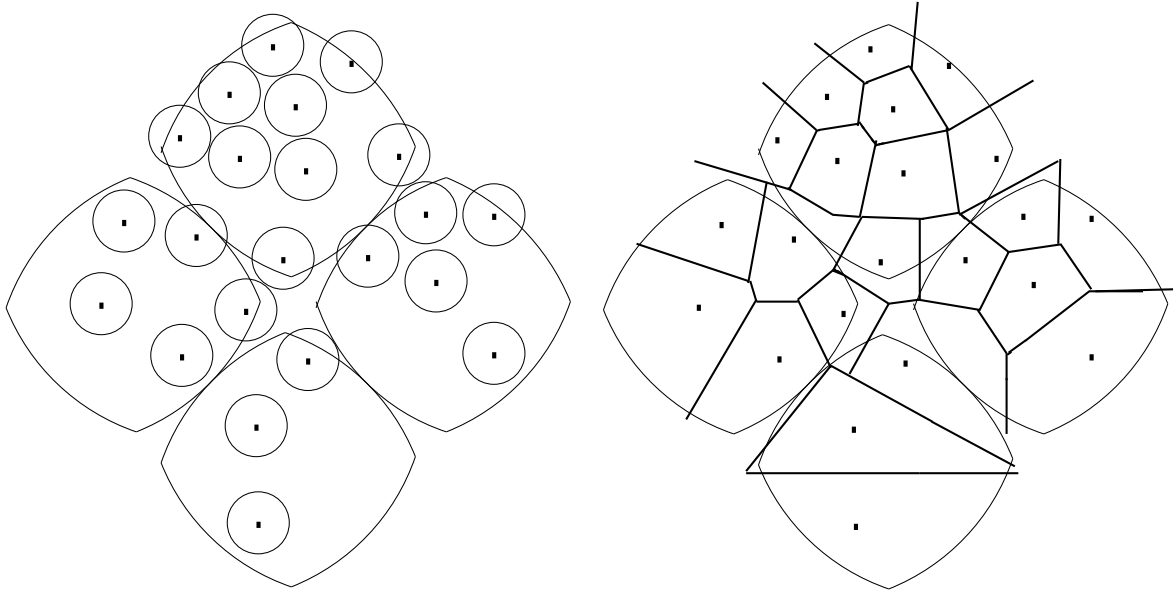


Abbildung 4.11: Menge mit Minimaldistanz größer  $\varepsilon$  (zweifacher Kreisradius) dargestellt als Menge von Punkten innerhalb der Einbettungsregionen (links), sowie Aufteilung in Voronoi-Zellen bezüglich dieser Punkte (rechts).

2. Teil von Beispiel 4.2.3. Abb. 4.11 zeigt eine Menge mit Minimaldistanz  $\varepsilon$  dargestellt als Punkte innerhalb der Einbettungsregionen sowie (links) die zweidimensionalen Kugeln um die Elemente der Menge. Es können jedoch nicht nur Fehler mit  $Th \in B_{\frac{\varepsilon}{2}}(h)$  korrigiert werden sondern auch solche, die in der Voronoi-Zelle  $V(h)$  bezüglich  $H$ , also  $Th \in V(h)$ , liegen. In Abb. 4.11 (rechts) ist die Aufteilung in Voronoi-Zellen für die betrachtete Menge skizziert. Die Maximierung der Kapazität an einbettbarer Information bei gegebenen Einbettungsregionen, unter der Voraussetzung der Korrigierbarkeit eines Arithmetikfehlers  $\leq \varepsilon/2$ , entspricht offensichtlich dem folgenden

**Sphären-Packproblem mit Nebenbedingungen**

Maximiere  $\#H : H \subseteq \bigcup_{i,k} \mathcal{E}_{i,k} \wedge \forall h \neq h' \in H : B_{\frac{\varepsilon}{2}}(h) \cap B_{\frac{\varepsilon}{2}}(h') = \emptyset$ .

Wie Abb. 4.11 zeigt, kann auch Platz außerhalb der  $\mathcal{E}_{i,k}$  innerhalb der Sphären liegen. Dies ist Platz, der auch von den  $\mathcal{F}_{0,k}$ 's eingenommen wird. Im Fall nichtleerer  $\mathcal{F}_{0,k}$ 's kann somit mehr Platz zur Einbettung mit Fehlerkorrektur ausgenutzt werden als ohne. Sind die  $\mathcal{F}_{0,k}$  jedoch alle leer, ist dies nicht möglich. Die Entkopplung der Fehler- und Einbettungsregionen bringt somit an dieser Stelle einen Vorteil, der über den Gewinn an Kapazität pro Bit hinausgeht. Abschließend sollen für den Fall der Skalarquantisierung konkrete Beispiele für endliche Wertebereiche gegeben werden. Die Beispiele haben die Gemeinsamkeit, daß die Einbettungsregionen den gesamten betrachteten Wertebereich ausschöpfen.

**Beispiel 4.6.3** 1. Der Wertebereich sei  $[0 : 2^N - 1]$ . Im Falle des Midrise-Quantisierers (4.21) sind Voronoi-Zellen der Länge  $2^M = 2^L \cdot 2^{M-L}$  gegeben. Dann besitzt

$$H_L := \{2^{L-1} + k2^L | 1 \leq k \leq 2^{N-L}\}$$

Minimaldistanz  $\varepsilon$  für alle  $\varepsilon < 2^L$ . Bezüglich des gewählten Wertebereichs  $[0 : 2^N - 1] \subset \mathbb{Z}$  ist diese Einteilung nicht ganz ideal, da jeweils ein Gitterpunkt genau zwischen zwei benachbarten Elementen aus  $H_L$  liegt. Ein Vorteil der Konstruktion ist jedoch, daß in allen Voronoi-Zellen gleichviele Elemente aus  $H_L$  liegen.

2. Wir betrachten zu  $\lambda \in \mathbb{N}$

$$H^\lambda := \{\lambda + 1 + (2\lambda + 1)k | k \in \mathbb{Z}\}.$$

$H^\lambda$  hat Minimaldistanz  $(2\lambda + 1)$ . Über  $\mathbb{Z}$  können somit  $\lambda$  Fehler korrigiert werden. Bei dieser Konstruktion liegen im Gegensatz zum ersten Beispiel nicht immer gleichviele Elemente aus  $H^\lambda$  in Intervallen von Zweierpotenzlänge bzw. obigen Voronoi-Zellen.

## 4.6.2 Direkte Biteinbettung und Arithmetische Codes

Charakteristisch für Arithmetikfehler (AR-Fehler) ist deren Modellierbarkeit durch Addition bzw. Subtraktion eines Fehlerterms. AR-Fehler sind daher in gewisser Weise ähnlich unserer Einbettungsmethode. Etwas anders verhält sich dies bei der direkten Biteinbettung  $E_1$  aus dem ersten Abschnitt des Kapitels. Da diese Einbettungsart bei den beschriebenen Realisationen der in den nächsten Kapiteln beschriebenen Codecs eine Rolle spielt, wird hier ein Konzept zur Behandlung von AR-Fehlern für die Einbettung vom Typ  $E_1$  vorgestellt werden. Dabei handelt es sich um eine Kombination von AR-Codes [47] und dem hier vorgestellten Einbettungskonzept.

Wir gehen zunächst kurz auf AR-Codes ein. Diese sind allgemein von der Form

$$AI + B := \{An + B | n \in I\}$$

mit ganzzahligen  $0 \leq B < A$ ,  $A \geq 2$ , und  $I \subset \mathbb{Z}$ . Im folgenden interessieren uns zu festem  $A$  spezielle  $I$ , die über das *arithmetische Gewicht*

$$w_r(N) := \min\{wt(N) | N = \sum_{i=0}^n a_i r^i, |a_i| < r\}$$

von  $N \in \mathbb{N}$  zur Basis  $r$  definiert sind. Mit  $wt(a)$  wird hier das Hamminggewicht von  $a$ , also die Anzahl der  $i$  mit  $a_i \neq 0$ , bezeichnet. Wird Arithmetik zur Basis  $r$  verwendet, so entsteht ein *einfacher Fehler* in diesem Modell durch Addition eines Terms  $a_i r^i$  wo  $|a_i| < r$ .

Der *arithmetische Abstand* zwischen  $N_1 \leq N_2$  ist definiert als  $w_r(N_2 - N_1)$ . Für  $\mathcal{C} \subseteq \mathbb{Z}, |\mathcal{C}| \geq 2$ , heißt

$$d = d(\mathcal{C}) := \min\{w_r(N_2 - N_1) \mid N_1, N_2 \in \mathcal{C}, N_1 < N_2\}$$

die *Minimaldistanz* von  $\mathcal{C}$ .

**Satz 4.6.4 ([47])** *Hat  $\mathcal{C} \subseteq \mathbb{Z}$  die Minimaldistanz  $d$ , so können bis zu  $d - 1$  Fehler erkannt und bis zu  $\lfloor \frac{d-1}{2} \rfloor$  Fehler korrigiert werden.*

Für ganzzahlige  $A, d, r \geq 2$  sei

$$M_r(A, d) := \min\{n \in \mathbb{N} \mid w_r(An) < d\}.$$

$M_r(A, d)$  existiert und es gilt

**Satz 4.6.5 ([47])** *Für*

$$I := [0 : M_r(A, d) - 1] \quad \text{und} \quad J := [\lceil -\frac{1}{2}M_r(A, d - 1) \rceil : \lceil \frac{1}{2}M_r(A, d) \rceil - 1]$$

*haben  $AI$  und  $AJ$  Minimaldistanz  $\geq d$ .*

Diese Codes können die für uns interessanten Arithmetik- und Rekonstruktionsfehler korrigieren. Problematisch sind jedoch die relativ langen Codewörter wie sie schon für kleine  $d$  auftreten. Tabelle 4.12 zeigt den Codewertebereich  $M_2$  und die Codelänge in Bits für AR-

$A$	$M_2(A, 3)$	$\lceil \log_2(A) \rceil$	$\log_2(M_2(A, 3))$
11	3	4	1, 58
13	5	4	2, 32
19	27	5	4, 75
23	89	5	6, 47
29	565	5	9, 14

Abbildung 4.12: Codewertebereich  $M_2$  und Codelänge in Bits für AR-Codes des Typs  $AI$  mit  $d = 3$ .

Codes des Typs  $AI$  der Distanz 3. Da  $w_2(A) = w_2(2^\alpha A)$ , sind in der Tabelle O.B.d.A. nur ungerade  $A$ 's aufgeführt.

Bei 16-Bit Wortlänge wie sie in der Audiokompression üblich ist, sind 4-5 Bit Codeoverhead normalerweise nicht praktikabel: z.B. liefern 5 Bit Codebreite,  $2^5 = 32$ , laut Tabelle zusätzlichen Bedarf nach  $\sim 5$  Bit Overhead, also insgesamt 10 Bits, die einzubetten wären. Soviel Platz steht jedoch normalerweise nicht zu Verfügung. Realistisch wäre eine maximale Redundanz von 2-3 Bits für beliebige Codewortbreiten.

Um dies zu realisieren, werden die AR-Codes im Stil des letzten Abschnitts modifiziert. Da in einem Code  $\mathcal{C} = A\mathbb{Z} + B$  für  $c \neq d \in \mathcal{C}$  stets  $|c - d| = A|n_1 - n_2| \geq A$  mit entsprechenden  $n_1, n_2 \in \mathbb{Z}$  gilt, können trivialerweise alle Fehler vom Betrag kleiner  $A/2$  korrigiert werden. Soll eine Einbettung mittels  $E_1$  für das Codewort  $(d_k \dots d_1) = d := Ac + B$  mit  $c = (c_n \dots c_1)$

$$E_1 : (b_m \dots b_{k+1} b_k \dots b_1), (d_k \dots d_1) \mapsto (b_m \dots b_{k+1} d_k \dots d_1) =: \tilde{b}$$

vorgenommen werden, so daß aus  $\tilde{b}$  nach einer fehlerbehafteten Transformation  $T\tilde{b}$  mit  $|T\tilde{b} - \tilde{b}| < A/2$  das Wort  $d$  und somit  $c$  via  $(d - B)\text{div } A$  wiedergewonnen werden kann, muß  $k$  in

Abhängigkeit von  $n$  hinreichend groß genug gewählt werden. Gleichzeitig sollte  $k$  natürlich nicht größer als nötig sein.

Wir bestimmen

$$b(A, B) := \min\{b \in \mathbb{N} \mid (2^n - 1)A + B + \frac{A}{2} < 2^{n+b}\},$$

die notwendige Erweiterung der Wortlänge  $n$  zum Erhalt eines ausreichend langen Codes. Dieser Code hat die Breite  $k := n + b(A, B)$ . Ist ein Unterlauf in der Zahldarstellung nicht gestattet, so muß zusätzlich  $B \geq A/2$  gewählt werden, damit Fehler  $< A/2$  erkannt werden, und Fehler, die wie eine Subtraktion wirken, keinen Unterlauf erzeugen. Sei  $B := A/2$ , dann gilt

$$(2^n - 1)A + B + \frac{A}{2} < 2^{n+b} \iff \log_2 A < b,$$

und somit  $b(A, A/2) = \lceil \log_2 A \rceil$ . Wird hier lediglich  $B < A/2$  gewählt, so können Fehler  $\leq B$  korrigiert werden. Es gilt dann

$$b(A, 0) \leq b(A, B) \leq \lceil \log_2 A \rceil.$$

Sind Unterläufe erlaubt, wie z.B. beim Rechnen im Zweierkomplement, so kann die Einschränkung bezüglich  $B$  fallen gelassen werden. Allerdings ist die Erweiterung der Wortlänge, z.B. auf

$$b'(A, B) := \min\{b \in \mathbb{N} \mid 2^n A + B < 2^{n+b}\},$$

zu modifizieren, da ansonsten ein Unterlauf zu einer Verletzung des minimalen Codeabstands führen könnte. Eine grobe Abschätzung für  $B := A$  liefert

$$A(2^n + 1) = 2^n A + A < 2^{n+b} \iff b > \log_2 \frac{2^n + 1}{2^n} + \log A.$$

**Beispiel 4.6.6** 1.  $A := 3, B := 0$  mit möglichem Unterlauf. Dies korrigiert Fehler  $\pm 1$  über  $\mathbb{Z}$ . Da  $2^n A + 0 < 2^{n+b} \iff \log_2 A < b$  und  $1 < \log_2 3 < 2$ , gilt für die zusätzliche Codebreite  $b'(3, 0) = 2$ .

2.  $A := 8, B := 3$  ohne Unterlaufannahme. Es gilt

$$(2^n - 1)8 + 7 < 2^{n+b} \iff b > \log_2((2^n - 1)8 + 7) - n$$

und  $2 < \log_2(8 \cdot 2^n - 1) - n < 3$ , also  $b(8, 3) = 3$ . In dieser Codierung können Fehler kleiner  $\pm 3$  korrigiert werden.

### 4.6.3 Stabilität gegenüber sonstigen Fehlern

Sonstige Fehler und Signalveränderungen spielen sich in unserem Fall im Zeitbereich, d.h. nach Einbettung und Umkehr- bzw. Rekonstruktionstransformation  $T^{-1}$  bzw.  $\tilde{T}$ , laut Algorithmus 4.2, ab. Wäre dies nicht der Fall, könnten Standardverfahren (z.B. lineare Codes zur Vermeidung von Bitfehlern im Falle der direkten Biteinbettung nach  $E_1$ ) zur Fehlerkorrektur angewendet werden.

Die Veränderungen im Zeitbereich übertragen sich via  $T$  auf den Transformationsbereich auf dem die Decoder für die eingebettete Information arbeiten. Im Falle eines Bitfehlers in Bit  $j$  an der  $i$ -ten Stelle eines Signalblocks  $x \in \mathbb{R}^n$  ist dies

$$T(x + (2^j)_i) = Tx + T(2^j)_i = Tx + (a_{1i}, \dots, a_{ni})^T 2^j$$

mit  $j \in \mathbb{Z}$  falls  $T = (a_{ij})_{1 \leq i, j \leq n}$  und  $(2^j)_i = 2^j(0, \dots, 0, 1, 0, \dots, 0)^\top$  wo die Eins an der  $i$ -ten Stelle steht. Falls  $2^j$  klein ist, führt dies bei Filterbänken (wo oft  $|a_{ij}| \leq 1$  bzw. klein) u.U. nur zu einem kleinen arithmetischen Fehler, der mit den obigen Methoden korrigierbar ist. Im allgemeinen kann ein solcher Fehler mit hinreichend großem Betrag aber den vollständigen Verlust der eingebetteten Information zur Folge haben. In dieser Hinsicht liefert also das Einbettungsverfahren einen nicht optimalen Störungsschutz. Andersherum kann allerdings argumentiert werden, daß sich solch große Fehler sowieso negativ auf die perzeptuelle Qualität des Signals auswirken werden und daher auch ohne Einbettung Beeinträchtigungen entstehen. Eine weitere Quelle für Signalmodifikationen sind Signalverarbeitungsoperationen wie Lautstärkeveränderungen, Equalization oder das Mischen von Audiosignalen. Diese verändern im allgemeinen den Charakter des Signals dermaßen, daß die Einbettung der Informationen nach psychoakustischen Gesichtspunkten sowieso in der angewendeten Art und Weise nicht mehr sinnvoll ist. Im Fall solcher Operationen wäre eine erneute Einbettung, eine Umbettung, der Informationen durchzuführen. Ansätze zur Änderung einer gegebenen Einbettung nach einer solchen erfolgten Modifikation werden später untersucht.

## 4.7 Variable Einbettungsbreiten, Blockbildung und Rekonstruktion

Die eingebetteten Vektoren können nach Konstruktion aus den Datenworten in  $\mathcal{E}_{i,k}$  abgelesen werden. Dies gilt jedoch nur, wenn

1. das Einbettungsverfahren auf allen Datenworten eines Blocks angewendet wurde, und
2. für alle Datenworte eines Blocks eine bestimmte anfangs vereinbarte Einbettungsart verwendet wurde.

Die psychoakustischen Audiocodex, die für die Verwendung mit dem Einbettungsmechanismus in Frage kommen, erlauben jedoch normalerweise weder eine durchgehende Verwendung der Einbettung auf allen Transformationswerten, noch eine konstante Einbettungsbitbreite bzw. feste Einbettungsart. Dies liegt an dem i.a. stark variierenden Signal-Maskierungsschwellenabstand und der daraus resultierenden unterschiedlichen Bitkapazität. Es stellt sich sogar heraus, daß es häufig einige Spektralbereiche gibt, so daß auf den entsprechenden Transformationskoeffizienten gar keine Einbettung vorgenommen kann. In jedem Fall ist es sinnvoll, Transformationskoeffizienten zu Blöcken zusammenzufassen, um zumindest lokal eine konstante Einbettungsbitbreite zu erreichen. Im Beispiel des MPEG-Standards [18], Layer II, kommen hierzu z.B. die Skalierungsblöcke der Länge von jeweils 12 Subbandwerten in Betracht.

Insgesamt stellen sich folgende Fragen:

1. Wie kann ein Decoder verschiedene Einbettungsarten unterscheiden?
2. Wie kann ein Decoder eine vorhandene Einbettung erkennen?
3. Wie können Blöcke aus Transformationswerten geschickt organisiert werden, so daß zu große Redundanz bei der Lösung der beiden vorherigen Probleme vermieden wird?

### 4.7.1 Variable Bitraten und Einbettungsbreiten

Eine intuitive Lösung zur Unterscheidung variabler Einbettungsarten ist die Verwendung disjunkter Einbettungsregionen. Dabei reicht es prinzipiell aus, wenn nur in einigen Vektoren eines Verarbeitungsblocks diese spezielle Einbettungsart verwendet wird. Innerhalb dieser Vektoren wird die Einbettungsart und -bitbreite für die restlichen Werte des Verarbeitungsblocks geklärt.

Wir betrachten  $N$  Quantisierungs- bzw. Einbettungsarten. Seien für  $1 \leq n < N, k \in I \subset \mathbb{Z}, 1 \leq i \leq m(k)$  Einbettungsregionen  $\mathcal{E}_{i,k}^n$  gegeben, für festes  $n$  die Einbettungsregionen des  $n$ -ten Quantisierers.

Gilt  $\forall 1 \leq n < N, k \in I \subset \mathbb{Z}, 1 \leq i \leq m(k)$ :

$$\mathcal{I}_{i,k}^n := \mathcal{E}_{i,k}^n - \bigcup_{(j,r,\ell) \neq (i,k,n)} \mathcal{E}_{j,r}^\ell \neq \emptyset, \tag{4.30}$$

so ist eine vorhandene Einbettung in  $\mathcal{I}_{i,k}^n$  eindeutig einem  $n \in [1 : N]$  zuzuordnen. Obige Forderung nach disjunkten Einbettungsregionen wurde dabei abgeschwächt. Soll eine vorhandene Einbettung aus bestimmten vereinbarten Vektoren eines Signalblocks abgelesen werden, ist es notwendig, daß für diese Werte stets eine Einbettung möglich, also

$$\text{für alle } 1 \leq n \leq N : \bigcup_k \bigcup_{i \geq 1} \mathcal{F}_{i,k}^n = D,$$

wobei  $D$  der Wertebereich der VQ ist, gilt. Dies entspricht der Forderung, daß stets  $\mathcal{F}_{0,k}^n = \emptyset$  gilt.

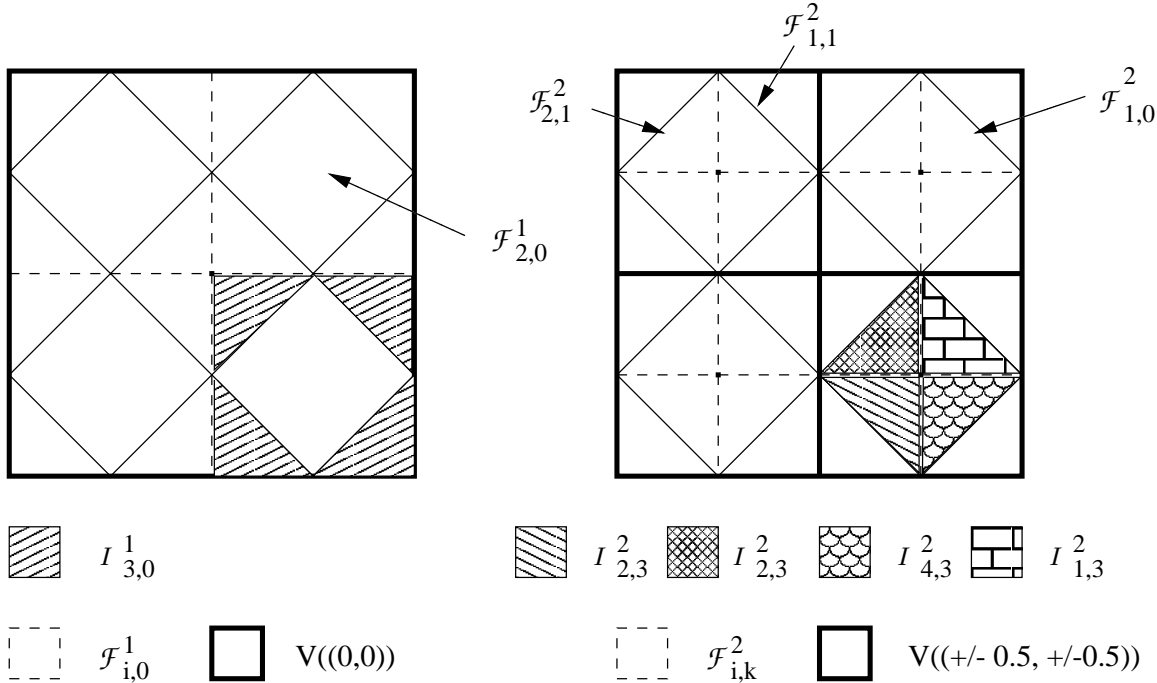


Abbildung 4.13: Regionen  $\mathcal{E}_{i,k}^1, \mathcal{F}_{i,k}^1$  und  $\mathcal{I}_{i,k}^1$  des  $2\mathbb{Z}^2$ -Gitters (links) sowie  $\mathcal{E}_{i,k}^2, \mathcal{F}_{i,k}^2$  und  $\mathcal{I}_{i,k}^2$  im  $\mathbb{Z}^2 + (0.5, 0.5)$ -Gitter (rechts).

**Beispiel 4.7.1** 1.  $D = \mathbb{R}^2$  mit  $\mathcal{P}_1 := 2\mathbb{Z}^2$  und  $\mathcal{P}_2 := \mathbb{Z}^2 + (0.5, 0.5)$  als Quantisierern. Die Fehlerregionen seien entsprechend Beispiel 4.2.3, Teil 1., gewählt (siehe Abb. 4.13). Für  $p_1 := \{(1, 0), (2, 1), (1, 2), (0, 1)\}$ ,  $p_2 := p_1 - (2, 0)$ ,  $p_3 := p_1 - (0, 2)$  und  $p_4 := p_1 - (2, 2)$ , wobei die Subtraktion jeweils punktweise gemeint ist, sei  $K(p_i)$  die konvexe Hülle von  $p_i$  (Rauten in Abb. 4.13). Dann seien  $\mathcal{I}_{i,k}^1 := \mathcal{E}_{i,0}^1 := \mathcal{F}_{i,0}^1 - K(p_i)$  die Einbettungsregionen für  $V((0, 0))$  im  $2\mathbb{Z}^2$ -Gitter. Im  $\mathbb{Z}^2 + (0.5, 0.5)$ -Gitter seien  $\mathcal{I}_{i,k}^2 := \mathcal{E}_{i,k}^2 := \mathcal{F}_{i,k}^2 \cap K(p_i)$ . Diese Konstruktion erfüllt die Bedingung (4.30).

2. Gegeben sei wieder der Midrise-Quantisierer für die Schrittweite  $2^n$ ,  $\mathcal{P}_n := \frac{2^n-1}{2} + 2^n\mathbb{Z}$ . Für Fehlerregionen

$$\mathcal{F}_{1,k}^n := [k2^n : (k + \frac{1}{2})2^n - 1] \quad \text{und} \quad \mathcal{F}_{2,k}^n := [(k + \frac{1}{2})2^n : (k + 1)2^n - 1]$$

sei  $I := [K : N]$  mit  $K \geq 2$ . Für  $\mathcal{P}_n$ ,  $n \in I$ , sind

$$\mathcal{E}_{1,k}^n := [(k + \frac{1}{2})2^n - 2^{K-2} : (k + \frac{1}{2})2^n - 1] \quad \text{und} \quad \mathcal{E}_{2,k}^n := [(k + \frac{1}{2})2^n : (k + \frac{1}{2})2^n + 2^{K-2} - 1]$$

Einbettungsregionen. Dann erfüllt

$$\{\mathcal{E}_{i,k}^n | n \in I, k \in \mathbb{Z}, i \in \{1, 2\}\}$$

die Bedingung (4.30). Die Anzahl einbettbarer Vektoren ist hier konstant gleich  $2^{K-2}$ . Soll eine redundante Einbettung z.B. zwecks Fehlerkorrektur verwendet werden, ist  $K$  entsprechend groß zu wählen.

## 4.7.2 Erkennung einer Einbettung

Da Audiosignale mit eingebetteter Zusatzinformation dasselbe Format wie solche ohne diese Zusatzinformation haben, z.B. könnten beide aus  $[-2^N : 2^N - 1]^n$  stammen, ist auch im Transformationsbereich eine Unterscheidung schwierig. Daher ist es normalerweise notwendig, die Einbettung mit Hilfe einer speziellen Signatur, z.B. in Form spezieller Einbettungsvektoren, zu kennzeichnen. Diese sollten so gewählt werden, daß die Signatur mit hoher Wahrscheinlichkeit vom Decoder der eingebetteten Vektoren erkannt wird.

Unter der Annahme unabhängiger und gleichverteilter Zufallsvariablen für die Transformationswerte sorgt eine  $n$ -Bit Signatur für eine Fehlerwahrscheinlichkeit von  $2^{-n}$  pro Signaturtest. Da die Anzahl der Tests relativ groß werden kann, muß auch  $n$  entsprechend groß gewählt werden. Am Beispiel der 12 Transformationswerte umfassenden Blöcke bei MPEG und einer Einbettungskapazität von z.B. 5 Bits pro Sample sorgt eine 10 Bit Signatur für eine Fehlerwahrscheinlichkeit von  $1/1024$  bei einem Nutzdatenverlust von  $1/6$ . Da pro Datenrahmen im Falle von MPEG jedoch  $32 \cdot 3 \cdot 2 = 192$  solcher Blöcke vorkommen können, bei 48 kHz sind dies ca. 8000 Blöcke pro Sekunde, ist diese Fehlerwahrscheinlichkeit meist nicht akzeptabel. Natürlich trifft die Annahme der Gleichverteilung der Signalwerte nicht allgemein zu, sondern höchstens auf niederwertige Bits in der Signaldarstellung. Eine geschickte Wahl der höherwertigen Anteile der Einbettung entgegen Signalcharakteristiken kann hier eine Erniedrigung der Fehlerhäufigkeit erbringen. Ein weiterer wichtiger Punkt ist die Korrelation zwischen benachbarten Samples, welche insbesondere bei harmonischen Signalanteilen bzw. Subbändern ansgenutzt werden kann. Konkrete Beispiele hängen vom gegebenen Kontext ab und werden in unserem Fall in den Anwendungskapiteln behandelt.

### 4.7.3 Blockbildung und -organisation

Die Gruppierung benachbarter Teilbandwerte zu Einbettungsblöcken hat mehrere Vorteile:

- Eine einheitliche Einbettungsart im Nutzdatenteil des Blocks bzw. eine konstante Bitbreite wird ermöglicht, wenn auf alle Werte des Blocks dieselbe Quantisierungsart Anwendung findet.
- Eine Detektierung der Einbettung und des Einbettungstyps wird nur einmal pro Block nötig.
- Eine Paketbildung wird ermöglicht. Diese kann z.B. zum Zwecke der Fehlerkorrektur in Verbindung mit redundanter Übermittlung einzelner Blöcke oder Teilblöcke verwendet werden.
- Das Wechseln der Einbettungsart oder die Verwendung von Signaturen zur Markierung gültiger Daten ist nur in Verbindung mit Blöcken von Daten sinnvoll.

Zwei mögliche Organisationsformen der Blockbildung pro Datenrahmen sind

1. unabhängige Blöcke und
2. Trennung von Deskriptor- und Datenblöcken.

Im Fall unabhängiger Blöcke werden in jedem Block

- Signatur,
- Bitbreite bzw. Einbettungsart, und
- Nutzdaten

übertragen. Vorteile sind die Unabhängigkeit der Blöcke und die Lokalität von bei Blockbeschädigungen auftretenden Fehlern. Ein wichtiger Nachteil ist der große Overhead für Signatur und Bitbreiteninformation.

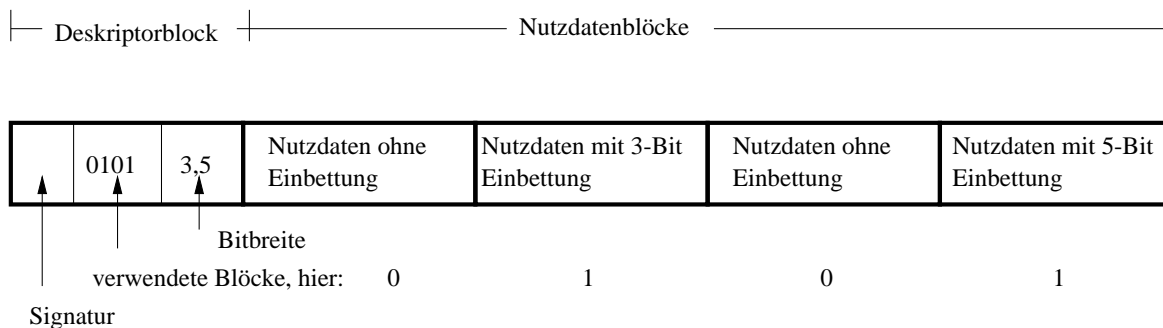


Abbildung 4.14: Aufteilung in Deskriptor- und Datenblöcke. Der Datenrahmen besteht aus fünf Blöcken mit konstanter Vektorenzahl. Der erste Block ist der Deskriptorblock.

Bei Trennung in Deskriptor- und Datenblöcke speichern einige wenige Deskriptorblöcke die Position und Bitbreiten aller Datenblöcke innerhalb des Datenrahmens. Abb. 4.14 zeigt ein Beispiel für solch eine Blockorganisation. In den Datenblöcken werden, bis auf einen etwaigen

Overhead zur Fehlerkorrektur, ausschließlich Nutzdaten gespeichert. Hierbei wird die Redundanz, wie sie bei unabhängigen Blöcken auftritt, reduziert, da die Signatur nur einmal gespeichert werden muß. Ein Nachteil dieser Blockorganisation ist die Anfälligkeit gegenüber Beschädigungen der Deskriptorblöcke, die den Verlust aller Daten eines Verarbeitungsrahmens zur Folge haben können.



## Kapitel 5

# Codecs zur Vermeidung von Generationseffekten

In diesem Kapitel werden Codecs zur Vermeidung von Generationseffekten bei psychoakustischen Transformations- oder Subbandcodierern entworfen. Die Darstellung ist hier möglichst allgemein gehalten, so daß die vorgestellten Konzepte auf eine möglichst große Klasse solcher Kompressionsverfahren angewendet werden können. Das Ziel der vorgestellten Verfahren ist der Erhalt der perceptuellen Qualität von Audiosignalen erster Generation über beliebige Generationen hinaus.

Zunächst wird ein generisches Codec-Modell vorgestellt. Hierbei werden optionale und verpflichtende Komponenten unterschieden. Der zweite Abschnitt untersucht die Beziehung der in Kapitel 4 vorgestellten Fehlerregionen zu Audiosignalen erster Generation. Im Hauptteil des Kapitels werden daraufhin zwei Codec-Varianten vorgestellt, die unter Verwendung der Techniken aus Kapitel 4 die angestrebte Zielsetzung in einem Fall ganz bzw. im anderen Fall approximativ erfüllen. Abschließend wird auf die Erweiterung des Codec-Modells auf Kompressionsverfahren mit mehrstufigen Transformationen eingegangen. Eine konkrete Realisierung der Konzepte dieses Kapitels wird in Kapitel 6 vorgestellt.

### 5.1 Das Codec-Modell

Eine relativ allgemeine Form eines psychoakustischen Audiocodecs ist in Abb. 5.1 dargestellt. Es wird anhand der einzelnen Bauteile kurz diskutiert, welche Teile eines solchen Codecs für die nachfolgenden Betrachtungen von Bedeutung sind. Anhand des MPEG-Verfahrens [18] werden mögliche Realisierungen der Module beschrieben.

**Transformation:** Das Inputsignal wird block- oder rahmenweise in eine Subband- oder Spektralkomponentendarstellung transformiert. Dies wird durch eine meist lineare, reell- oder komplexwertige, und invertierbare Abbildung geleistet. Beispiele sind Multiratenfilterbänke und cosinusmodulierte Transformationen (*Modulated Lapped Transforms, MLTs*)[6]. Im allgemeinen sind dies Abbildungen auf  $\ell^2(\mathbb{Z})$ , daher sind die den einzelnen Signalblöcken zugeordneten Einschränkungen der Abbildungen i.a. nicht „lokal“ invertierbar, sondern nur unter Einbeziehung von Daten benachbarter Signalblöcke. Um die im nächsten Abschnitt erfolgenden theoretischen Überlegungen nicht unnötig zu erschweren, soll in diesem Abschnitt angenommen werden, daß die Einschränkungen  $T$

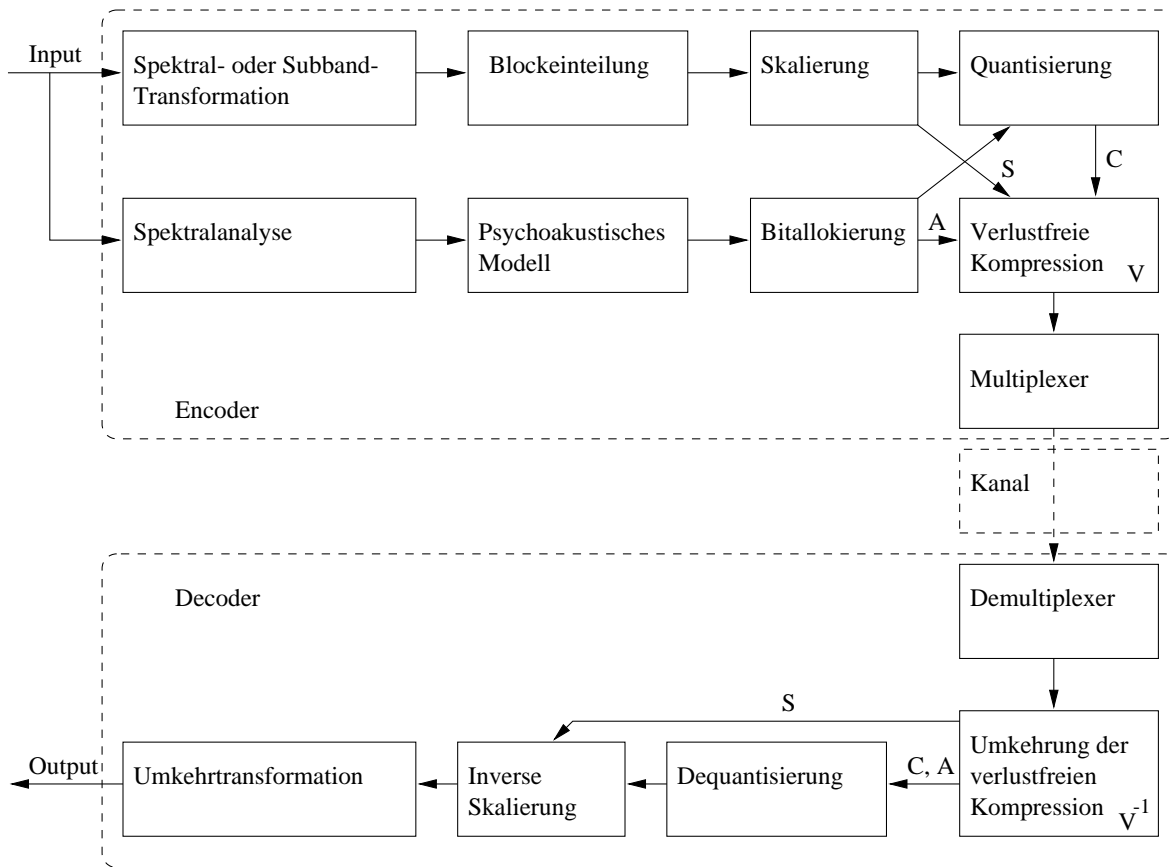


Abbildung 5.1: Allgemeine Form eines psychoakustischen Audiocodexs.

der Transformationen auf einen Signalblock der Länge  $n$  aus  $GL(n, \mathbb{R})$  sind. Aufgrund der theoretischen Natur der Ergebnisse dieses Abschnitts, ist dies ohne Probleme zu rechtfertigen.

Die Umkehrtransformation entspricht der Inversen zu  $T$ . Wie in Kapitel 4 erläutert, wird die Inverse in der Praxis durch eine „numerische“ oder approximative Umkehrtransformation  $\tilde{T}$  ersetzt.

Bei MPEG Audiocodern nach [18] ist die Transformation eine 32-Band Multiratenfilterung. In einer Ausführung (Layer III) wird diese Filterbank mit einer nachfolgenden gefensterten Cosinustransformation kaskadiert. Die Blocklänge beträgt in den Layern II und III jeweils 1152 Samples.

**Spektralanalyse:** Als Vorverarbeitung für die psychoakustische Analyse wird eine Spektraltransformation vorgenommen. Dies kann mit Hilfe trigonometrischer Transformationen geschehen (meist gefensterte Fourier- oder Cosinustransformation). Wichtig im Zusammenhang mit den zu entwickelnden Verfahren ist lediglich, daß diese Analyse *synchron* zur obigen Subbandtransformation vorgenommen wird. Dies ist jedoch i.a. gegeben. Die psychoakustische Analyse ist außerdem naturgemäß *lokal*, d.h. es werden nur Informationen aus wenigen benachbarten Datenrahmen in die Berechnung des psychoakustischen Modells, und somit in die Spektralanalyse, einbezogen.

Im Falle von MPEG Layer II werden sogar lediglich Werte des aktuellen Datenrahmens für die Spektralanalyse verwendet.

**Blockeinteilung:** Als Vorverarbeitung ist eine feste oder variable Blockeinteilung der Subbandsignale möglich. Im Falle von MPEG Audio werden Blöcke von 36 Subbandsamples in drei Blöcke von jeweils 12 Samples aufgeteilt ( $3 \cdot 12 \cdot 32 = 1152$ ). Somit wird die Bestimmung der Skalierungsfaktoren dynamischer.

Die Blockeinteilung ist für unser Verfahren interessant, da hieraus eine natürliche Einteilung in Einbettungsblöcke bestimmt werden kann.

**Skalierung:** Die in Kapitel 4 beschriebene redundanzreduzierende Skalierung wird blockweise auf den Subbandsignalen vorgenommen. Die Skalierungsfaktoren und die Information über die Art der Skalierung oder die Verteilung auf die einzelnen Blöcke  $S$  wird an den Decoder übertragen.

Im Falle von MPEG Layer II können 12er-Blöcke mit ähnlichen Skalierungsfaktoren zwecks weiterer Speicherplatzersparnis zusammengefaßt werden.

**Psychoakustisches Modell:** Das psychoakustische Modell  $\Psi$  liefert im allgemeinen den SMR eines Signalblocks  $x$ . Dieser kann sofort in einen maximal erlaubten Quantisierungsfehler umgerechnet werden. Hier soll zunächst nicht weiter auf Einzelheiten des psychoakustischen Modells eingegangen werden. Wichtig ist lediglich die Bestimmung des maximalen Quantisierungsfehlers und damit die Bestimmung möglicher Quantisierer.

Die im MPEG-Standard [18] vorgeschlagenen psychoakustischen Modelle liefern beide den SMR eines gegebenen Datenrahmens.

**Bitallokierung:** In Abhängigkeit der globalen Coderparameter  $P$ , des psychoakustischen Modells  $\Psi(x)$  für den Signalblock  $x$  und gegebenenfalls des Subbandsignals  $s = Tx$  sowie  $x$  selbst wird eine Aufteilung der zur Verfügung stehenden Bitkapazität für die einzelnen Subbänder und Skalierungsblöcke sowie eine geeignete Quantisierungsauflösung und -art bestimmt. Hierfür gibt es sehr viele verschiedene Methoden. Meist sind dies iterative Algorithmen, die die Audioqualität unter der Nebenbedingung der Einhaltung einer gewissen maximal erlaubten Bitrate pro Datenrahmen maximieren.

Die Bitallokierung wird wie in Kapitel 4 durch eine Abbildung

$$b : (\Psi(x), P, x, s) \mapsto (p_1, p_2, \dots)$$

beschrieben. Mit  $E := E(x) := (p_1, p_2, \dots)$  seien alle Encoderparameter des aktuellen Datenrahmens, d.h. alle zur Komprimierung des transformierten Signals  $s$  notwendigen Informationen bezeichnet. Die zur Decodierung benötigten Informationen  $D = D(x)$  können aus  $E$  bestimmt werden,  $E \Rightarrow D$ . Bezüglich des Informationsgehalts stimmen  $E$  und  $D$  de facto meist überein. Da jedoch die Darstellung oder Repräsentation der Daten in  $D$  oder  $E$  häufig sehr unterschiedlich ist, soll weiterhin zwischen beiden unterschieden werden.

**Quantisierung:** Aufgrund der Bitallokierung wird eine Quantisierung der u.U. skalierten Subbandsamples vorgenommen. Hierbei wird für bestimmte Blöcke von Subbandsignalen jeweils ein durch die verfügbare Bitrate bestimmter Quantisierer verwendet.

Im Falle von MPEG-1, Layer II, wird in jedem Datenrahmen für jedes Subband ein linearer Skalarquantisierer bestimmt. Gruppen von 36 Subbandsamples besitzen somit einen gemeinsamen Quantisierer. Da die Bitallokierung i.a. ab einer gewissen Grenze keinem darüberliegenden Subband Bits zur Verfügung stellt, müssen hierbei nicht immer alle Subbänder beachtet werden.

**Verlustfreie Kompression:** Dieser Schritt der zusätzlichen Redundanzverringern wird in einigen Codern als abschließende Operation vorgenommen. Hierbei gehen keine Informationen verloren. Gängige Methoden, die in der Kompressionsrate die Entropie des Signals gut approximieren, sind Huffman- und Arithmetic-Coding.

In Layer III des MPEG Standards wird Huffman-Codierung mit adaptiver Codebuchauswahl verwendet. Hierzu stehen 30 Codebücher zur Verfügung.

**Multiplexer:** Der Multiplexer bettet die Codeworte und sämtliche Seiteninformationen geeignet in den entsprechenden Transportdatenstrom ein. Dieses Modul kann für die weiteren Überlegungen vernachlässigt werden.

Einige dieser Module sind für das prinzipielle Funktionieren eines psychoakustischen Transformationscodierers nicht zwingend erforderlich, in der Praxis jedoch meist gebräuchlich:

- Die verlustfreie Kompression bzw. Dekompression wird manchmal weggelassen um die Komplexität des Encoders zu senken. Dabei spielt meist der Rechenaufwand auf langsamer Hardware eine entscheidende Rolle. Im Falle von MPEG-1 wird die Huffman-Codierung nur in Layer III, nicht aber in den Layern I und II, vorgenommen.
- Die Skalierung ist ebenfalls optional, senkt aber erfahrungsgemäß die Bitrate erheblich. Dies liegt auch in der Natur von Audio- und Musiksignalen begründet, wo oft Segmente gleicher Energie bzw. Amplitudbereiche auftreten. Allerdings hat die Skalierung, wie in Kapitel 4 beschrieben, Auswirkungen auf die Kapazität der Biteinbettung und ist daher im weiteren zu beachten.
- Die Subbandtransformation kann auch durch eine Spektraltransformation ersetzt werden. Dies entspricht einer genaueren Frequenzauflösung und fand schon bei den ersten vorgeschlagenen psychoakustischen Transformationscodern Anwendung [28]. In diesem Fall bilden diese Transformation und die Spektralanalyse *eine* Transformation.

Wichtige globale Eigenschaften der hier betrachteten Klasse von Codern sind:

- Rahmen- oder blockweise Verarbeitung der digitalen Signale.
- Verwendung von Quantisierern  $\mathcal{P} = (P_1, P_2, \dots)$  wie in Kapitel 4 beschrieben. Hier handelt es sich meist um Skalarquantisierer.

## 5.2 Theoretische Überlegungen

Gegeben seien eine blockorientierte Subbandtransformation  $T \in \text{GL}(n, \mathbb{R})$  sowie ein Signalblock  $x \in \mathbb{R}^n$  mit zugehörigem Subbandblock  $s := Tx$ . O.B.d.A. wird angenommen, daß

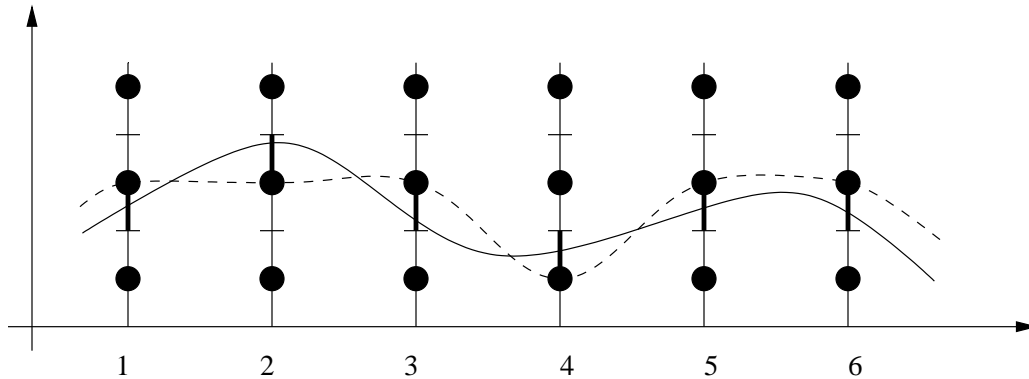


Abbildung 5.2: Signal (durchgezogene Kurve) mit diskreten Abtastwerten. Auf den vertikalen Linien durch die Abtastwerte sind die Voronoi-Zellen mit Zentroiden eingezeichnet. Die gestrichelte Linie deutet ein aus der Rückquantisierung gewonnenes Signal an. Exemplarische Fehlerregionen sind durch dicke Linien hervorgehoben.

innerhalb eines Audiocoders ein einziger Skalarquantisierer  $\mathcal{P} = (P_1, P_2, \dots)$  von der Bitallokieringroutine für den Subbandblock  $s$  ausgewählt wird. Bei mehreren verschiedenen Quantisierern wird lediglich die Notation komplexer. Die Quantisierungsfunktion sei wie üblich mit  $Q$  bzw.  $\tilde{Q}$  bezeichnet. Zur Erleichterung der Notation seien diese Funktionen durch punktweise Anwendung auf Vektoren in  $\mathbb{R}^n$  erweitert.

Abb. 5.2 zeigt ein Signal (durchgezogene Kurve) mit sechs diskreten Abtastpunkten. Auf den vertikalen Linien durch die Abtastpunkte sind die Voronoi-Zellen mit Zentroiden eingezeichnet. Die gestrichelte Linie deutet ein aus der Rückquantisierung gewonnenes Signal an.

Der Encoder erzeugt den Code  $(E, Qs)$ . Weiterhin wird angenommen, daß der Decoder die Decodierungsinformation  $D = D(x)$  sowie die Rekonstruktion des Subbandblocks  $\tilde{Q}Q(s) =: \tilde{s}$  erzeugt und daß der Coder derart gestaltet ist, daß aus der Decodierungsinformation auf die Codierungsinformation  $E = E(x)$  geschlossen werden kann,  $D \Rightarrow E$ . Überträgt der Decoder sowohl  $T^{-1}\tilde{s}$  als auch  $D(s)$ , so kann ein Encoder bei erneuter Codierung hieraus wegen  $TT^{-1}\tilde{s} = \tilde{s}$  und  $Qs = Q\tilde{Q}Qs = Q\tilde{s}$  exakt denselben Code  $(E, Qs)$  reproduzieren. Die Information, daß bei erneuter Encodierung gerade  $Q$  zur Quantisierung verwendet werden muß, ist in  $D$  enthalten.

Eine zusätzliche Übertragung von  $D(x)$  ist natürlich i.a. nicht wünschenswert. Dies kann zwar, falls mit den entsprechenden Daten nur lokal gearbeitet wird, sinnvoll sein, jedoch spricht für den allgemeinen Fall einiges dagegen:

- Werden Signale rechnerbasiert in einzelnen Dateien gespeichert, so muß für die Seiteninformationen  $D(x)$  ein neuer Dateityp bzw. ein neues Datenformat vorgesehen werden.
- Zu einem Audiosignal müssen dann, im Sinne der Verwendbarkeit auf Standardmedien, mindestens jeweils zwei Dateien existieren.
- Die Seiteninformationen sind nicht über Standardmedien bzw. innerhalb von Standard-audioformaten übertragbar (z.B. CD- oder DAT-basierte Audiospuren).
- Der Speicheraufwand steigt.

### Naiver Analysis-by-Synthesis Algorithmus

**Input:** Rekonstruiertes Subbandsignal  $\tilde{s}$ , Decodierparameter  $D(x)$  zum Signalblock  $x$ , globale Parameter  $P$ .

**Calculate**  $E(x)$  from  $D(x)$

**Repeat**

**Choose** next  $s' \in v(\tilde{s}, Q)$

**Calculate**  $b(\Psi(T^{-1}s'), P, T^{-1}s', s') =: E(T^{-1}s')$

**Until**  $E(T^{-1}s') = E(x)$ .

**Output:** Rekonstruktion  $T^{-1}s'$ .

Abbildung 5.3: Naiver Analysis-by-Synthesis Algorithmus zur Bestimmung eines Rekonstruktionsvektors  $s'$  zum Signalblock  $x$  mit der Eigenschaft  $E(T^{-1}s') = E(x)$ .

Wir zeigen nun, daß  $D(x)$  zur Rekonstruktion von  $(E, Qs)$  theoretisch gar nicht übertragen werden muß, falls das Einbettungsverfahren aus Kapitel 4 auf bestimmte Art und Weise verwendet wird. Zu  $s = (s_1, \dots, s_n) \in \mathbb{R}^n$  sei  $v(s, Q) := V(\tilde{Q}Qs_1) \times \dots \times V(\tilde{Q}Qs_n)$  die „Multi“-Voronoi-Zelle. Ein Decoder könnte nun innerhalb eines Analysis-by-Synthesis Verfahrens, wie in Abb. 5.3 angegeben,  $v(\tilde{s}, Q)$  erschöpfend durchsuchen. Da  $s \in v(s, Q) = v(\tilde{s}, Q)$ , terminiert der Algorithmus (bei sinnvollem, z.B. abzählbarem oder endlichem,  $v(\tilde{s}, Q)$ ) mit einem  $s' \in v(s, Q)$ . Da in diesem Fall  $(E(x), Qs) = (E(T^{-1}s'), Qs')$ , ist das Ziel erreicht. Der entscheidende Nachteil ist jedoch, daß in diesem Fall keine Aussage über die perzeptuelle Qualität  $T^{-1}s'$  möglich ist, da  $s' \in v(s, Q)$  so liegen kann, daß  $|s - s'|$  größer als der maximale Quantisierungsfehler ist.

Mit Hilfe der Fehlerregionen kann dieser Mißstand ausgeräumt werden. Abb. 5.2 zeigt die für das dargestellte Signal anfallenden Fehlerregionen für Fehlerregionenverbünde, wie sie in Kapitel 4 beschrieben wurden, als dicke hervorgehobene Linien. Es seien Fehler- und Einbettungsregionenverbünde mit  $\forall i, k : \mathcal{E}_{i,k} = \mathcal{F}_{i,k}$ , also  $\mathcal{F}_{0,k} = \emptyset$  stets für alle Voronoi-Zellen gegeben. Dann gilt  $\cup_{i,k} \mathcal{F}_{i,k} = \cup_k V(P_k)$ . Für  $s \in \mathbb{R}^n$  sei  $Q$  die Fortsetzung von (4.6) auf Vektoren:  $Q : s \mapsto (i, k)$  wo  $i, k \in \mathbb{N}^n$  und  $Q(s_j) = (i_j, k_j)$  mit  $Q$  wie in (4.6) definiert. Dann wird durch  $\mathcal{E}(s) := \mathcal{E}_{i,k} := \mathcal{E}_{i_1, k_1} \times \dots \times \mathcal{E}_{i_n, k_n}$  die Einbettungsregion des Vektors  $s \in \mathbb{R}^n$  definiert. Analog definiert  $\mathcal{F}(s)$  die Fehlerregion von  $s$ .

Abb. 5.4 zeigt eine Variante des naiven Algorithmus, bei der der Suchraum auf  $\mathcal{E}(s) \subset v(s, Q)$  eingeschränkt ist.

**Satz 5.2.1** *Der Algorithmus in Abb. 5.4 terminiert mit einem Vektor  $s' \in \mathbb{R}^n$ , so daß für  $1 \leq j < n$  für den Rekonstruktionsfehler  $|s_j - s'_j| < R_k$  und weiterhin  $(E(x), Qs) = (E(T^{-1}s'), Qs')$  gilt.*

### Analysis-by-Synthesis Algorithmus mit Einbettungsverfahren

**Input:** Rekonstruiertes Subbandsignal  $\tilde{s}$ , Decodierparameter  $D(x)$  zum Signalblock  $x$ , globale Parameter  $P$ , sowie Einbettungsparameter  $i, k \in \mathbb{R}^n$  mit  $QTx = (i, k)$ .

**Calculate**  $E(x)$  from  $D(x)$

**Repeat**

**Choose** next  $s' \in \mathcal{E}(s) = \mathcal{E}_{i,k}$

**Calculate**  $b(\Psi(T^{-1}s'), P, T^{-1}s', s') =: E(T^{-1}s')$

**Until**  $E(T^{-1}s') = E(x)$ .

**Output:** Rekonstruktion  $T^{-1}s'$ .

Abbildung 5.4: Analysis-by-Synthesis Algorithmus zur Bestimmung eines Rekonstruktionsvektors  $s'$  zum Signalblock  $x$  mit der Eigenschaft  $E(T^{-1}s') = E(x)$  unter Verwendung der Einbettungsmethode.

*Beweis.* Der Algorithmus beschreibt einen Durchlauf durch möglicherweise alle Einbettungsvektoren. Da  $\forall r, \ell : \mathcal{E}_{r,\ell} = \mathcal{F}_{r,\ell}$  nach Voraussetzung, gilt  $s \in \mathcal{E}_{i,k}$  und somit gibt es mindestens ein  $s' \in \mathcal{E}_{i,k}$ , so daß die Abbruchbedingung erfüllt ist. Weiterhin gilt  $\forall s' = (s'_1, \dots, s'_n) \in \mathcal{E}_{i,k}$ ,  $s'_j \in \mathcal{F}_{i_j, k_j} = \mathcal{E}_{i_j, k_j}$  und daher  $|s'_j - s_j| < R_k$ .  $\square$

Die Einbettungsregionen beschreiben somit den gültigen Teilbereich  $\mathcal{E}(s) \subset v(s, Q)$  für die Suche nach den im Sinne des Algorithmus optimalen Rekonstruktionssignalen. Für solch ein Signal  $s'$  muß zwar nicht  $T^{-1}s' = x$  gelten, jedoch liefert es bei einer erneuten Encodierung denselben Code wie  $x$  und überschreitet weiterhin nicht den maximalen Rekonstruktionsfehler. Bestimmt dieser Rekonstruktionsfehler die „perzeptuell gültigen“ Rücktransformationen, so ist  $T^{-1}s'$  perzeptuell nicht von  $x$  unterscheidbar.

Leider ist das obige Verfahren in dieser Form nicht praktikabel, da eine erschöpfende Suche in  $\mathcal{E}(s)$  selbst bei in der Praxis auftretenden endlichen Wertebereichen die exponentielle Laufzeit  $O(m^n)$  besitzt. Hier ist  $m$  die Größe der Einbettungsregionen. Die Entwicklung eines schnellen iterativen Algorithmus zur Bestimmung eines geeigneten  $s'$  ist ein weiterführendes Problem, das jedoch, bei fehlender Korrelation zwischen benachbarten Einbettungsvektoren, nicht einfach zu lösen sein dürfte.

Eine interessante Frage im Zusammenhang mit dem obigen Algorithmus ist die Kaskadierbarkeit: kann das Vorgehen wiederholt und somit eine beliebig hohe Codegeneration gleicher Qualität erzeugt werden? Hierzu reicht es zu zeigen, daß im zweiten ( $n$ -ten) Encoder derselbe Code zum Decoder übertragen wird, wie im ersten ( $n-1$ -ten). Für die Codeworte  $Qs$  und die Seiteninformation  $E(x)$  ist dies offensichtlich der Fall. Zu betrachten bleiben somit die

Fehlerregioneninformationen  $i \in \mathbb{R}^n$ . Da die  $\mathcal{F}_{r,\ell}$  aber disjunkt und weiterhin sogar identisch den  $\mathcal{E}_{r,\ell}$  sind, gilt  $\mathcal{E}(s) = \mathcal{E}(s')$  und somit  $\mathcal{F}(s') = \mathcal{F}(s)$ .

### 5.3 Codec ohne Generationseffekte

In diesem Abschnitt werden Audiocodecs zur Vermeidung von Generationseffekten vorgestellt und deren Funktionsweise besprochen. Ein grundlegendes Hilfsmittel hierbei ist die in Kapitel 4 vorgestellte Einbettung. Je nach verwendeter Einbettungsart gibt es zwei prinzipiell verschiedene Codec-Varianten:

1. Codec, bei dem die direkte Biteinbettung angewendet wird. Bei dieser Codec-Art wird keine Zusatzinformation zur Speicherung der Fehlerregionen zum komprimierten Datenstrom hinzugefügt, die Bitrate somit nicht erhöht. Der Nachteil dieser Methode ist ein möglicher Qualitätsverlust des Audiosignals.
2. Codec, bei dem die Einbettungsmethode mit Fehler- und Einbettungsregionen Verwendung findet. Hier wird die Fehlerregionsinformation im komprimierten Datenstrom vom Encoder zum Decoder übertragen und die Bitrate steigt etwas an. Der Vorteil dieser Methode ist, daß sie unter gewissen Bedingungen den Erhalt der perzeptuellen Qualität der ersten Codegeneration gewährleistet.

Alle weiteren Entwicklungen dieses Abschnitts beruhen auf folgender Überlegung: Wurde ein Audiosignal bereits einmal verlustbehaftet komprimiert und wieder dekomprimiert, so sind, ohne weitere Kenntnis des Kompressionsverfahrens, einige Informationen über das Signal unwiederbringlich verlorengegangen. Eine andere Sichtweise ist, daß solch ein Signal unter Umständen nicht von einem ähnlichen Signal, welches aufgrund anderer Umstände vom Original abweicht, unterscheidbar ist. Daher zielen alle Anstrengungen dieses Abschnitts darauf ab, die zur Erstkompression verwendeten Parameter an alle folgenden Codecstufen weiterzugeben.

#### 5.3.1 Grundaufbau

Abb. 5.5 zeigt den groben Aufbau aller hier diskutierten Codecs. Dieser Aufbau ist vom verwendeten Einbettungsverfahren unabhängig. Es wurde zunächst vereinfachend auf Verfeinerungen wie Skalierung oder zusätzliche Entropiecodierung verzichtet. Die dem Gesamtverfahren zugrundeliegende Idee ist die folgende:

*In psychoakustische Transformationscodecs läßt sich hervorragend die in Kapitel 4 vorgestellte Dateneinbettung integrieren. Der Decoder bettet mit Hilfe dieser Verfahren die zur ersten Kompression verwendeten Encoderparameter in das Audiosignal erster Generation ein. Auf diese Art und Weise werden die Parameter von Codecstufe zu Codecstufe übermittelt und können bei einer erneuten Kompression wiederverwendet werden.*

Zur Beschreibung des Verfahrens konzentrieren wir uns zunächst auf den *Decoder*, Abb. 5.5 unten. Der Decoder erhält die komprimierten Daten eines Datenblocks, die grob aus Codeworten und Seiteninformation  $D$  bestehen. Mit Hilfe der Seiteninformation können aus den Codeworten durch Dequantisierung die rekonstruierten Subbandsamples gewonnen werden. Eine wichtige Eigenschaft der Seiteninformation ist, daß mit ihrer Hilfe meist nicht nur die Dequantisierung, sondern auch encoderseitig die Quantisierung möglich ist (Bedingung  $D \Rightarrow E$ ,

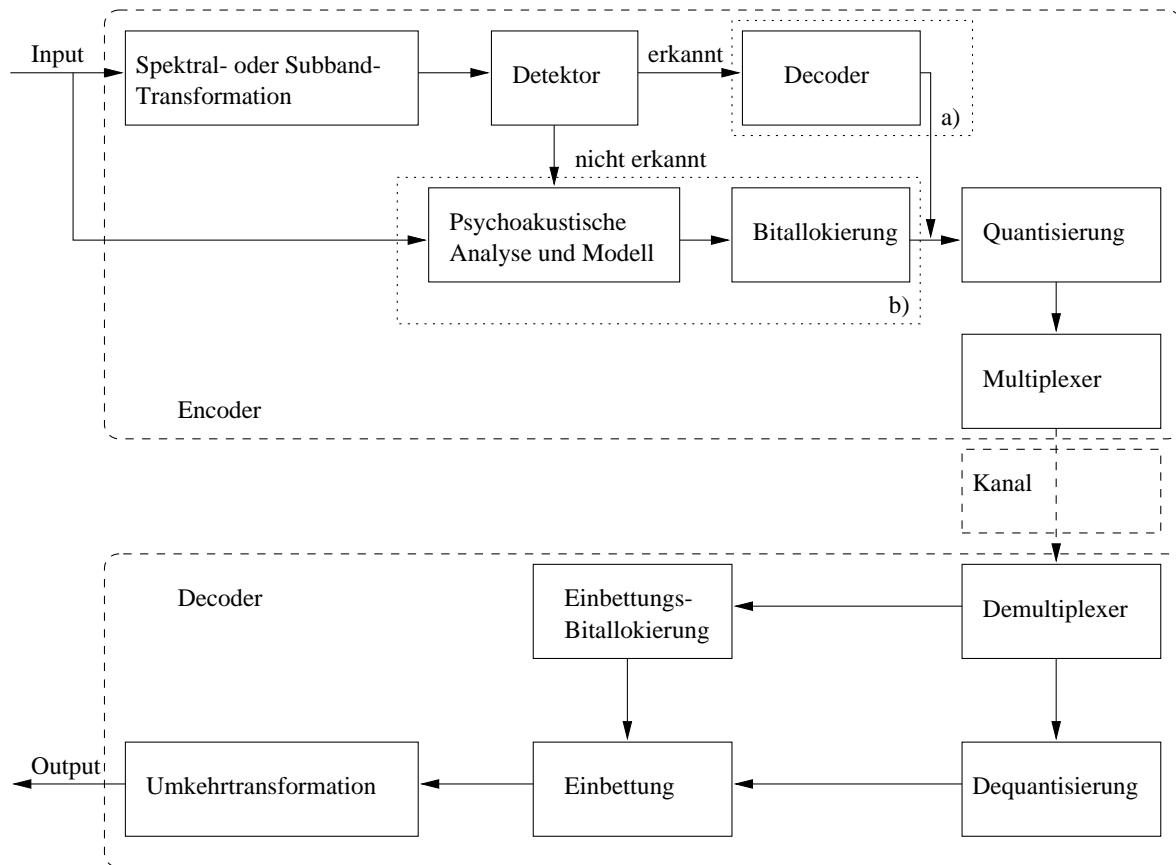


Abbildung 5.5: Encoder und Decoder zur Vermeidung von Generationseffekten.

s.o.). Dies soll im folgenden vorausgesetzt werden. Nachfolgend an die Dequantisierung wird die Seiteninformation  $D$  mit einem der Einbettungsverfahren aus Kapitel 4 entsprechend der durch  $D$  selbst angegebenen Quantisierungsart und -auflösung in die Subbandsamples des Blocks eingebettet. Abschließend wird der Block zurücktransformiert.

Der *Encoder* zeichnet sich durch eine hybride Funktionsweise aus. Für einen Block wird zunächst die Spektral- oder Subbandtransformation durchgeführt. Im anschließenden Detektor wird geprüft, ob in dem Datenblock eingebettete Informationen enthalten sind. Ist dies nicht der Fall, oder sind die eingebetteten Daten beschädigt bzw. unbrauchbar, so wird der Datenrahmen mit dem normalen Kompressionsalgorithmus weiterverarbeitet. Hierzu wird die Bitallokierung mit Hilfe des psychoakustischen Modells berechnet, siehe Block *b*) in Abb. 5.5. Konnten eingebettete Daten detektiert werden, werden diese von einem weiteren Modul decodiert, siehe Block *a*) in Abb. 5.5. Dies liefert die im Decoderteil des Codecs eingebettete Seiteninformation  $D$ . Aus  $D$  werden nun die Encoderparameter  $E$  und insbesondere die Bitallokierung, mit deren Hilfe daraufhin die Quantisierung vorgenommen wird, gewonnen. Der Multiplexer schließt den Codiervorgang wie gehabt ab.

### 5.3.2 Direkteinbettungscodec

In Abb. 5.6 ist der Algorithmus eines Encoders zur Verwendung in Zusammenhang mit der Direkteinbettung angegeben. Beschrieben ist wie üblich die Verarbeitung *eines* Datenrahmens,

**Encoder-Algorithmus** (zur Verwendung mit Direkteinbettung)

**Input:** Signalblock  $x \in \mathbb{R}^n$ , Subbandtransformation  $T$ , Coderparameter  $P$ , Psychoakustisches Modell  $\Psi$ , Bitallokierungsfunktion  $b$ , Detektorfunktion  $\mathcal{D}$  auf  $\mathbb{R}^n$ , wo  $\mathcal{M} \subset \text{Im}(\mathcal{D})$  die Menge aller gültigen Markierungen ist.

1. Berechne  $s := Tx$ .
2. Detektiere  $\mathcal{D}(s) =: M$ .
3. Falls  $M \notin \mathcal{M}$ 
  - (a) Berechne  $\Psi(x)$ . Setze  $s' := s$ .
  - (b) Berechne  $b(\Psi(x), P, s, x)$  und daraus Encoderparameter  $E = E(x)$ .
4. Sonst
  - (a) Decodiere  $s \mapsto \tilde{Q}(s) =: (s', D)$ .
  - (b) Berechne Encoderparameter  $D \mapsto E$ .
5. Quantisiere  $s'$  mit Hilfe von  $E$ :  $s' \mapsto q(s')$ .
6. Bestimme Decoderparameter  $E \mapsto D$ .

**Output:** Codeworte  $q(s')$ , Decoderparameter  $D$ .

Abbildung 5.6: Encoder-Algorithmus zur Verwendung mit der Direkteinbettung.

repräsentiert durch ein Signal  $x \in \mathbb{R}^n$ . In der Praxis wird hier anstatt  $\mathbb{R}$  für ein geeignetes  $k \in \mathbb{N}$  die Menge  $\mathbb{B}_k$  aller  $k$ -stelligen Binärzahlen, mit einer für Digitalrechner geeigneten Fließ- oder Festkommainterpretation und entsprechender Arithmetik ausgestattet, gewählt. Dies kann z.B. eine Zweierkomplementdarstellung mit Addition und Multiplikation modulo  $2^k$  sein. Dabei wird häufig eine Konvertierung zwischen Festkommadarstellung (zur Speicherung der digitalen Audiodaten, z.B. in 16-Bit Darstellung) und Fließkommadarstellung (für die Berechnungen innerhalb der Kompressionsverfahren, z.B. mit 32-Bit Genauigkeit) vorgenommen. Die in den Algorithmen angegebenen Einbettungsfunktionen können als auf der Binärdarstellung von Werten aus  $\mathbb{R}$  arbeitend angenommen werden.

Die *Subbandtransformation* wird vereinfachend als Blocktransformation angenommen. Somit ist der Subbandsignalblock als  $s = Tx$  aus  $\mathbb{R}^n$  gegeben. Oft wird jedoch eine überlappende blockweise Verarbeitungsform gewählt, wo innerhalb der Verarbeitung eines Datenrahmens noch auf die Werte von vorhergehenden Datenrahmen zurückgegriffen wird. Ein Beispiel hierfür sind Multiratenfilterbänke. Für eine überlappende Verarbeitung sind jedoch keine konzeptionellen Veränderungen des Verfahrens nötig. Konzepte für die überlappende Signal-

**Decoder-Algorithmus** (zur Verwendung mit Direkteinbettung)

**Input:** Codeworte  $q(s)$ , Decoderparameter  $D$ ,  
Einbettungsfunktion  $E_1$

1. Rückquantisiere  $q(s) \mapsto \tilde{s} =: (\tilde{s}_1, \dots, \tilde{s}_n)$ .
2. Bestimme Einbettungsstellen und -breiten

$$(\tilde{s}, D) \mapsto (i, t) =: ((i_1, \dots, i_m), (t_1, \dots, t_m)),$$

sowie eine geeignete Detektormarkierung  $M$ , so daß  $1 \leq i_k < i_{k+1} \leq n$  und eine geeignete Binärstringdarstellung  $(D, M) \mapsto \text{bin}(D, M)$ , die aus  $|\text{bin}(D, M)| = \sum_{j=1}^m t_j$  Bits besteht.

Hierbei gibt  $t_j$  die Einbettungsbreite in Bits für die  $j$ -te Stelle an.

3. Partitioniere  $\text{bin}(D, M) =: e_1 \cdots e_m$ , wobei  $e_j \in \{0, 1\}^{t_j}$ .
4. Bette für  $1 \leq j \leq m$  mit  $t_j$ -Bit Einbettung ein:

$$E_1 : (\tilde{s}_{i_j}, e_j) \mapsto \tilde{\tilde{s}}_{i_j}.$$

5. Setze  $\tilde{\tilde{s}}_k := \tilde{s}_k$  für alle  $k$ , wo  $k \neq i_j$  für  $1 \leq j < m$ .
6. Rücktransformiere  $\tilde{\tilde{s}} \mapsto T^{-1}\tilde{\tilde{s}}$ .

**Output:** Signalblock  $T^{-1}\tilde{\tilde{s}}$ .

Abbildung 5.7: Decoder-Algorithmus zur Verwendung mit der Direkteinbettung.

verarbeitung finden sich in [31].

Die *Detektorfunktion* arbeitet auf dem Subbandsignalblock  $s$  mit der Aufgabe, vorhandene Einbettungen wie in Abschnitt 4.7 beschrieben zu erkennen. Da es sinnvoll ist, zur Unterscheidung verschiedener Einbettungsbitbreiten und -blockstrukturen unterschiedliche Markierungen zu verwenden, wird eine Menge  $\mathcal{M}$  gültiger Markierungen verwendet. Zur Vereinfachung der Darstellung wurde hier nur von einem markierten Datenblock ausgegangen. Eine Aufteilung in Blöcke mit Größen  $k < n$ , vorzugsweise mit  $k|n$ , ist i.a. sinnvoller (siehe 4.7).

Wird eine gültige Markierung detektiert,  $D(s) = M \in \mathcal{M}$ , so wird in Schritt 4 die eingebettete Information extrahiert. Hierbei wurde idealisierend davon ausgegangen, daß die Subbandsamples keinen Verfälschungen, wie Arithmetikfehlern usw., unterliegen. Zur Erkennung bzw. Korrektur solcher Fehler sind Mechanismen wie in Abschnitt 4.6 besprochen auf das Subbandsignal  $s$  oder auf die extrahierten Daten anzuwenden. Die Gültigkeit der extrahierten Daten ist dann, falls keine Verfälschungen aufgetreten sind, mit einer gewissen (hohen)

Wahrscheinlichkeit gegeben.

Aus der extrahierten Information können somit die Decoderparameter des vorliegenden Datenrahmens und somit die Encoderparameter  $E = E(\tilde{x})$  bestimmt werden. Hierbei ist  $\tilde{x}$  der Datenblock, für den die Encoderparameter ursprünglich bestimmt wurden. Weiterhin werden die eingebetteten Bits aus allen Subbandwerten in  $s$  entfernt und es entstehen bereinigte Subbandwerte  $s'$ . Im Falle der Direkteinbettung ist es möglich, daß  $s' \neq T\tilde{x}$  gilt. Dies entspricht dem Effekt der möglichen Erhöhung des maximalen Rekonstruktionsfehlers bei dieser Methode.

Wird in Schritt 2 keine gültige Markierung gefunden, so wird angenommen, daß das Signal nicht mit einem Decoder laut Algorithmus 5.7 decodiert worden ist. In Schritt 3 werden daher auf konventionelle Art und Weise mit Hilfe einer psychoakustischen Analyse die Encoderparameter  $E(x)$  bestimmt.

In Schritt 5 wird mit Hilfe der Encoderparameter die Quantisierung vorgenommen. Vorgesaltet können dabei redundanzverringende Verarbeitungsschritte wie Skalierungen ausgeführt werden. Diese Verarbeitungsschritte, ebenso wie eine zusätzliche nachfolgende Entropiecodierung, sind hier zur Vereinfachung weggelassen worden.

Schließlich werden in Schritt 6 aus  $E$  die zur Decodierung nötigen Parameter  $D$  bestimmt und diese zusammen mit den Codeworten übertragen bzw. in einen Bitdatenstrom gespeichert.

Abb. 5.7 zeigt den Algorithmus eines entsprechenden Decoders. Aus dem Bitstrom werden die zur Decodierung benötigten Codeworte  $q(s)$  und Decoderparameter  $D$  für einen Datenrahmen  $s$  gelesen.

In Schritt 1 wird mit Hilfe von  $D$  die Rückquantisierung des Subbandblocks durchgeführt. Der komplexeste Schritt ist die Bestimmung der Einbettungsstellen und -breiten (2). Die Anzahl der einzubettenden Bits  $A$  bestimmt sich aus dem Platzbedarf für eine Binärcodierung von  $D$  sowie einer geeigneten Einbettungsmarkierung  $M$ . Da die Einbettungsmarkierung andererseits von der gewählten Einbettungsbitbreite und somit indirekt vom Platzbedarf abhängt, ist eine gegenseitige Abhängigkeit gegeben. Den einfachsten Ausweg liefert eine Menge gültiger Markierungen mit der Eigenschaft, daß jede Markierung mit derselben Anzahl Bits codierbar ist. Soll eine Fehlerkorrektur durchgeführt werden, sind entsprechend mehr Bits bereitzustellen. Aus der Quantisierungsart und -schrittweite läßt sich die Einbettungskapazität für jedes Subbandsample bestimmen. Die Verteilung der  $A$  Bits auf die Subbandsamples ist einer Bitallokierung wie sie in psychoakustischen Encodern vorkommt sehr ähnlich. Es gibt somit mehrere Bitallokierungsstrategien. Da bei der Direkteinbettung der maximale Quantisierungsfehler erhöht werden kann, ist es z.B. in diesem Fall ratsam, die  $A$  Bits besonders gleichmäßig auf die Werte  $\tilde{s}$  zu verteilen. Ein konkreter Bitallokierungsalgorithmus wird in Kapitel 7 anhand der Realisierung unter MPEG-1 Layer II vorgestellt. In Schritt 2 legt das Tupel  $(i_j, t_j)$  fest, daß in das  $i_j$ -te Subbandsample eine  $t_j$  Bit breite Einbettung vorgenommen wird.

In Schritt 3 wird die binärcodierte einzubettende Information  $\text{bin}(D, M)$  in eine Folge von Binärstrings geeigneter Breite partitioniert. Schritt 4 vollzieht die Einbettung nach Einbettungsart  $E_1$ . Die nicht zur Einbettung verwendeten Subbandsamples werden unverändert übernommen (5). Output des Decoders ist der rücktransformierte Datenblock  $T^{-1}\tilde{s}$ .

Es sei erwähnt, daß die Bezeichnungen „Encoder“ und „Decoder“ in diesem Zusammenhang eine Doppelbedeutung haben. In erster Linie werden damit das Kompressions- und Dekompressionsverfahren beschrieben. Andererseits kann auch der decoderseitige Vorgang der Dateneinbettung 5.6 als Encoding und der Vorgang der Wiedergewinnung dieser Daten im Encoder 5.7 als Decoding bezeichnet werden. Soweit nicht im Zusammenhang erwähnt, werden

diese Begriffe hier stets in der ersten Bedeutung verwendet.

Der hier vorgestellte Codec verletzt wegen der Verwendung der Direkteinbettung u.U. die aufgrund des psychoakustischen Modells bestimmten maximalen Rekonstruktionsfehler. Daher kann der Erhalt der Qualität des Audiosignals gegenüber einer herkömmlichen Kompression und Dekompression nur approximativ gewährleistet werden. Wichtig in diesem Zusammenhang ist auch eine konservative Bitallokierungsstrategie für die Einbettung (s.o.).

### 5.3.3 Codec basierend auf Einbettungsregionen

Abb. 5.8 zeigt einen Encoderalgorithmus zur Verwendung mit Einbettungsregionen. Die Unterschiede zum vorherigen Algorithmus 5.6 liegen in der Bestimmung der Fehlerregionen  $\mathcal{F}(s, Q)$ : wird eine Einbettung erkannt, d.h.  $M \in \mathcal{M}$ , so können in Abhängigkeit des aus  $D$  bestimmbaren Quantisierers  $Q$  und des Vektors  $s$  die Fehlerregionen bestimmt werden. Da die Einbettungsregionen untereinander einen leeren Schnitt haben, kann aus jedem  $e \in \mathcal{E}_{i,k}$  eindeutig auf das zugehörige  $\mathcal{F}_{i,k}$  geschlossen werden. Ist  $M \notin \mathcal{M}$ , so werden die Fehlerregionen auf herkömmliche Art und Weise bestimmt. Der Output dieses Encoders enthält zusätzlich die Fehlerregionsinformationen.

Ein zugehöriger Decoder ist in Abb. 5.9 angegeben. Der wesentliche Unterschied zum Decoder in Abb. 5.7 liegt in der Einbettungsart. Entsprechend einer Verwendung von Fehlerregionen, die einen zusätzlichen Input für jeden Datenrahmen darstellen, sind die Schritte 3 und 4 abgeändert. In Schritt 3 werden die Einbettungsstellen so ausgewählt, daß bei einer entsprechenden Aufteilung des einzubettenden Binärstrings alle resultierenden Teilstrings (in ihrer Interpretation als Binärzahl) gültige Einbettungsvektoren sind. Dies entspricht der Anforderung  $(e_j)_2 \in [0 : e_{i,j}]$ . Die Mengen der Einbettungsvektoren wurden im Algorithmus o.B.d.A. als Mengen positiver Zahlen gewählt. Natürlich kann diese Wahl leicht auf negative Zahldarstellungen erweitert werden. In Schritt 4 findet die Einbettung in Form einer Addition statt. Der Rest des Algorithmus ist analog zum vorherigen Decoder-Algorithmus.

Das Hauptresultat dieses Kapitels ist, daß Codecs nach Abb. 5.8 und Abb. 5.9 aufgrund der Resultate über Einbettungs- und Fehlerregionen aus Kapitel 4 den Erhalt der perzeptuellen Qualität des Audiosignals erster Generation bei iterierter Kompression und Dekompression garantieren gdw.

- bei jeder Einbettung genügend Einbettungskapazität zur Verfügung steht,
- encoderseitig keine Fehldetektionen auftreten (vgl. Abschnitt 4.7), und
- keine Arithmetik- oder sonstige Fehler zwischen bzw. während Codier- und Decodiervorgang auftreten.

Genauer wird in diesem Fall das Signal erster Generation bei allen weiteren Generationen exakt reproduziert. Eine wichtige Hypothese ist hierbei natürlich, daß keine der durch das psychoakustische Modell erlaubten Abweichungen des Signals erster Generation vom Originalsignal zu perzeptuellen Beeinträchtigungen führt.

### 5.3.4 Implementierungsfragen

Viele Details der Komponenten obiger Verfahren sind implementationsabhängig. Es sind allerdings einige allgemeine Punkte zu beachten.

### 5.3.4.1 Synchronisation

Bisher wurde bei Angabe der Algorithmen idealisierend implizit vom Vorhandensein eines entsprechenden Datenblocks ausgegangen, der für die Zwecke der Verfahren zeitlich *synchronisiert* vorliegt. Die Annahme der zeitlichen Synchronisation ist aber aufgrund fehlender Translationsinvarianz der betrachteten Verfahren nicht zutreffend.

In den Layern II und III des MPEG-Verfahrens findet die rahmenweise Verarbeitung z.B. in Blöcken von jeweils 1152 neuen Samples statt. Somit gibt es alle 1152 Samples einen solchen Verarbeitungsblock. Da die eingebetteten Informationen stets alle Informationen für den betreffenden Datenrahmen enthalten, kann die Verarbeitung mit den obigen Encoder-Algorithmen mit jedem dieser Blöcke beginnen. Vor Beginn einer Encodierung muß allerdings aus allen 1152 theoretisch möglichen Blockanfängen der richtige herausgefunden werden.

Eine einfache aber dennoch handhabbare Methode ist das Testen aller Möglichkeiten für einen Blockbeginn nacheinander. Da die vorliegenden Subbandtransformationen linear sind und in unserem Fall aus einer bestimmten Anzahl von FIR Filtern aufgebaut sind, können hierzu Algorithmen zur schnellen Faltung eingesetzt werden. Im Falle von MPEG kann etwa eines der 32 Filter,  $h$ , gewählt und das Eingangssignal  $x$  auf einer Länge von  $2 \cdot 1152$  Samples hiermit gefaltet werden,  $s = c_h(x)|_{[i:i+2304]}$  für ein geeignetes  $i \in \mathbb{Z}$ . Die 32 möglichen Subbänder entstehen hieraus für  $0 \leq k < 32$  durch  $s_k := (\downarrow 32) \circ T_k[s]$ . Im folgenden wird jeder der Kandidaten auf Einbettungsmarkierungen getestet. Ist die Suche erfolgreich, errechnet sich aus der Subbandnummer  $k$  die Blockposition der Filterbank (translationsinvariant unter Shifts um 32 Samples) und aus der Position  $p$  innerhalb des Subbands die Absolutverschiebung  $p \cdot k$ . Zum Funktionieren der Methode muß natürlich das Subband bzw. das Filter geeignet gewählt sein, z.B. als eines der Subbänder, in die üblicherweise als erstes eingebettet wird. Tests haben gezeigt, daß es solch typische Kandidaten gibt.

### 5.3.4.2 Einbettungsregionen und Kompressionsraten

Der Preis für die Einbettung unter Gewährleistung des Erhalts des maximalen Rekonstruktionsfehlers ist eine Erhöhung der Bitrate im komprimierten Datenstrom. Daher ist die Einbettung in diesem Falle günstigstenfalls an so wenigen Stellen wie möglich durchzuführen. Problematisch ist allerdings, daß die Fehlerregionen für ein Originalsignal nur encoderseitig bestimmt werden können. Der Encoder muß also bereits über den Bitbedarf zur Einbettung informiert sein. Daher muß der Bitallokierungsalgorithmus für die Einbettung sowohl im Encoder als auch im Decoder durchgeführt werden.

Zu bemerken ist weiterhin, daß die Fehlerregionen eindeutig durch die Einbettungsregionen festgelegt sind. Daher muß die Fehlerregionsinformation *nicht* eingebettet werden, und der Decoder kann aus den Einbettungsregionen die Fehlerregionen bestimmen. Dies findet z.B. in Schritt 4.(d) des Algorithmus in Abb. 5.8 Anwendung.

### 5.3.4.3 Bitreservoirtechniken

Ein wichtiges Konzept in Zusammenhang mit der Gewährleistung einer genügend großen Einbettungskapazität für jeden Datenrahmen ist die rahmenadaptive Einbettung. Dieses Konzept ist mit der im dritten Layer von MPEG-1 verwendeten Bitreservoirtechnik vergleichbar. Die Idee besteht darin, in gewisse Datenrahmen, für die eine größere Einbettungskapazität zur Verfügung steht, bereits die Encoderparameter für darauffolgende Datenrahmen abzulegen. Dies hat den Vorteil, daß eine zu geringe Einbettungskapazität einzelner Datenrahmen

ausgeglichen werden kann. Bei der Realisierung solch einer Technik sind jedoch gewisse Problematiken zu beachten:

- Gegebenenfalls können selbst mit einer Bitreservoirtechnik nicht die Parameter aller Datenrahmen eingebettet werden. Dies kann z.B. bei Stücken geringer Lautstärke vorkommen. Für diesen Fall ist ein Konzept zur Verwaltung derjenigen Rahmen, für die Parameter eingebettet wurden, zu erstellen. Dabei ist zu beachten, daß die Wahl dieser Rahmen günstigstenfalls abhängig von deren perzeptueller Wichtigkeit getroffen wird.
- Bei Verwendung dieser Technik wird es notwendig eine a priori unbekannte Anzahl von Datenrahmen während der Rücktransformation zwischenspeichern, da in diese die Parameter nachfolgender Rahmen eingebettet werden sollen. Zur praktischen Realisierbarkeit im Hinblick auf einen akzeptablen Speicherplatzbedarf ist hier ein vernünftiger Kompromiss bezüglich eines maximal möglichen look-aheads (in Datenrahmen) zu finden.

Die Bitreservoirtechnik ist besonders attraktiv, da hierdurch prinzipiell (fast) der gesamte für die Einbettungsmarkierungen zu verwendende Platz eingespart werden kann. Dies wird dadurch ermöglicht, daß alle Encoderinformationen eines bestimmten Datenrahmens schon im vorherigen Datenrahmen übertragen werden. Die Idee hierzu wurde durch eine Bemerkung von Hassenrik [13] inspiriert. Wie die Ergebnisse des folgenden Kapitels zeigen, ist eine solche Einsparung bei der Verwendung des Verfahrens in realen Audiocodierern von fundamentaler Bedeutung. Bei dieser Vorgehensweise muß jedoch eine geeignete Sonderbehandlung für den jeweiligen Einbettungsbeginn, also den ersten Block, der decoderseitig detektiert werden muß, getroffen werden, da für diesen die Einbettungsstellen nicht vorher bekannt sind. In diesem Fall kann das Standardverfahren verwendet werden.

## 5.4 Mehrstufige Codecs

Zahlreiche gebräuchliche Codecs sind durch eine Hybridstruktur ausgezeichnet, sie verwenden mehrere kaskadierte Transformationsstufen. Das für uns wichtigste Beispiel ist MPEG-1, Layer III. Hier wird eine 32-Band Filterbank mit ein bis zwei nachgeschalteten Orthogonaltransformationen verwendet, siehe Kapitel 7. Prinzipiell ist solch eine Transformationskaskade beschreibbar durch eine Folge von Abbildungen

$$\mathbb{R}^n \xrightarrow{T_1} \mathbb{R}^n \xrightarrow{T_2} \dots \xrightarrow{T_\ell} \mathbb{R}^n.$$

Beispielsweise ist im Falle von MPEG  $\ell = 2$  oder  $\ell = 3$  je nach Blocktyp. Wir schränken uns auf den Fall  $\ell = 2$  ein und betrachten

$$\mathbb{R}^n \xrightarrow{T} \mathbb{R}^n \xrightarrow{S} \mathbb{R}^n$$

für invertierbare lineare Abbildungen  $T, S$ . Wir bezeichnen die Koordinatenräume des  $\mathbb{R}^n$  für Koordinatenmengen  $M \subseteq [1 : n]$  mit  $\mathbb{R}^M$ . Weiterhin nehmen wir an, daß  $S$  als direkte Summe

$$S = S_1 \oplus S_2 \oplus \dots \oplus S_N$$

mit  $\forall i : S_i : \mathbb{R}^{M_i} \rightarrow \mathbb{R}^{M_i}$  und  $M$  als disjunkte Vereinigung  $M_1 \sqcup M_2 \sqcup \dots \sqcup M_N = [1 : n]$  geschrieben werden kann. Dies entspricht den in der Praxis vorkommenden Blocktransformationen, z.B. der Aliasingcancellation oder der MDCT im Falle von MPEG. Die blockweise

Einbettung wird ebenfalls durch Abbildungen

$$E = E_1 \oplus E_2 \oplus \dots \oplus E_N, \quad E_i : \mathbb{R}^{M_i} \longrightarrow \mathbb{R}^{M_i},$$

beschrieben. Bei zweistufigen Einbettungen sind entsprechend zwei Einbettungsfunktionen  $E, F$  gegeben. Die Rücktransformation für einen Datenrahmen oder Verarbeitungsblock inklusive Einbettung wird dann durch  $T^{-1}FS^{-1}E$  beschrieben. Damit sich die Einbettungen nicht gegenseitig „stören“, sind diese in gewisser Weise unabhängig zu wählen. Mit *Störung* ist hier eine Einbettung von  $E$  und  $F$  in denselben Koordinatenraum, z.B.  $\mathbb{R}^{M_1}$ , gemeint. Die Auswirkung solch einer Doppeleinbettung ist im Falle einer Direkteinbettung i.a. die Zerstörung der zuerst eingebetteten Information, auf jeden Fall jedoch eine potentielle Überschreitung des maximalen Rekonstruktionsfehlers. Eine Doppeleinbettung kann im einfachsten Fall durch die Wahl disjunkter Einbettungsräume bzw. -koordinaten für  $E$  und  $F$ ,

$$\begin{aligned} E_i \neq id_{\mathbb{R}^{M_i}} &\implies F_i = id_{\mathbb{R}^{M_i}} && \text{und} \\ F_i \neq id_{\mathbb{R}^{M_i}} &\implies E_i = id_{\mathbb{R}^{M_i}}, \end{aligned}$$

erreicht werden.

**Encoder-Algorithmus** (zur Verwendung mit Einbettungsregionen)

**Input:** Signalblock  $x \in \mathbb{R}^n$ , Subbandtransformation  $T$ , Coderparameter  $P$ , Psychoakustisches Modell  $\Psi$ , Bitallokierungsfunktion  $b$ , Detektorfunktion  $\mathcal{D}$  auf  $\mathbb{R}^n$ ,  $\mathcal{M} \subset \text{Im}(\mathcal{D})$  Menge aller gültigen Markierungen.

1. Berechne  $s := Tx$ .
2. Detektiere  $D(s) =: M$ .
3. Falls  $M \notin \mathcal{M}$ 
  - (a) Berechne  $\Psi(x)$ . Setze  $s' := s$ .
  - (b) Berechne  $b(\Psi(x), P, s, x)$  und daraus Encoderparameter  $E(x)$ . Insbesondere ist durch  $b(\cdot)$  ein Quantisierer  $Q$  bestimmt.
  - (c) Bestimme Fehlerregionsinformation  $(s, Q) \mapsto \mathcal{F}(s, Q)$ .
4. Sonst
  - (a) Decodiere  $s \mapsto \tilde{Q}(s) =: (s', D(x))$ .
  - (b) Berechne Encoderparameter  $D(x) \mapsto E(x)$ .
  - (c) Bestimme Quantisierer  $Q$  aus  $E(x)$ .
  - (d) Berechne Fehlerregionsinformation  $(s, Q) \mapsto \mathcal{F}(s', Q)$ .
5. Quantisiere  $s'$  mit Hilfe von  $E(x)$ :  $s' \mapsto Q(s')$ .
6. Bestimme Decoderparameter  $E(x) \mapsto D(x)$ .

**Output:** Codeworte  $Q(s')$ , Decoderparameter  $D(x)$ , Fehlerregionsinformation  $\mathcal{F}(s, Q)$ .

Abbildung 5.8: Encoder-Algorithmus zur Verwendung mit der auf Einbettungsregionen basierten Einbettung aus Kapitel 4.

**Decoder-Algorithmus** (zur Verwendung mit Einbettungsregionen)

**Input:** Codeworte  $Q(s)$ , Decoderparameter  $D(x)$ , Fehlerregionsinformation  $\mathcal{F}(s, Q)$  mit entsprechenden Einbettungsvektorregionen  $E(s, Q) = ([0 : E_1], \dots, [0 : E_n])$ .

1. Rückquantisiere  $Q(s) \mapsto \tilde{s} =: (\tilde{s}_1, \dots, \tilde{s}_n)$ .
2. Bestimme Einbettungsstellen

$$(\tilde{e}, D(x), E(s, Q)) \mapsto (i_1, \dots, i_m), \quad 1 \leq i_k < i_{k+1} \leq n,$$

bilde einen Binärstring  $\text{bin}(D(x), M)$  aus  $D(x)$  und einer geeigneten Detektormarkierung  $M$  und partitioniere  $\text{bin}(D(x), M) =: e_1 \cdots e_m$ , so daß  $(e_j)_2 \in [0 : E_{i_j}]$ .

3. Bette für  $1 \leq j \leq m$  ein:

$$E_1 : (\tilde{s}_{i_j}, e_j) \mapsto s_{i_j} + (e_j)_2 =: \tilde{s}_{i_j}.$$

4. Setze  $\tilde{\tilde{s}}_k := \tilde{s}_k$  für alle  $k$ , wo  $k \neq i_j$  für  $1 \leq j < m$ .
5. Rücktransformiere  $\tilde{\tilde{s}} \mapsto T^{-1}\tilde{\tilde{s}}$ .

**Output:** Signalblock  $T^{-1}\tilde{\tilde{s}}$ .

Abbildung 5.9: Decoder-Algorithmus zur Verwendung mit der auf Einbettungsregionen basierten Einbettung aus Kapitel 4.

## Kapitel 6

# Codec-Realisierung: MPEG-1/Audio, Layer II

In diesem Kapitel wird die konkrete Realisierung eines Codecs zur Vermeidung von Generationseffekten aufbauend auf dem MPEG-1/Audio Standard (ISO/IEC 11172-3) [18] beschrieben. Von den drei möglichen, in der Komplexität der Codecs verschiedenen Layern dieses Standards wurde Layer II ausgewählt. Dieser Layer liefert bei mittlerer Codec-Komplexität Kompressionsraten von bis zu 1 : 8 in HiFi-Qualität. Eine Realisierung unter Layer I ist einfacher und prinzipiell mit einer Untermenge der Methoden dieses Kapitels möglich. Eine Realisierung unter Layer III ist aufwendiger, da hier eine mehrstufige Transformation sowie eine variable Bitrate zu berücksichtigen sind. Ein Entwurf für eine Realisierung unter Layer III wird in Kapitel 7 gegeben. Zur Implementierung wurde der Quellcode [40] der *MPEG/Audio Software Simulation Group* für die Layer I und II, der in der Programmiersprache C für verschiedene Plattformen zur Verfügung steht, verwendet.

Zunächst wird kurz der Aufbau eines Layer II-Codecs nach ISO/IEC 11172-3 beschrieben. Dabei wird insbesondere auf die globalen Coderparameter, die Verarbeitungsrahmen und die anfallenden De- und Encoderparameter eingegangen. Im zweiten und dritten Abschnitt werden die notwendigen Modifikationen zur Realisierung eines Direkteinbettungscodecs angegeben. Anschließend wird ein Fehlerregionencodec konzipiert. Im fünften Abschnitt werden Testergebnisse der entworfenen Codecs angegeben und deren Leistungsfähigkeit demonstriert. Abschließend wird auf Erweiterungsmöglichkeiten zu den vorliegenden Implementationen eingegangen.

### 6.1 Codecs nach ISO/IEC 11172-3, Layer II

In Abb. 6.1 ist der Aufbau eines Layer-II Codecs nach dem MPEG-1 Standard dargestellt. Obwohl im Standard [18] lediglich das Format des Audio-Datenstroms und die Decodierprozedur verbindlich festgeschrieben sind und die dort angegebenen Encodierprozeduren sowie die psychoakustischen Modelle nur *Vorschläge* sind, soll hier in allen auf den Standard bezogenen Diskussionen nicht explizit zwischen vorgeschlagenen und vorgeschriebenen Komponenten unterschieden werden.

Innerhalb des Standards gibt es weiterhin die Möglichkeit, vier verschiedene Kanalmodi zu verwenden: Mono, Stereo, Dual (zwei getrennte Kanäle, z.B. für Zweikanalton) und Joint Stereo. Im Joint-Stereo Modus besteht durch die Verwendung der *Intensity Stereo*-Codierung

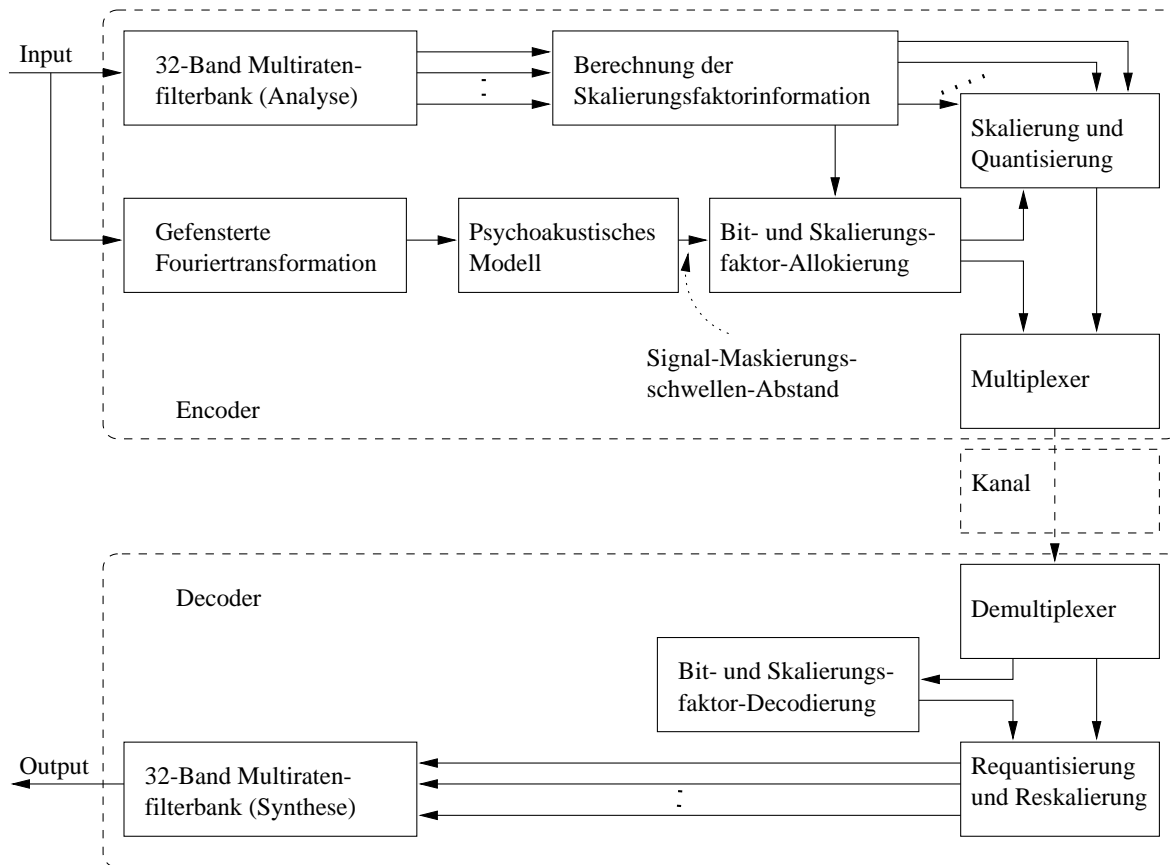


Abbildung 6.1: Aufbau eines Codecs nach ISO/IEC 11172-3, Layer II.

bzw. optionaler *M/S*-Codierung in Layer III die Möglichkeit, zusätzliche Redundanzen zwischen linkem und rechtem Kanal auszunutzen [44]. Wir wollen uns hier auf die Betrachtung des Stereomodus beschränken. Dazu reicht es aus, im folgenden jeweils *einen* Kanal zu betrachten. Die Verarbeitungsschritte für den anderen Kanal laufen entsprechend ab.

### 6.1.1 Aufbau des Codecs

MPEG-1 Audiocoders sind der Klasse der Subbandcodierer zuzurechnen. Das PCM-Signal wird in Rahmen von jeweils 1152 Samples mit einer 32-Band Polyphasenfilterbank in 32 Subbänder transformiert. Hierbei handelt es sich um eine Quadrature-Mirror-Filterbank, die eine fast perfekte Rekonstruktion (engl. *Near-Perfect-Reconstruction* (NPR)), siehe [42, 43], erlaubt. Da jedes der 32 Filter aus einem Prototypfilter durch Cosinusmodulation bzw. spektrale Verschiebung hervorgeht, ist eine lineare Frequenzbandaufteilung gegeben. Diese entspricht besonders in niedrigen Frequenzbändern ( $48000 \text{ Hz} / 32 = 1500 \text{ Hz}$ ) nicht einer Aufteilung in kritische Bänder (Breite von ca. 100 Hz unterhalb von 500 Hz, danach ca. 20% der Zentrumsfrequenz eines Bandes), liefert jedoch einen guten Kompromiß bezüglich der Zeit-Frequenz-Auflösung: wird eine zu feine Spektralaufklärung gewählt, entstehen unter Umständen Pre- und Postechos, die sich sehr störend auswirken können. Weiterhin ist die gewählte Filterbank mit Hilfe schneller Cosinustransformationen sehr effizient implementierbar. Die Filter haben eine Länge von jeweils 512 Filterkoeffizienten und weisen eine Stop-

banddämpfung von ca. 96 dB auf. Der Rekonstruktionsfehler der Filterbank liegt ungefähr in derselben Größenordnung und entspricht somit ungefähr der fünften Dezimalstelle des 16-Bit PCM Wertebereichs  $[-32768 : 32767]$ .

In jedem Subband werden aus den  $1152/32 = 36$  Subbandsamples drei Blöcke aus jeweils 12 aufeinanderfolgenden Samples gebildet. Für jeden der Skalierungsblöcke wird ein Skalierungsfaktor aus  $(\sqrt[3]{2^{(3-n)}})_{n \in [0:62]}$  berechnet. Dies entspricht einer Dynamik von 120 dB bei einer Auflösung von ca. 2 dB. Da innerhalb der Subbänder bei stationären Signalen häufig Skalierungsfaktoren ähnlicher Größenordnungen aufeinanderfolgen, wird in diesem Falle nur der jeweils größte dieser Faktoren übertragen. Die Auswahl richtet sich nach den Größenordnungen der drei benachbarten Skalierungsfaktoren und liefert einen bis drei verschiedene Skalierungsfaktoren pro Subband. Die Auswahlinformation ist zusätzlich zum Decoder zu übertragen.

Die *Spektralanalyse* wird mit Hilfe einer gefensterten Fouriertransformation der Länge 1024 durchgeführt, wobei ein Cosinus- oder Hann-Fenster verwendet wird. Aus diesen Spektralinformationen bzw. dem Leistungsdichtespektrum berechnet man mit Hilfe eines psychoakustischen Modells für jedes Subband den Abstand zwischen Signal und Maskierungsschwelle (SMR, in dB).

Die *Bitallokierung* wird innerhalb eines iterativen Algorithmus ausgeführt. Dabei wird für jedes Subband ein aktueller Abstand zwischen Maskierungsschwelle und Rauschpegel (Noise-to-Mask-Ratio, NMR) bestimmt. In einem Durchlauf der Hauptschleife des Algorithmus wird nun dem Subband mit dem niedrigsten NMR eine höhere Bitrate zugeschlagen. Die Darstellung der Subbänder wird somit mit fortschreitender Laufzeit immer genauer. Der Rauschpegel wird hierbei in Abhängigkeit von der momentanen Bitrate für jedes Subband bestimmt. Es werden die bekannten Rauschpegel für lineare Quantisierer, siehe [26], zugrundegelegt und jeweils aus Tabellen nachgeschlagen.

Während der Bitallokierung wird über den momentanen Bitverbrauch für den jeweiligen Datenrahmen Buch geführt. Werden einem Subband erstmalig Bits zugeschlagen, müssen auch erstmalig die für dieses Subband benötigten Skalierungsfaktoren in den Bitbedarf einbezogen werden. Die für jedes Subband mögliche Quantisierungsauflösung ist abhängig von den globalen Parametern Gesamtabtastrate, MPEG Layer und Gesamtbitrate sowie von der Subbandnummer. Dies wird im Standard innerhalb einiger Tabellen ([18], ab S. 46) verwaltet. Die Zuordnung zwischen Quantisierungsbitbreite und Anzahl verwendeter Quantisierungsstufen wird durch die Quantisierungsklassen ([18], S. 53) geregelt. Bei den 3-, 5- und 9-Schritt Quantisierern werden hier stets drei Samples zu einem Codewort zusammengefaßt (engl. *grouping*). Dadurch kann nach Erfahrungswerten ca. 40% der Bitrate eingespart werden [5].

Entsprechend der Bitallokierung werden die Subbandsamples nachfolgend skaliert und linear quantisiert. Hierzu wird ein Subbandwert  $x$  in Abhängigkeit von der gewählten Quantisierungsauflösung  $q$  via  $x \mapsto A_q x + B_q$  linear transformiert. Die signifikanten Bits hiervon (mit invertiertem höchstwertigem Bit) ergeben das Codewort. Im Falle des Groupings kommt noch eine lineare Transformation als Vorverarbeitung hinzu.

Abschließend werden die Seiteninformationen und Codeworte in den MPEG-Bitstroms geschrieben. Der Decodierer zur rahmenweisen Rekonstruktion arbeitet entsprechend.

Es ist wichtig zu erwähnen, daß sich aufgrund der Verwendung einer Polyphasenfilterbank benachbarte Verarbeitungsrahmen überlappen. Daher müssen sowohl während der Decodierung als auch während der Encodierung einige Samples am Ende des aktuellen Verarbeitungsblocks zur Verwendung mit dem nächsten Block zwischengespeichert werden. Bei einer 32-Band Filterbank mit Filterlänge 512 sind dies 480 Samples.

### 6.1.2 Bitdatenstrom, globale und rahmenbezogene Parameter

Das Format eines MPEG-1 Audio-Datenrahmens wird in [18] ab S.14 beschrieben. Der Rahmen besteht grob aus Headerdaten, Fehlerkorrekturbits, den eigentlichen Audionutzdaten, sowie einem Freiraum für sonstige Daten (ancilliary data). Im Header jedes Rahmens finden sich Informationen über globale Parameter, insbesondere die Anzahl der Kanäle, die Bitrate und den gewählten Layer. Diese drei Informationen steuern indirekt den Platzbedarf und die Art der Biteinbettung:

- Soll der Joint-Stereo-Mode verwendet werden, werden ab einer bestimmten Subbandnummer (abgelegt in der Variablen `bound`) linker und rechter Kanal gemeinsam codiert. Ab diesem Subband wird für beide Kanäle eine gemeinsame Quantisierungsaufösung verwendet.
- Die Bitrate steuert in Verbindung mit der Layer-Information, welche Quantisierungstabelle verwendet wird. Aus der Quantisierungstabelle kann, abhängig von der gewählten Quantisierungsaufösung, die Einbettungsbitbreite bestimmt werden. Die Quantisierungstabelle legt ebenfalls fest, bis zu welchem Subband maximal Bits allokiert werden können.

Die globalen Parameter sollen als fest betrachtet werden. Wir gehen somit davon aus, daß sich beim wiederholten Encodieren die gewünschte Bitrate, die verwendete Abtastrate und die Kanalverwendung nicht ändern. Daher sind nur die rahmenbezogenen Daten vom einen zum anderen Codec zu übertragen bzw. einzubetten. Die relevanten Encoderparameter (hier gleich den Decoderparametern) sind die Bitallokierung, die Skalierungsfaktorauswahl und die Skalierungsfaktoren selbst. Im folgenden sind in Klammern jeweils die im Standard verwendeten Variablennamen angegeben.

**Bitallokierung (allocation):** Bit- und Abtastrate legen die Anzahl der Subbänder fest. Pro Subband (und pro Kanal) wird in 2 – 4 Bits die Quantisierungsaufösung festgelegt. Da diese Anzahl Bits für jedes Subband fest ist, kann sie im voraus bestimmt werden. Bei 48 kHz und einer Bitrate von 96 kBit/Sek. (pro Kanal) ergeben sich beispielsweise 27 Subbänder und

$$11 \cdot 4 \text{ Bit} + 12 \cdot 3 \text{ Bit} + 4 \cdot 2 \text{ Bit} = 88 \text{ Bit},$$

bei 44,1 kHz und 128 kBit/Sek. hingegen 30 Subbänder und 96 Bit.

**Skalierungsfaktorauswahl (scfsi):** Für jedes verwendete Subband (d.h. falls Bits hierfür allokiert wurden, siehe Variable `allocation`) wird das verwendete Muster der Skalierungsfaktoren in zwei Bits abgelegt. Bei Skalierungsblöcken in der Reihenfolge  $s_1, s_2, s_3$  wird damit ausgewählt, ob bei keinem, bei  $s_1$  und  $s_2$ , bei allen, oder bei  $s_2$  und  $s_3$  ein gemeinsamer Skalierungsfaktor verwendet wird. Es werden somit maximal

$$2 \cdot \#\text{Subbänder} \cdot \#\text{Kanäle} \text{ Bit}$$

benötigt. In den obigen Fällen ergeben sich 54 bzw. 60 Bit.

**Skalierungsfaktoren (scalefactor):** Für jedes verwendete Subband werden, je nach Anzahl verwendeter Skalierungsfaktoren (siehe `scfsi`), ein bis drei Skalierungsfaktoren mit jeweils 6 Bit abgelegt. Es werden maximal

$$6 \cdot \#\text{Subbänder} \cdot \#\text{Kanäle} \cdot 3 \text{ Bit}$$

benötigt. In den obigen Fällen ergeben sich 486 bzw. 540 Bit.

Im ersten Beispiel ergibt sich ein maximaler Gesamtbitaufwand von  $88 + 54 + 486 = 628$  Bit Encoderinformation pro Datenrahmen. Eine Speicherung der dabei maximal verwendeten 27 Subbänder mit einer Genauigkeit von 16 Bit benötigt hingegen 15552 Bit. Dies entspricht einem Verhältnis von  $\sim 1 : 24,75$ .

### 6.1.3 Leistungsmerkmale

Unter Layer II werden die drei MPEG-1 Abtastraten von 32000, 44100 und 48000 Hz in den Stereobitraten von 64, 96, 112, 128, 160, 192, 224, 256, 320 und 384 kBit pro Sekunde unterstützt. Im Vergleich dazu liegt die reine Audiobitrate für Stereosignale bei einer Abtastrate von 44100 kHz auf Compact Disks und DAT-Recordern bei 1411,2 kBit pro Sekunde ( $= 2 \cdot 44100 \cdot 16$  Bit). DAT-Recorder unterstützen zusätzlich die beiden anderen MPEG-1 Abtastraten, wobei die Quantisierungsart bei 32000 Hz Abtastrate allerdings durch einen logarithmischen 12-Bit Quantisierer gegeben ist. Bei CD und DAT kommen noch Fehlerkorrekturmechanismen hinzu, so daß die Gesamtbitrate bei CDs mit ca. 4320 kBit/Sek. und bei DATs mit ca. 3080 kBit/Sek. signifikant erhöht wird. Die Kompressionsraten werden allerdings immer anhand eines Vergleichs zwischen reinen Audiobitraten gebildet. Für eine Abtastrate von 48 kHz ergibt sich in den Layern I bzw. II bzw. III perzeptuell transparente Qualität, falls eine Kompressionsrate von mindestens 384 bzw. 192 bzw. 128 kBit/Sek. eingehalten wird. Dies entspricht Kompressionsraten von  $1 : 4$ ,  $1 : 8$  und  $1 : 12$ . Der hier untersuchte Layer II Codec erreicht bei einer Bitrate von 128 kBit/Sek. in der verwendeten Encoder-Realisierung [40] eine zwar leicht beeinträchtigte aber dennoch akzeptable Audioqualität.

## 6.2 Direkteinbettungscodec: Decoder

Da der hier beschriebene Direkteinbettungscodec keine Fehlerregionen verwendet, wird keine Erhöhung der MPEG-Bitrate verursacht. Ebenso wird die Einbettungsbitallokierung erst im Decoder notwendig.

### 6.2.1 Strukturierung und Modifikation des Originalcodecs

Die einzubettenden Encoder- und gleichzeitig Decoderparameter sind die in den Strukturen `allocation`, `scfsi` und `scalefactor` abgelegten Daten. Diese liegen zwecks Speicherung im MPEG Bitdatenstrom bereits in einer sehr kompakten Form vor. Im Decoder werden zunächst wie gewöhnlich diese Parameter sowie die Codeworte aus dem Bitdatenstrom extrahiert. Die Codeworte werden rückquantisiert. Es folgen die Schritte

- Ablegen der einzubettenden Information in einem Bitbuffer,
- Bestimmung des zur Einbettung verfügbaren Speicherplatzes,
- Bitallokierung für die Einbettung,
- Einbettung der Daten in die rückquantisierten Codeworte.

Anschließend wird wie gewöhnlich die Rücktransformation durchgeführt.

### 6.2.2 Blockeinteilung und Bitdatenstrom

Da sowohl die Quantisierungsaufösung als auch die Skalierungsfaktoren Einfluß auf die Einbettungsbitbreite haben, werden die Subbänder entsprechend in Blöcke aufgeteilt. Da der Quantisierer für ein vollständiges Subband, die Skalierungsfaktoren jedoch gegebenenfalls für jedes Drittel dieses Subbands separat verwendet werden, bestimmen sie auch die Einbettungsblockeinteilung. Jedes Subband wird zur Einbettung in drei Skalierungsblöcke der Länge 12 Samples eingeteilt. Innerhalb dieser Blöcke ist die Einbettungsbitbreite konstant.

Alle einzubettenden Daten eines Rahmens werden in einen Bitbuffer gepackt, der einen Blockzugriff auf beliebige Blöcke von  $n \times k$  Bits ausgehend von einer aktuellen Zeigerposition erlaubt. In der Implementation ist dieser Buffer durch die Datenstruktur `robust_buffer` realisiert. Der Zeiger auf eine Bufferposition ist bitweise positionierbar, der blockweise Zugriff schreibend und lesend möglich.

### 6.2.3 Verfügbare Einbettungsbitbreite

Für jeden der drei Skalierungsblöcke des  $k$ -ten Subbands wird aus der Quantisierungsaufösung  $A_k$  (in Bits) und dem entsprechenden Skalierungsfaktor  $s_{k,i}$ ,  $1 \leq i \leq 3$ , die maximal verfügbare Einbettungsbitbreite  $V_{k,i}$  bestimmt. Hierbei werden nicht verwendete Subbänder ( $A_k = 0$ ) nicht beachtet ( $\forall i : V_{k,i} := 0$ ). Werden weniger als drei Skalierungsfaktoren in einem Subband verwendet, bestimmen sich die  $s_{k,i}$  aus dem verwendeten Übertragungsmuster (siehe `scfsi`). Ist  $s_{k,i} = s_n := \sqrt[3]{2^{(3-n)}}$  für  $n \in [3 : 62]$  der Skalierungsfaktor des Skalierungsblocks ( $k, i$ ), so berechnet sich  $V_{k,i}$  unter Verwendung der Resultate aus Kapitel 4 zu

$$V_{k,i} := 16 - \min(16, \lceil -\log_2 s_{k,i} \rceil + A_k).$$

Wegen  $\lceil -\log_2 s_{k,i} \rceil = \lceil -\log_2 \sqrt[3]{2^{(3-n)}} \rceil = \lceil \frac{n}{3} \rceil - 1$  erhält man

$$V_{k,i} := 16 - \min(16, \lceil \frac{n}{3} \rceil - 1 + A_k).$$

Für  $n \in [0 : 2]$  verliert man durch die Skalierung ein Bit an Genauigkeit. Wir setzen in diesem Fall  $V_{k,i} := 16 - \min(16, A_k - 1)$ . Der Rest der Formel erklärt sich wie in Kapitel 4 motiviert: von einer maximalen PCM Quantisierungsaufösung von 16 Bit ausgehend, gibt die Summe aus  $A_k$  und  $\lceil \frac{n}{3} \rceil - 1$  die Genauigkeit der Rekonstruktion nach Rückquantisierung und Reskalierung in Bits an. Die Differenz aus beidem und der maximal möglichen Auflösung liefert die einbettbare Bitbreite. Ist die Genauigkeit höher als oder gleich 16 Bits, wird  $V_{k,i} := 0$  gesetzt und auf diesem Block keine Einbettung vollzogen. Da in MPEG die Auswirkung der genauigkeitserhöhenden Skalierung innerhalb der Bitallokierung *nicht* in die Berechnung des SMR mit einfließt, kommt es durchaus vor, daß  $\lceil \frac{n}{3} \rceil - 1 + A_k > 16$  ist.

### 6.2.4 Bitallokierungsalgorithmus

Abb. 6.2 zeigt den verwendeten Algorithmus zur Einbettungsbitallokierung anhand eines Codefragments in C. Die Variablen und Funktionen haben die folgende Bedeutung (in der Reihenfolge des Auftretens):

`bits_per_frame`: Anzahl der einzubettenden Bits des aktuellen Rahmens.

`ro_buffer_code_left`: Funktion auf dem Bitbuffer, die die Buffergröße von der aktuellen Bufferposition an zurückgibt.

---

**Einbettungs-Bitallokierung**

```

1  bits_per_frame = ro_buffer_code_left(ro_buf); // In diesem Rahmen
2                                          // benötigte Bits
3  bits_packed_now = 0;
4
5  while((bits_packed_now < bits_per_frame) && can_alloc){
6
7      can_alloc = 0;
8      bits_packed_old = bits_packed_now;
9      bits_packed_now = 0;
10
11     for(j=0; j<fr_ps.sblimit; j++) for(i=0; i<3 ; i++)
12         for(k=stereo-1; k>=0;k--)
13             ro_alloc_p_scale_block[k][i][j] = 0;
14
15     for(j = 0 ; (j<fr_ps.sblimit*stereo*3) &&
16         (bits_packed_now<bits_per_frame);j++){
17
18         if((*emb_blocks[j]).bitbudget>=6 ){
19             can_alloc = 1;
20             jj = (*emb_blocks[j]).j;
21             ii = (*emb_blocks[j]).i;
22             kk = (*emb_blocks[j]).k;
23             ro_alloc_p_scale_block[kk][ii][jj] =
24                 MIN((*emb_blocks[j]).bitbudget-3,(*emb_blocks[j]).alloc);
25             (*emb_blocks[j]).alloc++;
26
27             if(ro_alloc_p_scale_block[kk][ii][jj]>9)
28                 ro_alloc_p_scale_block[kk][ii][jj] = 9;
29
30             bits_packed_now = bits_packed_now +
31                 (12-SIGNATURE_SIZE)*ro_alloc_p_scale_block[kk][ii][jj];
32         }
33     }
34
35     if(bits_packed_old == bits_packed_now) can_alloc = 0;
36
37 }

```

---

Abbildung 6.2: Codefragment mit einer Realisierung eines Algorithmus zur Einbettungsbitallokierung in C.

**ro\_buf**: Instanz des Bitbuffers. Der Bufferzeiger wird vor Beginn der Bitallokierung auf den Anfang des Buffers gesetzt.

**bits\_packed\_now**: Zählt die Anzahl Bits, die während des momentanen Durchlaufs durch die Hauptschleife allokiert werden. Enthält nach dem Durchlauf die Gesamtzahl allozierter Bits.

**bits\_packed\_old**: Enthält die Anzahl allozierter Bits des letzten zurückliegenden Durchlaufs der Hauptschleife.

**can\_alloc**: Boolesche Variable, die anzeigt, ob überhaupt Bits allokiert werden können.

**fr\_ps**: Datenstruktur aus der MPEG-Implementation zur Verwaltung der Parameter des aktuellen Datenrahmens.

**sblimit**: Nummer des niedrigsten bei der aktuellen Bit- und Abtastrate noch verwendeten Subbands.

**i, j, k**: Schleifenzähler für Subbandnummer (**j**), Skalierungsband (**i**) und Kanal (**k**), bzw. für Skalierungsblöcke (**j**) in 15–33.

**stereo**: 1 im Monomodus, 2 in den Stereomodi.

**ro\_alloc\_p\_scale\_block**: Für einen Skalierungsblock während eines Durchlaufs veranschlagte Einbettungsbitbreite.

**emb\_blocks[]** (Feld-) Struktur zur sortierten Verwaltung der Skalierungsblöcke. Die einzelnen Komponenten sind:

**j, i, k**: Subbandnummer, Skalierungsblock innerhalb dieses Subbands und Kanalnummer,

**bitbudget**: Für den Skalierungsblock zur Verfügung stehende Einbettungsbitbreite,

**alloc**: Momentane Allokierungsstufe des Skalierungsblocks.

**bp\_scale\_block**: Nach perzeptuellen Gesichtspunkten für einen Skalierungsblock maximal verfügbare Einbettungsbitbreite (entspricht den  $V_{k,i}$  in 6.2.3).

**SIGNATURE\_SIZE**: Anzahl der Werte eines Skalierungsblocks, die im Falle einer Einbettung (ausschließlich) für die Blockmarkierung verwendet werden.

Ziel des Algorithmus ist die Verteilung der in der Variablen **bits\_per\_frame** gegebenen Anzahl einzubettender Bits auf diejenigen Skalierungsblöcke des aktuellen Datenrahmens, die eine Einbettung nach perzeptuellen Gesichtspunkten erlauben. Dies sind alle Skalierungsblöcke für die **bp\_scale\_block** mindestens eine bestimmte Bitbreite, die sich aus einer minimalen Einbettungsbitbreite und der zur Fehlerkorrektur hinzuzufügenden Redundanz ergibt, enthält. Außerdem dürfen für jeden Skalierungsblock nur höchstens so viele Bits wie in **bp\_scale\_block** angegeben allokiert werden. Zur Verteilung der Bits auf die Skalierungsblöcke wird ein *Greedy*-Algorithmus verwendet, der zunächst denjenigen Blöcken Bits zuschlägt, die die größte Einbettungskapazität besitzen.

Die Hauptschleife in Zeile 5–37 wird solange durchlaufen, bis die geforderte Anzahl Bits allokiert wurde oder sich die Anzahl allozierter Bits von einem auf den nächsten Schleifendurchlauf nicht geändert hat (Zeile 35). Konnten innerhalb des ersten Durchlaufs überhaupt keine Bits allokiert werden, wird die Schleife abgebrochen. Zunächst werden die im vorherigen Durchlauf allokierte Anzahl Bits gerettet (Zeile 8) und der Zähler für die im momentanen Durchlauf allokierte Bitanzahl initialisiert (Zeile 9). In den Zeilen 11–13 werden die den einzelnen Skalierungsblöcken zugewiesenen Einbettungsbitbreiten initialisiert. Die Zeilen 15–33 enthalten die innere Allokierungsschleife, die über alle Skalierungsblöcke läuft. Auf die Skalierungsblöcke wird hierbei über die Struktur `emb_blocks[]` zugegriffen. Dieses Feld liegt zum Zeitpunkt der Bitallokierung, absteigend nach der Größe des Eintrags `bitbudget` sortiert, vor. Hierbei gibt `bitbudget` die Einbettungsbitbreite des entsprechenden Skalierungsblocks an. Kann in einen Skalierungsblock potentiell eingebettet werden und liegt die entsprechende Kapazität über einer gewissen Konstanten (Zeile 18 - die Konstante ist hier 6, bestehend aus drei Bits Fehlerkorrektur und drei Bits minimaler Einbettungsbitbreite), werden diesem Block im folgenden Bits zugeschlagen. Die zu allozierende Bitbreite, Zeilen 23–24, berechnet sich aus dem Minimum der momentanen Allokierungsbreite des Skalierungsblocks `j` (`emb_blocks[j].alloc`) und der perzeptuell zulässigen Bitbreite (`bitbudget`) abzüglich einer globalen Sicherheitskonstanten. Diese Konstante (hier 3) sorgt dafür, daß nicht die volle Einbettungskapazität ausgeschöpft und damit der unter Umständen durch die Direkteinbettung sowieso erhöhte Rekonstruktionsfehler nicht allzu groß werden kann. In Zeile 25 wird die momentane Allokierungsbreite des Skalierungsblocks zur potentiellen Verwendung im nächsten Schleifendurchlauf erhöht. Die maximale absolute Einbettungsbitbreite wird in den Zeilen 27–28 festgelegt (hier 9 Bit). In den Zeilen 30–31 wird die Anzahl allozierter Bits entsprechend erhöht. Dabei muß berücksichtigt werden, daß pro verwendetem Skalierungsblock eine bestimmte Anzahl Samples ausschließlich zur Speicherung der Einbettungsmarkierung verwendet wird.

Zu beachten ist, daß die innere Allokierungsschleife höchstens solange durchlaufen wird, bis genügend Bits allokiert werden konnten (Bedingung in Zeile 16). Da die Hauptschleife verlassen wird, falls sich die Anzahl allozierter Bits von einem auf den nächsten Schleifendurchlauf nicht ändert, ist die Terminierung des Algorithmus stets gewährleistet. Der in der vorgestellten Version gewählte Parametersatz (maximale absolute Einbettungsbreite, Sicherheitskonstante, etc.) hat sich in der Praxis gut bewährt.

### 6.2.5 Einbettung und Einbettungsformat

Abb. 6.3 zeigt ein Codefragment zur Realisierung einer Einbettung von Daten aus dem Bitbuffer `buf` in einen Skalierungsblock. Die wichtigsten Variablen und Funktionen haben die folgende Bedeutung:

**robust\_buffer:** Struktur zur Realisierung des Bitbuffers. Ein Zeiger auf eine Instanz hiervon ist `buf`.

**fracs:** Zeiger auf einen Skalierungsblock, hier ein Feld von Fließkommazahlen (Typ `double`). Die Variable `num_fracs` gibt die Größe dieses Blocks an.

**bit\_width:** Gibt die Einbettungsbitbreite an.

**SC\_BLOCK\_SIZE:** Globale Konstante, die die maximale Größe eines Skalierungsblocks angibt.

---

**Einbettung für einen Skalierungsblock**

```

1  int ro_buffer_to_frac(robust_buffer* buf, double* fracs,
2                          unsigned int num_fracs, unsigned int bit_width){
3
4  unsigned int i;
5  unsigned long bits[SC_BLOCK_SIZE];
6  unsigned long bits_to_read, words_to_read, bits_left, left_in_buf;
7
8  for(i=0 ; i < SC_BLOCK_SIZE ; i++) bits[i]=0;
9
10 if(num_fracs <= SIGNATURE_SIZE) return -1;
11
12 for(i=0 ; i<SIGNATURE_SIZE ; i++){          // Einbetten der Markierung
13     fracs[i] = pack_bits_to_frac(fracs[i],
14     AR_CODER(bit_width+add[i]),bit_width+AR_CODE_WIDTH);
15 }
16
17 // Im folgenden werden die Daten des Bitbuffers eingebettet
18
19 bits_to_read = bit_width*(num_fracs-SIGNATURE_SIZE);
20 left_in_buf = ro_buffer_code_left(buf);
21
22 if(bits_to_read <= left_in_buf){           // Hole Bits aus Bitbuffer
23     ro_buffer_get(buf,bit_width,num_fracs-SIGNATURE_SIZE,bits);
24 } else {
25     words_to_read = ro_buffer_code_left(buf)/bit_width;
26     ro_buffer_get(buf,bit_width,words_to_read,bits);
27     bits_left = left_in_buf - words_to_read*bit_width;
28     ro_buffer_get(buf,bits_left,1,&bits[words_to_read]);
29 }
30
31 for(i=SIGNATURE_SIZE; i < num_fracs ; i++){ // Einbetten der Daten
32     fracs[i] = pack_bits_to_frac(fracs[i],
33     AR_CODER(bits[i-SIGNATURE_SIZE]),bit_width+AR_CODE_WIDTH);
34 }
35
36 return 0;
37
38 }

```

---

Abbildung 6.3: Codefragment mit einer Realisierung einer Funktion zur Einbettung von Daten aus dem Bitbuffer in einen Skalierungsblock.

`bits_to_read`: Gibt die Anzahl der in den vorliegenden Skalierungsblock einbettbaren Bits (ohne Markierungsbits) an.

`words_to_read`: Anzahl vollständiger Worte der aktuellen Bitbreite `bit_width`, die einzubetten sind.

`bits_left`: Ist die Anzahl einzubettender Bits kein Vielfaches der Bitbreite, gibt `bits_left` deren Rest modulo `bit_width` an.

`left_in_buf`: Anzahl noch einzubettender Bits im Bitbuffer.

`bits[]`: In diesem Feld werden die einzubettenden Bits vorbereitet.

`pack_bits_to_frac()`: Funktion zur Einbettung eines Datenworts einer bestimmten Bitbreite in eine durch einen `double`-Wert repräsentierte Fließkommazahl.

`AR_CODER()`: Makro zur Auswahl einer Funktion, die eine arithmetische Codierung zur Ermöglichung einer Fehlerkorrektur durchführt. Die Breite der durch die Codierung hinzugefügten Redundanz in Bits wird durch die Konstante `AR_CODE_WIDTH` angegeben.

`add[]`: Dieses Feld gibt ein Bitmuster für die verwendete Markierung an.

`ro_buffer_get()`: Funktion zur Extraktion von Blöcken von  $n \times k$  Bits aus dem Bitbuffer in ein Feld von Ganzzahlwerten.

Die Einbettung in einen Skalierungsblock ist unterteilt in

- Einbettung der Blockmarkierung (Zeilen 12-15),
- Vorbereitung der einzubettenden Daten (Zeilen 19-29), und
- Einbettung der Nutzdaten (Zeilen 31-34).

In der Konstanten `SIGNATURE_SIZE` wird die Breite der Blockmarkierung in Datenworten angegeben. Jeder Skalierungsblock auf dem eine Einbettung vollzogen wird, erhält eine Markierung dieser Wortbreite. Die Bitbreite der Markierung pro Sample wird festgelegt durch die Einbettungsbitbreite. Die Markierung besteht hier aus Wörtern, die aus der Summe der Bitbreite und einem bestimmten Bitmuster bestehen (Zeile 14). Als Vorverarbeitung zu den Einbettungen (Zeilen 14 und 33) wird jeweils eine arithmetische Codierung der einzubettenden Worte vorgenommen. Die Breite des Codes wird der Einbettungsbreite zugeschlagen. Dies ist bei der Bitallokierung entsprechend zu berücksichtigen.

Die Vorbereitung der einzubettenden Daten (Zeilen 19-29) entspricht der Extraktion der benötigten Bits aus dem Bitbuffer. Enthält der Bitbuffer weniger Bits als in den Skalierungsblock eingebettet werden können, wird eine Sonderbehandlung (Zeilen 25-28) angewendet. Die Dateneinbettung (Zeilen 31-34) schreibt die so vorbereiteten Datenworte in die Fließkommazahlen. Diese Direkteinbettung ist abhängig von der Fließkommadarstellung und in der Funktion `pack_bits_to_frac` realisiert.

### 6.2.6 NPR-Problematik und Fehlerkorrektur

Tests zeigen, daß der durch die in MPEG verwendete NPR-Filterbank bedingte Rekonstruktionsfehler zusammen mit den sonstigen rechenwerkbedingten Arithmetikfehlern relativ groß ist. Arithmetikfehler entstehen bei den Auswertungen der Filterbanktransformationen sowie bei den Konversionen der Fließkomma- in Ganzzahlwerte. Letztere Konversion fällt nach der Filterbankrücktransformation, bei der Erzeugung von 16-Bit PCM-Werten, an.

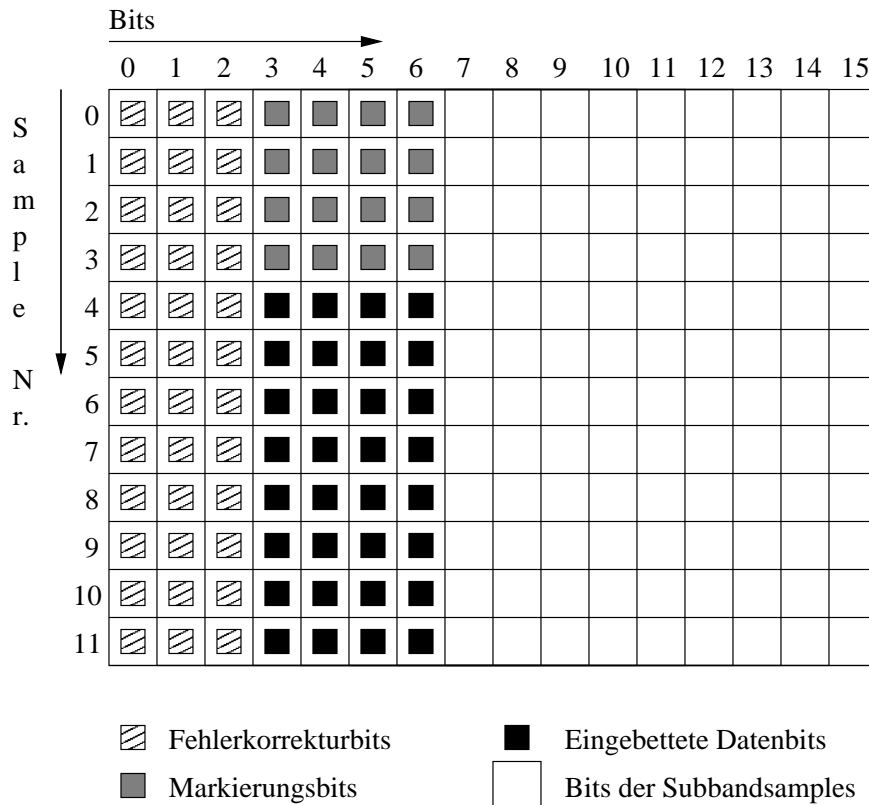


Abbildung 6.4: Bitaufteilung innerhalb eines Skalierungsblocks auf dem eine Direkteinbettung vollzogen wurde. Die reine Einbettungsbitbreite ist 4 Bit, die Bitbreite der Fehlerkorrektur 3 Bit und die Markierungsbreite 4 Samples.

Zur Vorbeugung gegen diese Fehlerarten wird als FEC (engl. *forward error correction*) eine arithmetische Codierung vom Typ  $Ax + B$ , siehe Kapitel 4, durchgeführt. Die Codierung mit  $A = 3, B = 0$  kann Fehler aus  $\{+1, -1\}$  korrigieren, die Codierung mit  $A = 8, B = 3$  solche im Intervall  $[-3 : 4]$  oder  $[-4 : 3]$ . Letztere Codierung erweist sich zur vollständigen Ausnutzung der zur Verfügung stehenden Einbettungsbitbreite als praktikabel. Der Codierungsoverhead beträgt 3 Bit pro Einbettung. In Abb. 6.4 ist die Bitaufteilung innerhalb eines Einbettungsblocks dargestellt. Bei der Einbettung wird unter der Annahme, daß sich die Subbandsamples im Wertebereich  $[-2, 2]$  befinden, eine Konversion der Fließkommazahlen in 32-Bit Ganzzahlwerte durchgeführt. Die Abbildung bezieht sich auf die 16 signifikanten Bits dieser Ganzzahlwerte.

## 6.3 Direkteinbettungscodec: Encoder

### 6.3.1 Strukturierung

Der Encoder hält sich eng an den Grobalgorithmus in Abb. 5.6. Für jeden Datenrahmen werden alle anfallenden Skalierungsblöcke in derselben Reihenfolge, die auch im Decoder verwendet wurde, abgearbeitet. Dies ist notwendig, um die richtige Reihenfolge der eingebetteten Daten herzustellen, da die hier verwendeten Blöcke keine Nummerierung enthalten. Für jeden Skalierungsblock wird unabhängig eine Einbettungsdetektierung mit anschließender potentieller Decodierung eingebetteter Daten vorgenommen. Unter Verwendung einer Checksumme wird nach Abarbeitung aller Sklierungsblöcke getestet, ob die Anzahl der extrahierten Bits mit der Checksumme übereinstimmt. Ist dies der Fall, werden die Encoderparameter in die entsprechenden Datenstrukturen, also in `allocation`, `scfsi` und `scalefactor` übertragen und die Encodierung wie gewöhnlich fortgesetzt.

### 6.3.2 Einbettungsdetektion und -decodierung

Abb. 6.5 zeigt die zur Einbettungsdetektierung und -decodierung verwendete C-Routine. Die Variablen- und Funktionsbezeichnungen entsprechen denen im Einbettungsalgorithmus in Abb. 6.3. Neu sind hier

`counter`: Zähler für die Anzahl der mit einer gültigen Markierung übereinstimmenden Worte bei Annahme einer festen Bitbreite (innerhalb eines Durchlaufs von 18-21).

`AR_DECODER()`: Entsprechende Decodierungsfunktion zu `AR_CODER`.

`ro_buffer_insert()`: Funktion zum Schreiben von Bitblöcken beliebiger Dimensionen  $n \times k$  in den Bitbuffer.

Der Algorithmus besteht aus zwei Hauptteilen, der Detektion möglicher Einbettungsmarkierungen in den Zeilen 12-25 und der Extraktion der Nutzdaten in den Zeilen 32-36. Innerhalb der Detektionsroutine werden alle möglichen Einbettungsbitbreiten mit zugehörigen Markierungen nacheinander getestet. Die Decodierung einer einzelnen Markierung findet dabei in Zeile 18 statt. Die niedrigste Bitbreite, für die alle Markierungen decodiert werden konnten, wird als detektiert angenommen (Zeilen 20-24). Die decodierten Daten inklusive den Markierungsdaten werden im Feld `bits[]` abgelegt. Die Nutzdaten werden aus diesem Feld in den Bitbuffer geschrieben (Zeile 35), aus dem sie in einem anschließenden Verarbeitungsschritt, in Umkehrung des Vorgangs im Decoder, in die Datenstrukturen für die Encoderparameter geschrieben werden.

### 6.3.3 Hybridencoder

Abschließend wird besprochen, wie die bei der Detektion und Decodierung der in einen Datenrahmen eingebetteten Daten möglichen Probleme behandelt werden. Bei der Detektion bzw. Decodierung können folgende Fälle auftreten:

1. Detektion und Decodierung sind erfolgreich.
2. Es können keine Einbettungen detektiert werden.

---

**Einbettungsdetektierung und -decodierung eines Skalierungsblocks**

```

1 int frac_to_ro_buffer(robust_buffer* buf, double* fracs,
2                       unsigned int num_fracs, FILE* text_file){
3
4   unsigned int i,j,counter,max_counter;
5   unsigned long bits[SC_BLOCK_SIZE];
6   int bit_width = 0; long difference; static int ct=0;
7   unsigned long decoded_values[SC_BLOCK_SIZE];
8
9   if(num_fracs <= SIGNATURE_SIZE) return -1; // Block zu klein
10  max_counter = 0;
11
12  for(j=3+AR_CODE_WIDTH ; j<MAX_BIT_WIDTH ; j++){
13                                     // Teste moegliche
14   counter = 0;                       // Einbettungsbitbreiten
15   for(i=0 ; i<SIGNATURE_SIZE ; i++){
16
17     bits[i] = get_bits_from_frac(fracs[i],j);
18     if(AR_DECODER(bits[i],j)==j-AR_CODE_WIDTH+add[i]) counter++;
19   }
20   if((counter==SIGNATURE_SIZE) && (counter > max_counter)){
21                                     // wurden alle Teile der
22     max_counter = counter;           // Markierung erkannt?
23     bit_width = j;
24   }
25 }                                     // Ende Signaturdetektion
26
27 for(i=0 ; i<SIGNATURE_SIZE ; i++)    // Markierung wird extrahiert
28   bits[i] = bit_width-AR_CODE_WIDTH+add[i];
29
30 if(!bit_width) return 0;             // Keine Einbettung erkannt
31
32 for(i=SIGNATURE_SIZE; i < num_fracs ; i++){ // Decodierung der Daten
33
34   bits[i] = AR_DECODER(get_bits_from_frac(fracs[i],bit_width),bit_width);
35   ro_buffer_insert(buf,MAX(0,bit_width-AR_CODE_WIDTH),1,&bits[i]);
36 }
37 return bit_width-AR_CODE_WIDTH;
38 }

```

---

Abbildung 6.5: Codefragment mit einer Realisierung einer Funktion zur Einbettungsdetektierung und -decodierung eines Skalierungsblocks.

3. Teile einer detektierten Einbettung sind fehlerhaft bzw. es konnten nicht alle Skalierungsblöcke mit vorhandener Einbettung innerhalb eines Datenrahmens detektiert werden.

Aufgrund der beschriebenen Realisierung sind die ersten zwei Fälle seitens des Encoders erkennbar, dabei der erste mit einer gewissen Wahrscheinlichkeit. Der dritte Fall ist hingegen nicht ohne weitere Modifikationen erkennbar. Im ersten Fall besteht das „Restrisiko“ in einer Fehldetektion, die hier in Kauf genommen wird. Im zweiten Fall schaltet der Encoder für den betreffenden Datenrahmen auf die normale Encodierung mit Hilfe des psychoakustischen Modells um. Diese Lösung stellt auch für den dritten Fall die einfachste Lösung dar. Sind die Daten nicht ganz verloren, sondern nur unvollständig, könnte dies weiterhin ausgenutzt werden. Dies scheitert jedoch daran, daß bei dem gewählten Verfahren i. a. nicht klar ist, *welche* Daten fehlen. Um das Auftreten des dritten Falles zumindest mit hoher Wahrscheinlichkeit zu erkennen, kann jeweils die Gesamtanzahl der eingebetteten Bits pro Datenrahmen zusätzlich als Checksumme im ersten Einbettungsblock übertragen werden. Wird dieser erste Block korrekt detektiert und decodiert, kann mit hoher Wahrscheinlichkeit erkannt werden, ob auch die restlichen Daten korrekt übertragen werden konnten.

## 6.4 Fehlerregionencodec

Zur Erweiterung des in den letzten Abschnitten beschriebenen Codecs auf die Verwendung von Fehlerregionen sind im wesentlichen an vier Stellen der Implementation Änderungen und Ergänzungen vorzunehmen:

1. Schon im Encoder müssen die Einbettungsstellen bekannt sein. Daher muß dort ebenfalls der Einbettungsbitallokieralgorithmus ausgeführt werden. Für die Samples der allokierten Skalierungsblöcke werden, o.B.d.A. in aufsteigender Reihenfolge der Blöcke, Fehlerregionen bestimmt. Es sind hier zwei Fälle zu unterscheiden (siehe Fallunterscheidung 3 in Abb. 5.8):
  - (a) Die vorliegenden Daten enthalten bereits eingebettete Bits. Dann sind die Fehlerregionen aus den Einbettungsregionen ablesbar.
  - (b) Es wird eine normale oder Erstbitallokierung durchgeführt. Dann werden die Fehlerregionen aus den Rekonstruktionswerten der Quantisierung bestimmt.
2. Die Fehlerregionen müssen im MPEG-Bitstrom an den Decoder übertragen werden.
3. Die decoderseitige Einbettung muß entsprechend den Fehlerregionen stattfinden. Aufgrund des Rekonstruktionsfehlers der Filterbank muß die Einbettung fehlertolerant sein.
4. Encoderseitig muß die Einbettungsdetektierung und -decodierung angepaßt werden.

### 6.4.1 Erweiterung des MPEG-Bitdatenstroms

Die Übertragung der Fehlerregionen an den Decoder erfordert eine Erhöhung der Bitrate des MPEG-Datenstroms. Da unter Layer II nur feste Bitraten pro Datenrahmen zulässig sind, ist ein Maximalbedarf an Fehlerregioneninformation in Bits abzuschätzen und die Bitrate entsprechend zu erhöhen.

Eine Erhöhung der Bitrate geschieht normalerweise durch die Wahl eines entsprechenden Werts in der MPEG-Variablen `bitrate_index`. Da die Verwendung von Bitallokierungstabellen jedoch von der Wahl der Bitrate in `bitrate_index` abhängt, muß in diesem Fall zwischen Kompressionsbitrate und Datenstrombitrate unterschieden werden. Die MPEG-konforme Methode wäre, das sogenannte *freie Format* (`bitrate_index=0`) zu wählen, das eine beliebige aber feste Bitrate, die unter der Maximalbitrate pro Layer liegen muß, erlaubt. Da die verwendeten MPEG-Player (`Winamp`, `MaPlay`) sowie der verwendete Quellcode jedoch diese Möglichkeit sämtlich nicht unterstützen, wurde eine andere Methode gewählt. Sowohl die obigen MPEG-Player als auch der im Quellcode vorliegende Decoder orientieren sich bei der Feststellung der Rahmengröße im Bitstrom lediglich an den Positionen der Synchronisationsbits (12 aufeinanderfolgende gesetzte Bits). Somit ist es ausreichend, die anfallenden Zusatzdaten einfach ans Ende der Rahmendaten anzuhängen. Es ist lediglich dafür Sorge zu tragen, daß dabei keine 12 aufeinanderfolgenden gesetzten Bits vorkommen und daß die konstante Bitrate pro Rahmen eingehalten wird.

Zur Verhinderung von Pseudo-Synchronisationsworten wurde die Funktionalität des MPEG-Bitstreams (innerhalb des Quellcodes in der Datei `common.c`) dahingehend erweitert, daß beim Schreiben der Zusatzinformationen jedem elften gesetzten Bits zwingend ein nicht gesetztes Bit folgt (Funktionen `getbits_sync_clear()` und `put_bits_sync_clear()`). Das mögliche Vorkommen dieser Zusatzbits ist in die Berechnung des Gesamtbedarfs einzubeziehen.

Der Gesamtbitbedarf läßt sich nach oben wie folgt abschätzen: Der maximale Bedarf  $b$  an Bits für die einzubettenden Encoderparameter läßt sich wie in 6.1.2 beschrieben abschätzen. Geht man weiterhin von einer minimalen Einbettungskapazität von  $n$  Bits Nutzdaten pro Skalierungsblock aus, so ist die Anzahl benötigter Skalierungsblöcke  $s(b, n) = \lceil b/n \rceil$ , die Anzahl Fehlerregionenbits bei Skalareinbettung somit  $s(b, n) \cdot 12$ . Wird eine Datenratenerhöhung von 12/11 zur Vermeidung von Pseudo-Synchronisationsworten angenommen, erhält man eine Bitratenvergrößerung von  $\lceil s(b, n) \cdot 144/11 \rceil$ . Im obigen Beispiel mit einem Bitbedarf von maximal  $b = 628$  Bits Encoderinformation pro Rahmen und Kanal, d.h.  $48000/1152 = 41,66\dots$  Rahmen pro Minute und somit 32750 Bits pro Sekunde zusätzlich, ergibt sich eine Senkung der Gesamtbitrate von 1 : 8 auf 1 : 6,71. Dabei wurde  $n = 21 = 3 \cdot 7$  zugrundegelegt, also eine minimale Einbettungsbreite von 3 Bit und eine Signaturbreite von 5 Worten.

Diese Schätzung ist allerdings sehr konservativ, da die Einbettungsbitbreite oft höher ist, man somit mit weniger Skalierungsblöcken auskommt, die Markierungsbreite erniedrigt werden kann und auch die Annahme des 12/11 worst-case Overhead sehr unrealistisch ist. Nimmt man beispielsweise die Verwendung von 15 Skalierungsblöcken zur Einbettung an, sinkt die Bitrate nur auf 1 : 7,23. Eine Verringerung des nötigen Overheads für die Fehlerkorrektur oder eine Verwendung von vektororientierten Einbettungsmethoden mit höherer Kapazität pro Bit können für eine weitere Erhöhung der Bitrate sorgen.

#### 6.4.2 Einbettung, Detektion und Extraktion

Die Einbettung erfolgt unter Verwendung einer Fehlerkorrektur wie in Kapitel 4 beschrieben. Dabei kann die zur Fehlerkorrektur hinzugefügte Redundanz in derselben Größenordnung wie beim Direkteinbettungscodec, also fehlerkorrigierend für Abweichungen um  $\pm 3$  bezüglich der Größenordnung der verwendeten Skalierung, gewählt werden. Ein Vorteil dieser Einbettungsmethode ist der Wegfall der bei der Direkteinbettung notwendigen Konversion zwischen Fließkomma- und Ganzzahldarstellung und zurück.

Als ein wenig problematisch in Zusammenhang mit einer Realisierung in einem MPEG-Codec

stellt sich die Detektion einer Einbettungsmarkierung heraus. Ist nämlich a priori nicht klar (wie z.B. in unserem obigen Direkteinbettungsbeispiel), welche Bitrate zur Einbettung verwendet wurde, muß der encoderseitige Detektor eines Fehlerregionencodex eine relativ große Anzahl möglicher Einbettungskonstellationen durchlaufen. Dies liegt daran, daß es bei dieser Art der Einbettung auf die Rekonstruktionswerte der Rückquantisierung ankommt, die wiederum von der gewählten Quantisierungsstufe *und* der jeweiligen Skalierung abhängen. Da allerdings zahlreiche Kombinationen von Allokierungs- und Skalierungsparametern von vorneherein eine Einbettung ausschließen, kann so schon eine große Anzahl Kandidaten ausgeschlossen werden.

Zwei Möglichkeiten zur Umgehung dieses Problems sind

- Verwendung der in Kapitel 5 vorgeschlagenen Bitreservoirtechnik. Stehen nämlich schon von vorneherein die Encoderparameter für einen Rahmen fest, so kann der Encoder bzw. Detektor mit Hilfe des Algorithmus zur Einbettungsbitallokierung die Einbettungsstellen und die zugehörigen Rekonstruktionswerte des Quantisierers bestimmen.
- Verwendung einer Hybridtechnik, wobei für einen Einbettungsblock Direkteinbettung und für alle anderen Fehlerregioneneinbettung verwendet wird. Bei dieser Methode enthält der Direkteinbettungsblock Informationen über die Position eines oder aller zu decodierenden Einbettungsregionenblöcke.

Der Vorteil letzterer Methode liegt in ihrer lokalen Beschränktheit auf einen Datenrahmen. Dies macht die Methode u.a. robuster gegen mögliche Rahmenverluste. Ein Nachteil sind die potentiell bei jeder Direkteinbettung möglichen größeren Rekonstruktionsfehler.

Ein auf MPEG-1 Layer II basierender Fehlerregionencodex wird in [12] aufbauend auf den obigen Ausführungen entwickelt.

## 6.5 Codec-Tests

### 6.5.1 Testkriterien und Anforderungen

Im Hinblick auf die theoretischen Eigenschaften eines Fehlerregionencodex stellt sich die Frage, in welcher Form die Funktionsweise der hier entwickelten Codex überprüft werden kann und muß. Da ein Fehlerregionencodex eine Einbettung unter Einhaltung des maximalen perceptuell zulässigen Rekonstruktionsfehlers vornehmen kann, hängt es lediglich von der Funktionalität der psychoakustischen Analyse und einer erfolgreichen Einbettung der Encoderparameter im Sinne von 5.3.3 ab, ob die Qualität der ersten Generation in allen weiteren reproduziert wird. Nimmt man sinnvollerweise die Funktionsfähigkeit des psychoakustischen Modells an, bleiben die drei Bedingungen aus 5.3.3, von denen die Fehldetektion durch eine geeignet redundante Einbettungsmarkierung und die Arithmetik- und Rekonstruktionsfehler durch fehlertolerante Einbettung prinzipiell ausgeschlossen werden können. Der durch Einbettungsmarkierungen und fehlertolerante Einbettung verursachte zusätzliche Bitbedarf ist jedoch zu untersuchen.

Wichtiger ist jedoch die Frage nach genügender Einbettungskapazität. Dies ist natürlich auch für die Direkteinbettung eine fundamentale Fragestellung. Hier sind zu untersuchen:

- An wievielen und welchen Stellen kann eingebettet werden?
- Von welchen Faktoren hängt die Einbettungskapazität ab?

- Sind Bitreservoirtechniken sinnvoll einsetzbar, steht also manchmal sehr viel mehr Kapazität zur Verfügung als für einen Datenrahmen alleine benötigt wird?

Im Zusammenhang mit Direkteinbettungscodecs sind weiterhin subjektive Überprüfungen und Befragungen von Testpersonen zur Verifikation der Codecfunktionalität sinnvoll, da hier entsprechend der Einbettungsart Qualitätsbeeinträchtigungen entstehen können. Da sich herausstellt, daß nicht immer genügend Einbettungskapazität zur Verfügung steht und somit nicht für alle Datenrahmen Encoderparameter übertragen werden können, sind solche Tests zur Überprüfung der Codecfunktionalität an den Übergangsstellen und der möglichen Degeneration des Audiomaterials an den Stellen, an denen dann das Standardverfahren angewendet wird, wichtig.

Bei der Evaluation des Layer II-Direkteinbettungscodecs durch Testhörer sollten die zu folgenden Fragestellungen herausgearbeiteten Resultate tendenziell überprüft werden:

- In wieweit wird die Qualität höherer Generationen beeinträchtigt? Wie stellen sich wahrnehmbare oder störende Artefakte dar?
- Wie unterscheiden sich höhere Generationen qualitativ von der ersten Generation?
- Sind verschiedene höhere Generationen (z.B. die vierte von der neunten oder die fünfte von der 25.) voneinander unterscheidbar?
- Wie stellt sich der Qualitätsunterschied zwischen höheren Generationen des Direkteinbettungscodecs und des Standardcodecs dar?

Als Referenzstücke wurde stets Audiomaterial der ersten Generation verwendet, da das Verfahren prinzipiell nur auf den Erhalt der Qualität der ersten Generation, und nicht der des Originals, abzielt.

### 6.5.2 Testmaterial und -konfiguration

Für die Tests wurde Audiomaterial aus einer Sammlung von ca. 150 Stücken in CD-Qualität bei 16 Bit linearer Quantisierung, Stereo und 44.1 kHz Abtastrate verwendet. Das Material besteht aus

- Stücken der EBU/SQUAM Test-CD [58]. Es wurden 110 Audiostücke verschiedenster Längen, einteilbar in die Kategorien
  - künstliche Signale,
  - Einzelinstrumente,
  - Vokalstücke,
  - Sprache,
  - Soloinstrumente,
  - Orchestralstücke und
  - Popmusik.

ausgewählt. Die Test-CD enthält zusätzlich Vorschläge zur Verwendung der verschiedenen Stücke mit bestimmten Klassen von Audioanwendungen, u.a. auch Kompression. Weiterhin sind exakte Aussteuerungshinweise und Informationen zu spezifischen Aufnahmebedingungen, z.B. über die Art des Aufnahmeraums, enthalten.

- einer Sammlung von Teststücken, die innerhalb der Arbeitsgruppe aus Erfahrungswerten im Zusammenhang mit Audiokompressionsverfahren zusammengestellt wurde. Die Sammlung umfaßt Stücke der Kategorien
  - Orchester,
  - Synthesizerstücke,
  - Popmusik,
  - Gitarrenstücke (Western-Gitarre).

Da der verwendete MPEG Layer II-Codec bei einigen Stücken zu großer Lautstärke decoderseitig ein digitales Übersteuern verursacht, wurden alle verwendeten Stücke auf den halben möglichen Amplitudenbereich ausgesteuert. Somit wird das Übersteuern vermieden. Die Ursache des Übersteuerns liegt darin, daß die inverse Filterbank auf der Menge der requantisierten Subbandsignale nicht genau auf den 16-Bit PCM-Bereich ( $[-32768 : 32767]$ ) beschränkt ist. Das dadurch verursachte Clipping stört zwar die subjektive Qualität des Audiostücks im allgemeinen nicht, zerstört aber unter Umständen die eingebetteten Informationen. Zu bemerken ist, daß die meisten Stücke der EBU/SQUAM Test-CD bereits geeignet ausgesteuert waren.

### 6.5.3 Ausgewählte Teststücke

Zur Abdeckung obiger Audiokategorien wurden die in Tabelle 6.6 aufgeführten 20 Teststücke verwendet. Für die Hörtests wurden hieraus wiederum zehn Stücke ausgewählt. Diese sind in der rechten Spalte der Tabelle gesondert markiert. Während der Entwicklungsphase wurde der Codec auf einer wesentlich größeren Teilmenge des im letzten Abschnitt angegebenen Materials getestet. Es wurde ebenfalls darauf geachtet, die Audiosignale möglichst vielseitig im Hinblick auf verschiedene Charakteristika auszuwählen:

- Instrumentalstücke, Vokalstücke, Instrumentalstücke mit Gesang, sowie gesprochene Passagen.
- Einzelinstrumente verschiedener Klassen, dabei sehr stimmhafte (z.B. Piano oder Violine) und eher percussive (z.B. Kastagnetten oder E-Baß).
- Im Zusammenhang mit Kompressionsverfahren als kritisch eingestufte Stücke (z.B. Kastagnetten, Glockenspiel, Sopranstück mit Begleitung 096sophr).
- Stücke mit Spezialeffekten, z.B. Stereoeffekte und stark unterschiedlich belegte Kanäle. Die Live-Aufnahme 122\_1 ist aufgrund der Hintergrundgeräusche ein für Codectests eher untypisches Teststück.
- Synthetische Musik, z.B. reine Synthesizermusik (s3\_2) und Musik mit synthetischen Teilen (104abba).
- Moderne und klassische Musik.

### 6.5.4 Einbettungskapazitäten

Die Einbettungskapazitäten werden zunächst anhand eines relativ „einfachen“ Audiostücks, einem Schlag einer einzelnen Glocke mit Nachhall, untersucht. Das Stück zeichnet sich durch

Nr.	Beschreibung	Datei	Test
1	Violine, Tonleiter	010violini	✓
2	Popmusik, E-Klavier und Gitarre (Abba)	104abba	✓
3	Kastagnetten	044kasta	✓
4	Weiblicher Sprecher, deutsch	088speec	✓
5	Männlicher Sprecher, französisch	087speec	✓
6	Glockenspiel (Melodie)	062gspie	✓
7	Blasinstrumente, Klassik	101wind	✓
8	Orchester, Tutti	ct7_2	✓
9	Pop, Gesang und Percussion (Jackson)	v11_2	✓
10	Percussion (fester Rhythmus)	d9_1	✓
11	Saxophon (Melodie)	031sax2	
12	Trommelwirbel	046sdrsn	
13	Klavier (Melodie)	069piano	
14	Sopran und Orchester	096sopr	
15	E-Baß (slapping)	b2_1	
16	Glocken	bells	
17	Trompete und Percussion	d12_2	
18	E-Gitarre (Rhythmus), Vokalanteile	g15_2	
19	Yesterday (live)	122_1	
20	Synthesizer (Melodie und Rhythmus)	s3_2	
21	Blues (1 Kanal vokal, 1 Kanal Begleitung)	vg15_1	
22	Percussion und Effekten (mit Kanalwechseln)	d9_3	

Abbildung 6.6: Teststücke für die Codec-Evaluation.

zum Zeitpunkt des Schlages starke Energiewerte aus, die während der Nachhallphase schnell abfallen. Die Wellenform des ca. 4 Sekunden langen Stücks ist in Abb. 6.8 unten dargestellt. Abb. 6.7 zeigt die am Direkteinbettungscodec anfallenden Einbettungskapazitäten. Dabei zeigt die fast waagerechte, durchgezogene Linie die benötigte Einbettungskapazität pro Rahmen (in Bits) an. Die gestrichelte Linie zeigt die verfügbare Bruttoeinbettungskapazität an. Hierbei wurden die im obigen Allokierungsalgorithmus verwendeten Kriterien einer Mindestbitbreite von 6 Bits pro Skalierungsblock berücksichtigt. Die gepunktete durchgezogene Linie zeigt demgegenüber die Nettokapazität, die aus der Bruttokapazität unter Abzug des für die Markierung und die Fehlerkorrektur benötigten Platzes entsteht. Eine Einbettung erfolgt somit genau dann, wenn die durchgezogene Linie sich unterhalb oder auf gleicher Höhe mit der gepunkteten durchgezogenen Linie befindet. Im vorliegenden Beispiel ist also eine Einbettung nur während der 25 Frames, während derer der Glockenschlag stattfindet, möglich. Ebenfalls ist jedoch die sehr große Differenz zwischen Brutto- und Nettobitkapazität erkennbar. Der obere Teil von Abb. 6.8 zeigt für alle Frames des Stücks diejenigen Skalierungsbänder, die überhaupt eine Einbettungskapazität zur Verfügung stellen. Die Größe der Kapazität ist dabei als Graustufe proportional zur Intensität dargestellt. Da hier auch Bänder dargestellt werden, die weniger als die geforderte Kapazität von 6 Bits pro Sample zur Verfügung stellen, erscheint der Einbettungsbereich hier breiter als in Abb. 6.7. Wird die Anforderung der 6 Bits, z.B. durch eine Verringerung der Fehlerkorrekturbreite, gesenkt, sind hier noch Ein-

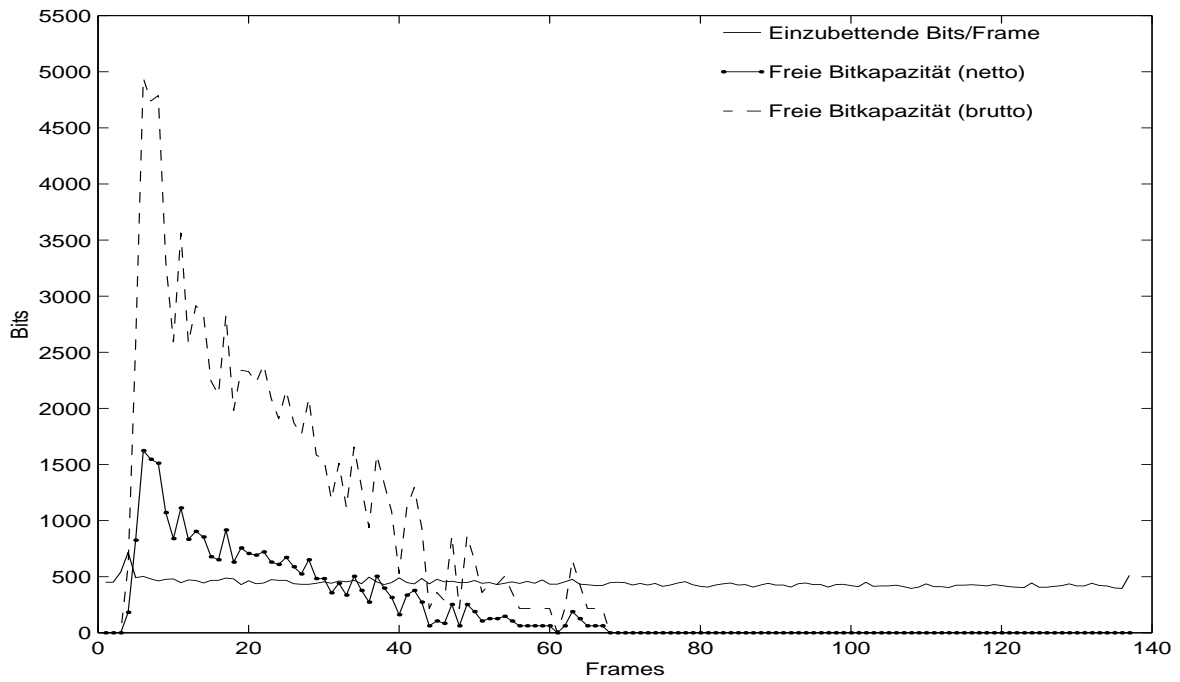


Abbildung 6.7: Einbettungskapazitäten für das Glockenschlagstück bells.

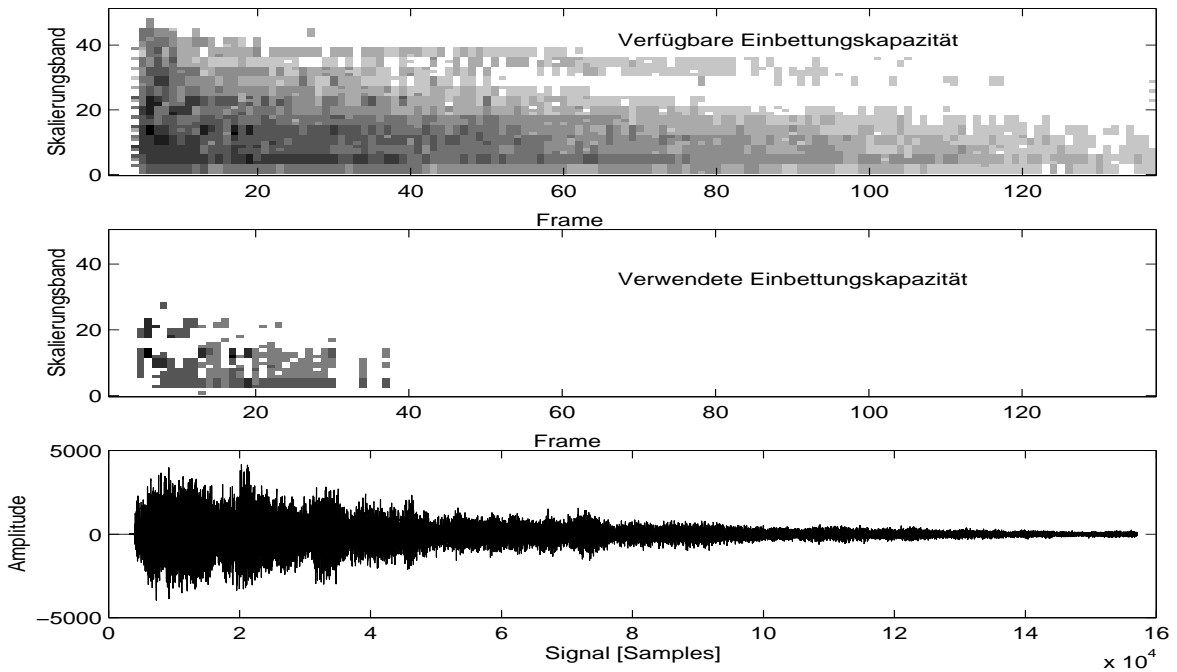


Abbildung 6.8: Verfügbare und verwendete Einbettungspositionen für das Glockenschlagstück bells

bettungskapazitäten zu gewinnen. Der mittlere Teil der Abbildung zeigt die tatsächlich zur Einbettung verwendeten Stellen.

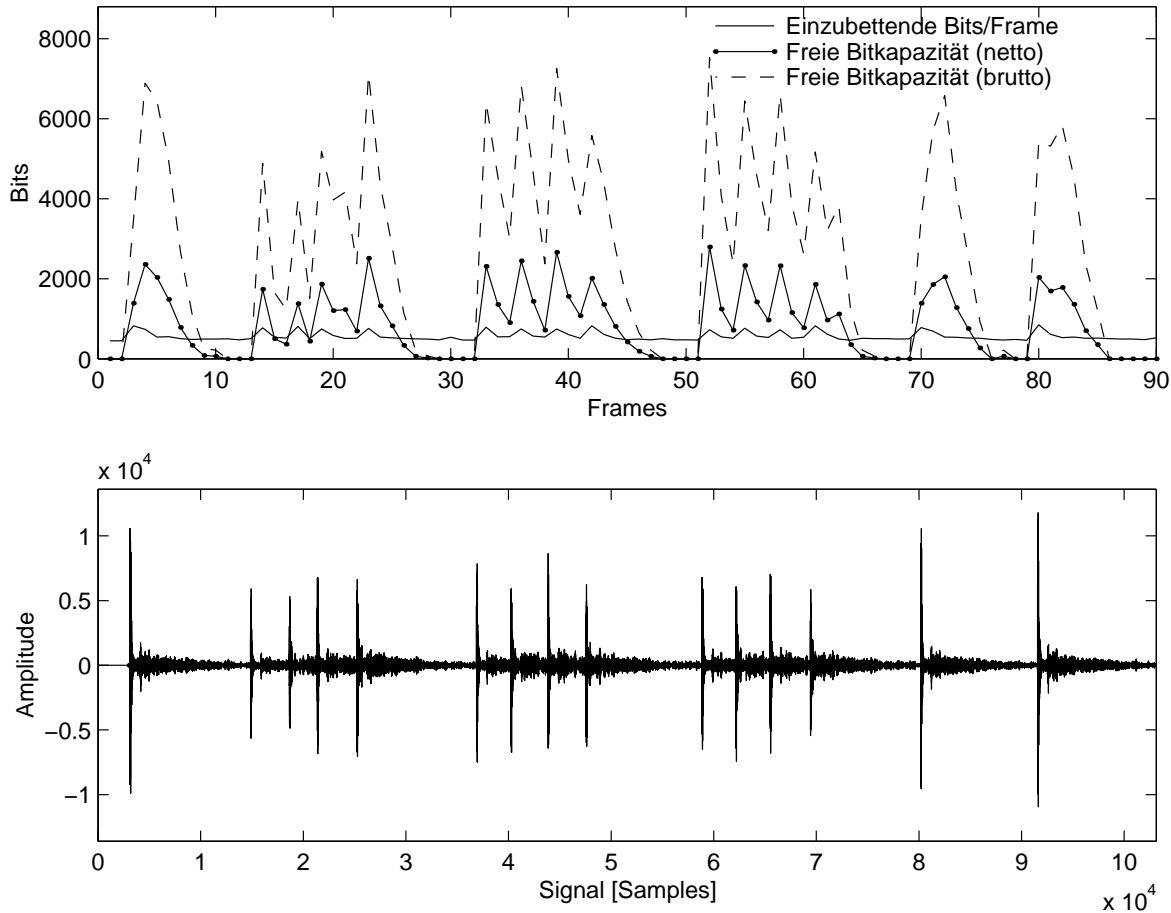


Abbildung 6.9: Einbettungskapazitäten und Signaldarstellung für das Kastagnettenstück.

Abb. 6.9 zeigt die Einbettungskapazitäten und Signaldarstellung für das Kastagnettenstück 044kasta. Hier ist besonders gut die Abhängigkeit der Einbettungskapazität von den Anschlägen der Kastagnetten zu erkennen. Auch ist die Bruttokapazität an den Einbettungsstellen deutlich höher als die Nettokapazität. Abb. 6.10 zeigt entsprechendes für den männlichen Sprecher (Signalausschnitt). Hier ist sehr deutlich zu sehen, daß die nichttonalen Passagen (z.B. Rahmen 21-26 oder 31-34) eine größere Einbettungskapazität zur Verfügung stellen, als die tonalen (z.B. Rahmen 18-20 oder 27-30). Dies entspricht auch genau dem Berechnungsmodell der Einbettungskapazitäten in Zusammenhang mit den Bemerkungen aus Kapitel 3 über die größere Maskierungswirkung bei nichttonalen Signalkomponenten. Hier liegt auch ein für das Funktionieren des Gesamtverfahrens fundamentaler Grund: Die Degeneration tritt wie festgestellt besonders stark im Zusammenhang mit nichttonalen und energiereichen Signalanteilen zutage. Da diese Komponenten aber immer eine sehr große Einbettungskapazität zur Verfügung stellen, kann der Qualitätsverlust durch das Einbettungsverfahren an diesen Stellen vermieden werden.

Abb. 6.11 zeigt die jeweiligen Kapazitäten an einem längeren Popstück (ca. 15 Sekunden). Klar

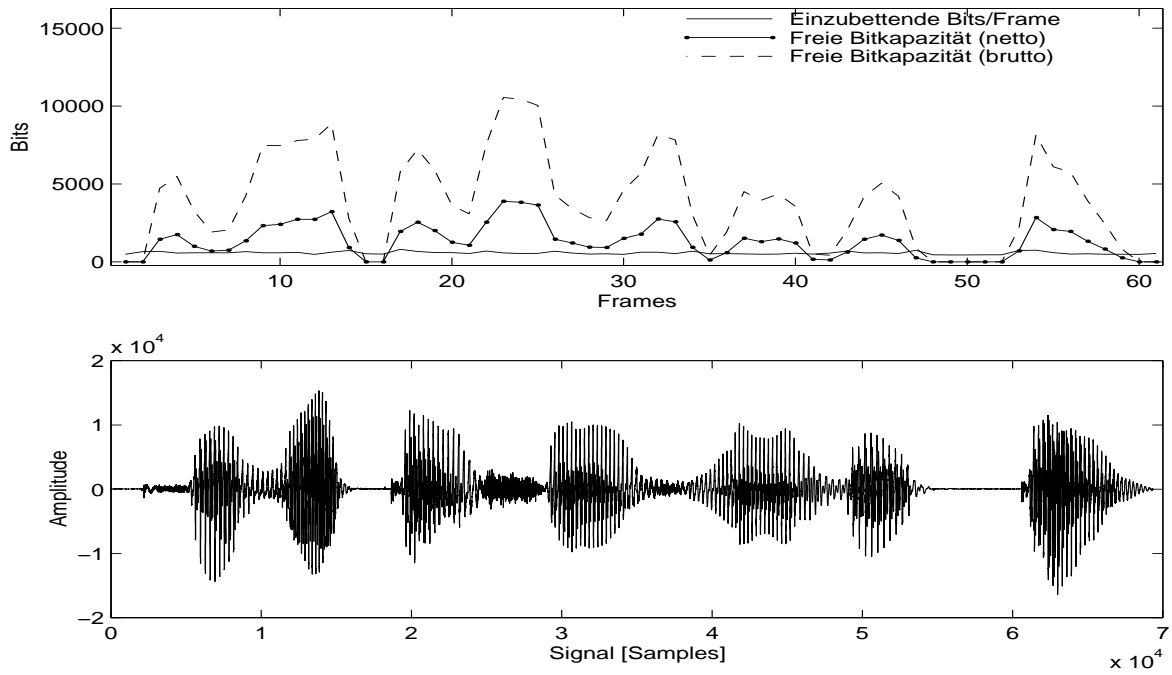


Abbildung 6.10: Einbettungskapazitäten und Signaldarstellung für das Sprachstück 087spec.

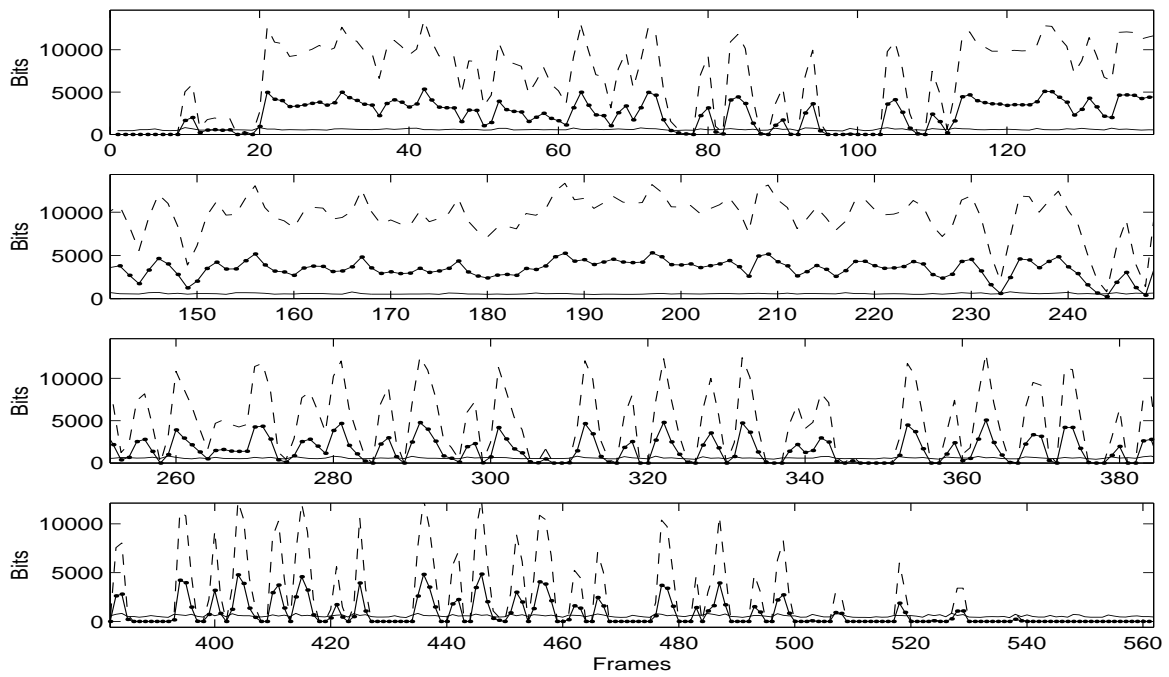


Abbildung 6.11: Einbettungskapazitäten für das Popstück 105rabit.

zu sehen ist, daß besonders in energiereichen Passagen ein Übermaß an Einbettungskapazität zur Verfügung steht. Da in diesem Beispiel nur an wenigen Stellen nicht eingebettet werden kann, wäre eine Bitreservoirtechnik sinnvoll anwendbar.

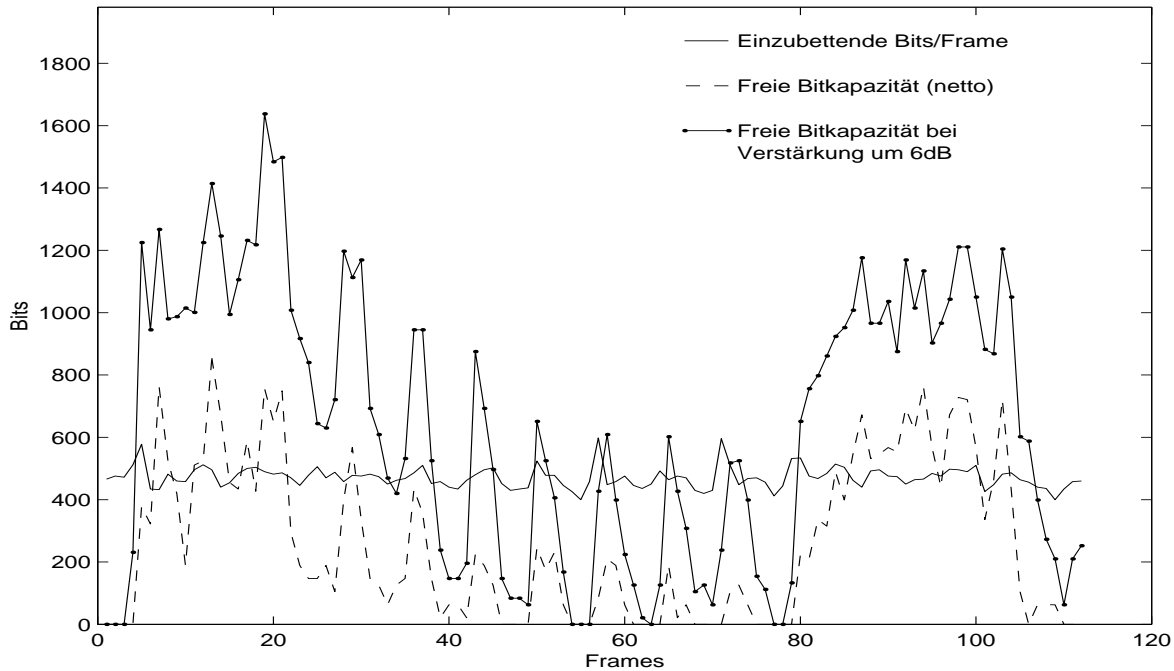


Abbildung 6.12: Vergleich der Nettokapazitäten bei Verdopplung der Lautstärke für das Sopranstück 096sophr.

Im Hinblick auf Dynamikeigenschaften ist die Untersuchung der Abhängigkeit der Einbettungskapazität von der Lautstärke des Signals sehr wichtig. Hierbei tritt für sehr geringe Lautstärken wieder der Vorteil des MPEG-Verfahrens in den Vordergrund, sich durch geeignete Skalierungsfaktoren, und damit einer entsprechend höheren Darstellungsgenauigkeit in der Bitallokierung, an diese Situation zu adaptieren. Abb. 6.12 zeigt einen Vergleich der Nettokapazitäten für einen Ausschnitt des Sopranstücks 096sophr, wobei die gestrichelte Linie die Kapazität bei normaler Lautstärke und die gepunktete durchgezogene Linie die Kapazität bei einer um 6 dB höheren Lautstärke (Verstärkungsfaktor 2) angibt. Bis zu Rahmen 20 und ab Rahmen 80 besteht das Stück aus Gesang mit Orchester, dazwischen befindet sich ein Koloraturteil des Soprans. In diesem Zusammenhang muß natürlich auch auf den Einbettungsbitbedarf pro Rahmen geachtet werden. Abb. 6.13 zeigt die Entwicklung des mittleren Bitbedarfs pro Rahmen von Verstärkungsfaktoren 0.25 (-12 dB) bis 4 (+12 dB). Es ist ein leicht logarithmisches Ansteigen des Bedarfs erkennbar. Da sich jedoch der Absolutbedarf lediglich um ca. 55 Bits pro Rahmen erhöht, ist dieser Anstieg im Verhältnis zur viel schneller steigenden Kapazität sehr gering.

### 6.5.5 Testläufe und Resultate

Für alle in Tabelle 6.6 aufgeführten Teststücke wurden für folgende Codiervorgänge die ersten 25 Generationen erzeugt:

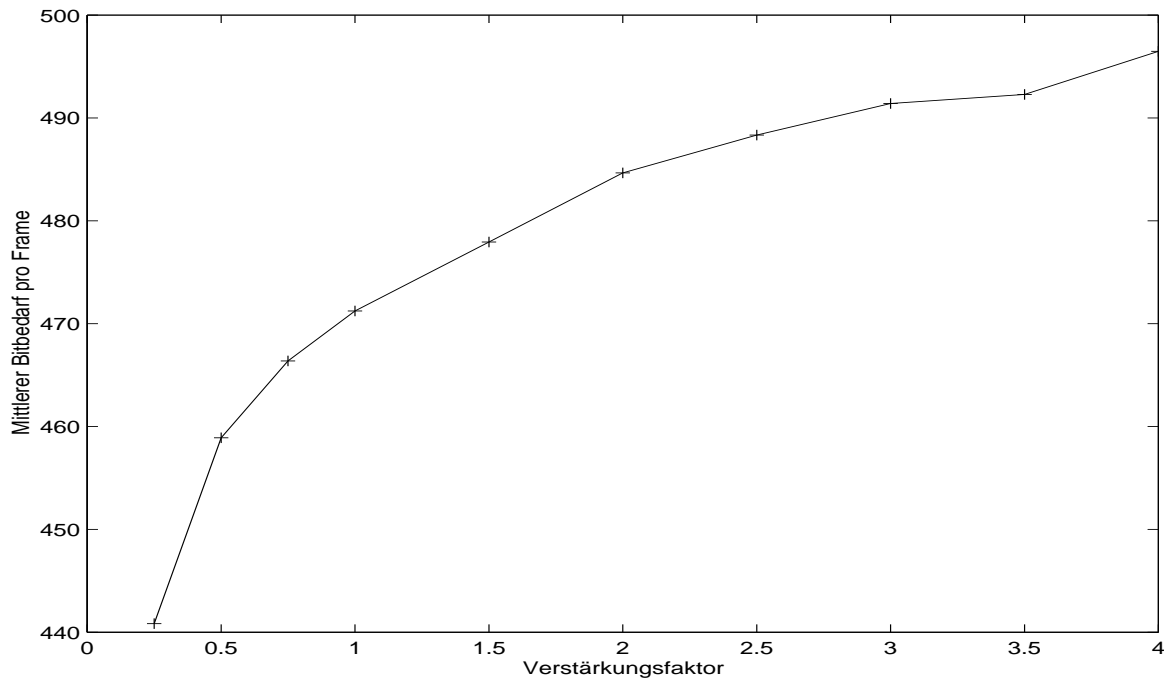


Abbildung 6.13: Entwicklung des Einbettungsbitbedarfs als Mittelwert über alle Rahmen des Sopranstücks 096sopr bei einer Veränderung der Lautstärke im Bereich  $-12\text{dB}$  bis  $+12\text{dB}$ .

- Direkteinbettungscodec zur Vermeidung von Generationseffekten,
- Standardcodec mit synchronisierender Translation um 481 Samples zwischen den Codecstufen (siehe Kapitel 3),
- Standardcodec ohne synchronisierende Translationen zwischen den Codecstufen (siehe Kapitel 3),
- Standardcodec mit Translationen zufällig gewählter Längen zwischen den Codecstufen (siehe Kapitel 3).

Insgesamt ergibt dies  $20 \cdot 25 \cdot 4 = 2000$  Audiostücke. Von den meisten der beobachteten Stücke wurden zusätzlich zur Überprüfung des Langzeitverhaltens die ersten 50 Generationen mit dem Direkteinbettungscodec erzeugt.

Zur Beurteilung der Resultate des Direkteinbettungscodecs wurden, wie im Falle der Beurteilung der Generationseffekte in Kapitel 3, neben ausführlichen Hörtests mit verschiedenen Abhörausstattungen auch wieder Mitglieder der Arbeitsgruppe, Diplomanden sowie Teilnehmer der Projektgruppen nach ihren Höreindrücken befragt.

Die Untersuchung zielte auf die Beurteilung folgender Punkte ab:

- Unterschied bzw. möglicher Qualitätsverlust zwischen Original und erster Generation zwecks Einschätzung des Grundverlustes des Layer II-Codecs mit einer Bitrate von 128 kbps.
- Unterschied zwischen der ersten Generation und der  $n$ -ten Generation für verschiedene  $n$ , insbesondere kleine  $n$ , wie z.B. 2,3 oder 5, und große  $n$ , etwa 10, 15, 20 oder 25.

- Unterschied zwischen den  $n$ -ten Generationen verschiedener Verfahren und der  $n$ -ten Generation des Direkteinbettungscodecs.
- Unterschied zwischen  $n$ -ter und  $m$ -ter Generation des Direkteinbettungscodecs. Für festes  $n$  und verschiedene größere  $m$  wird so geprüft, inwieweit sich die Qualität ab Generation  $n$  noch weiter verschlechtert.
- Absoluter Höreindruck der  $n$ -ten Generationen des Direkteinbettungscodecs für verschiedene  $n$ .

Ein Grundverlust des Layer II-Codec für die erste Generation ist bei genauerer Analyse bei fast allen Stücken erkennbar. Allerdings ist die Unterscheidung zwischen Original und erster Generation meist nur durch die Klassifikation als „unterschiedlich“ und nicht als „verschlechtert“ möglich. Qualitätsbeeinträchtigungen treten z.B. beim Kastagnettenstück (leichtes Vorecho), Glockenspiel und beim Violinenstück auf. Teilweise sind die hohen Frequenzanteile etwas verzerrt. Bis auf das Extrembeispiel des Violinenstücks wurden diese Effekte jedoch von un-geübten Testhörern mit wenigen Ausnahmen nicht bemerkt. Es lassen sich einige grundlegenden Resultate über die vom Direkteinbettungscodec erzeugten Generationen zusammenfassen:

- Bei den Stücken mit durchgehend hoher Einbettungskapazität (z.B. 2, 7, 8, 9, 11, 12, 17, 18, 20, 21) ist bei ersten Hörtests zunächst kein Unterschied zwischen irgendeiner der ersten 25 Generationen und der 1. Generation erkennbar. Bei vier Stücken zeigen sich bei intensivem Vergleich mit der ersten Generation an jeweils einer Stelle ein leichtes tieffrequentes Artefakt („Plop“-Geräusch). Die wahrgenommenen Artefakte sind ca. ab der fünften Generation schwach erkennbar, sind bis zur ca. 10. Generation ausgeprägt und ändern sich dann nicht mehr. Bei den restlichen der obigen Stücke konnten keine Veränderungen wahrgenommen werden.
- Bei den Sprachstücken ist eine leichte Verschlechterung der Klangqualität zu beobachten, die aber nicht schlechter als die zweite oder dritte Generation irgendeines der Standardverfahren wird.
- Bei allen Stücken liefert der Direkteinbettungscodec einen gewissen stabilen Zustand in dem sich die Klangqualität nicht mehr wesentlich ändert. Diese Zustand pendelt sich ca. ab der 5. Generation (von Stück zu Stück leicht variierend) ein.
- Zwei der percussiven Stücke zeigen Beeinträchtigungen: Beim Kastagnettenstück wird das sowieso vorhandene Vorecho leicht verstärkt, die Energieverteilung bleibt jedoch, im Gegensatz zur extremen Degeneration mit dem Standardverfahren erhalten. Beim Percussionstück d9\_1, das aus nur einigen Schlägen zweier Trommeln besteht, wird das hochfrequente Zischen zwischen den Schlägen verwischt. Die Degenerationen des Standardverfahrens zerstören hierbei den Charakter der Schläge völlig. Im Gegensatz dazu zeigt das Baßstück qualitativ gute Ergebnisse.
- Einige Artefakte, die in Ansätzen bereits in der ersten Generation enthalten waren, werden in den Generationen des Direkteinbettungscodecs verstärkt, so z.B. im Live-Stück 122\_1 oder im Sprachstück 088speec.
- In wenigen Stücken ist die Einbettung an ein bis zwei Stellen leicht als Hintergrundgeräusch wahrnehmbar. Diese Beobachtung wurde jedoch nur von sehr geübten Hörern gemacht.

- Die Qualität aller Generationen des Direkteinbettungscodecs ist (bis auf die Ausnahme des Glockenspielstücks, das beim Standardverfahren länger stabil bleibt) deutlich besser als sämtliche Generationen höher als fünf bei den Standardverfahren.
- In den Langzeittests änderte sich die Qualität von der 25. bis zur 50. Generationen nur noch wenig. Insgesamt war so die Qualität von der 5. bis zur 50. Generation sehr stabil.

Subjektiv entspricht die Qualität der Generationen des Direkteinbettungscodecs sogar fast immer mindestens der zweiten oder dritten Generation des Standardverfahrens. Um diesen Eindruck zu festigen und die tatsächliche Qualität zumindest tendenziell einordnen zu können, wurde ein Hörtest mit einer systematischen Befragung von 26 Testpersonen vorgenommen.

### 6.5.6 Hörtests und Befragung von Testpersonen

Für den Hörtest wurden aus der Menge der Stücke in Tabelle 6.6 die zehn markierten ( $\checkmark$ ) ausgewählt. Dabei wurde wiederum darauf geachtet, eine möglichst umfassende Auswahl zu treffen. Der Hörtest wurde an zwei aufeinanderfolgenden Tagen mit zwei disjunkten Mengen von jeweils 13 Personen durchgeführt, davon vier Frauen und 22 Männer. Die Personen setzten sich aus Mitarbeitern und Studenten der Arbeitsgruppe und der Abteilung V am Institut für Informatik zusammen. Die Personen wurde vorab über die Art des Tests informiert. Zur Gewöhnung an das Testmaterial wurde ihnen zunächst von jedem der Stücke die erste Generation vorgespielt. Zur Gewöhnung an die möglichen Klangbeeinträchtigungen wurde ihnen daraufhin eine zufällig ausgewählte Folge von Stücken verschiedener Generationen, die mit verschiedenen der obigen Codec-Konfigurationen erzeugt wurden, vorgespielt. Der Test dauerte jeweils ca. eine Stunde und zwanzig Minuten mit einer Pause von ca. fünf Minuten. Insgesamt wurden acht Testreihen durchgeführt, bei denen jeweils entweder eine absolute Einschätzung der Klangbeeinträchtigung laut MOS Bewertung auf der Impairment-Skala (siehe Kapitel 3), oder eine relative Beurteilung zweier hintereinander gespielter Stücke zueinander gefordert wurde (siehe ebenfalls Kapitel 3). Bei den relativen Tests wurden die Referenzstücke des Standardverfahrens diejenigen mit zufälligen Translationen zwischen den Codecstufen ausgewählt. Folgende Testreihen wurden im einzelnen durchgeführt:

1. Relativer Vergleich der 1. und der 3. Generation des Standardverfahrens (siehe Kapitel 3). Die zwei Versionen eines Stücks wurden dabei stets hintereinander, jedoch in einer zufällig gewählten Reihenfolge, wiedergegeben.
2. Relativer Vergleich der 5. Generation des Standardverfahrens und der 5. Generation des Direkteinbettungscodecs.
3. Relativer Vergleich der 1. Generation des Standardverfahrens und der 25. Generation des Direkteinbettungscodecs.
4. Relativer Vergleich der 5. Generation des Standardverfahrens und der 25. Generation des Direkteinbettungscodecs.
5. Absoluter Höreindruck auf der MOS-Skala für die
  - (a) 1. Generation des Standardverfahrens,
  - (b) 5. Generation des Standardverfahrens,

- (c) 5. Generation des Standardverfahrens *mit* synchronisierenden Translationen,
- (d) 5. Generation des Direkteinbettungscodecs,
- (e) 15. Generation des Direkteinbettungscodecs.

Es wurden jeweils alle fünf Versionen eines Stücks hintereinander abgespielt. Die Abfolge der einzelnen Versionen wurden dabei zufällig ausgewählt.

6. Relativer Vergleich der 3. Generation des Standardverfahrens und der 5. Generation des Direkteinbettungscodecs.
7. Befragung der Testpersonen nach einer groben Beschreibung eventuell wahrgenommener Klangbeeinträchtigungen in der 25. Generation des Direkteinbettungscodecs. Hier sollten stichpunktartig die Eindrücke der Testpersonen aufgezeichnet werden.
8. Relativer Vergleich der 10. Generation des Standardverfahrens und der 25. Generation des Direkteinbettungscodecs.

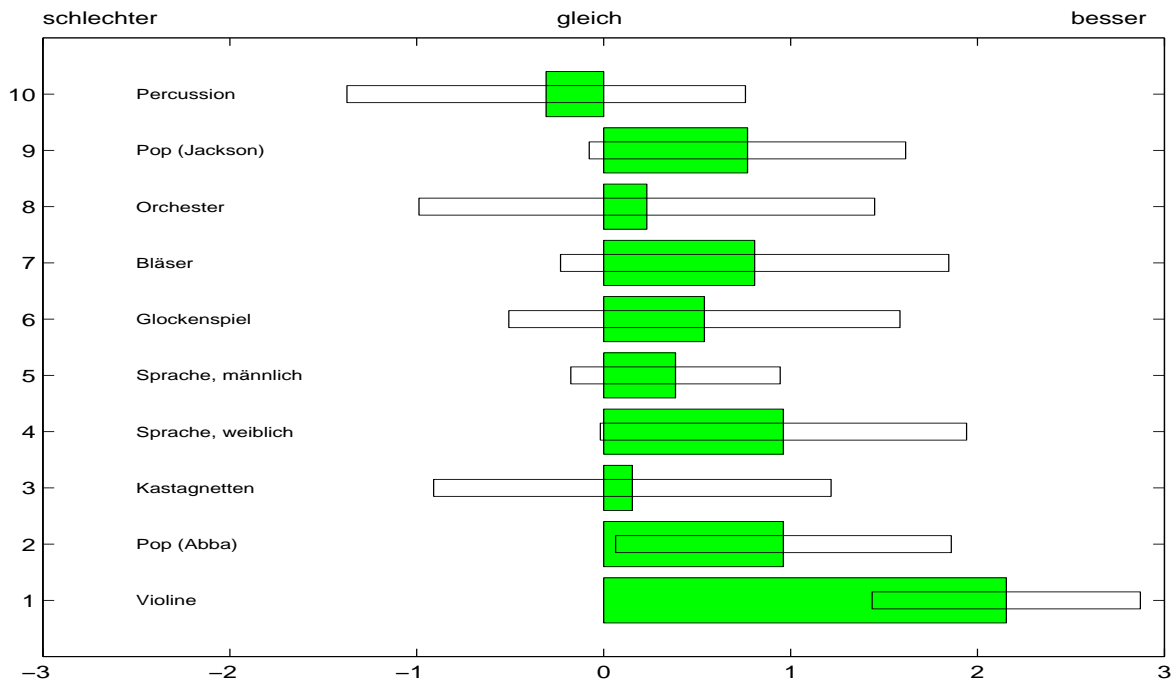


Abbildung 6.14: Relativer Vergleich der 5. Generation des Standardverfahrens und der 5. Generation des Direkteinbettungscodecs. Bei positiven Werten wurde der Direkteinbettungscodec als besser bewertet.

Es ist sehr wichtig zu bemerken, daß alle Arten solcher informellen Hörtests sehr kritisch zu sehen sind und höchstens als Lieferant für Tendenzinformationen verwendet werden können. So zeigt die Erfahrung aus verschiedenen vorherigen Hörtests sowie aus Gesprächen mit den Testhörern, daß einige Randbedingungen zwangsläufig unklar oder unscharf bleiben müssen:

- Begriffe wie *Klangbeeinträchtigung* oder *Qualität* sind rein subjektiv.

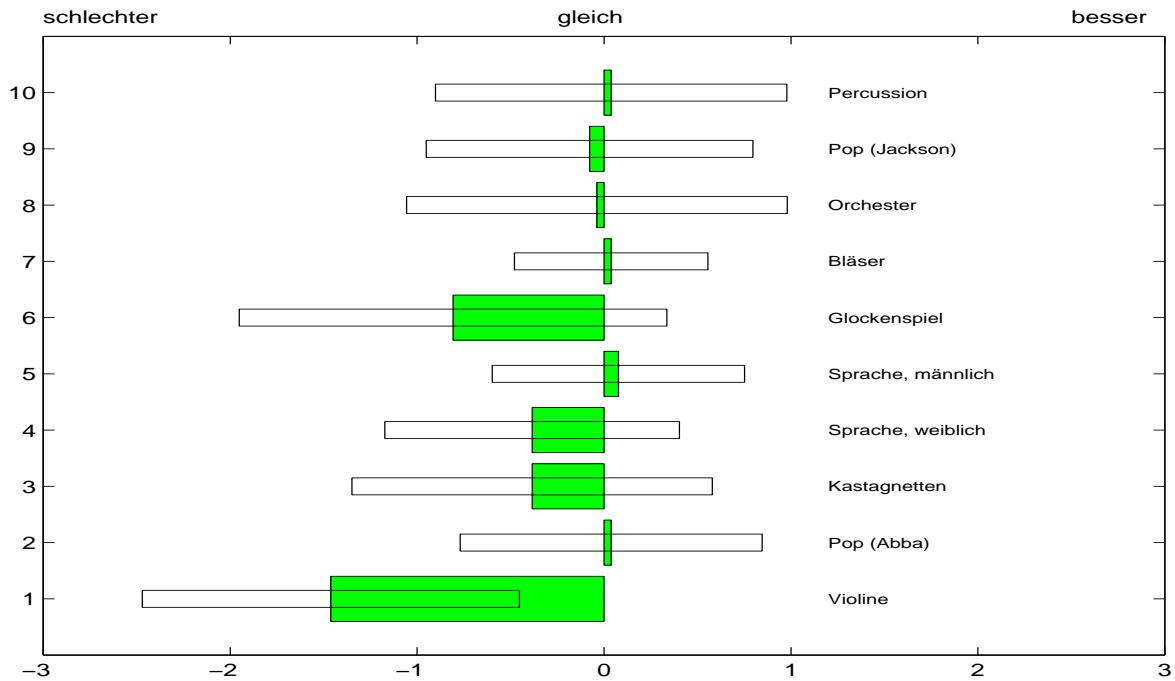


Abbildung 6.15: Relativer Vergleich der 1. Generation des Standardverfahrens und der 25. Generation des Direktinbettungscodecs. Bei negativen Werten wurde die erste Generation als besser bewertet.

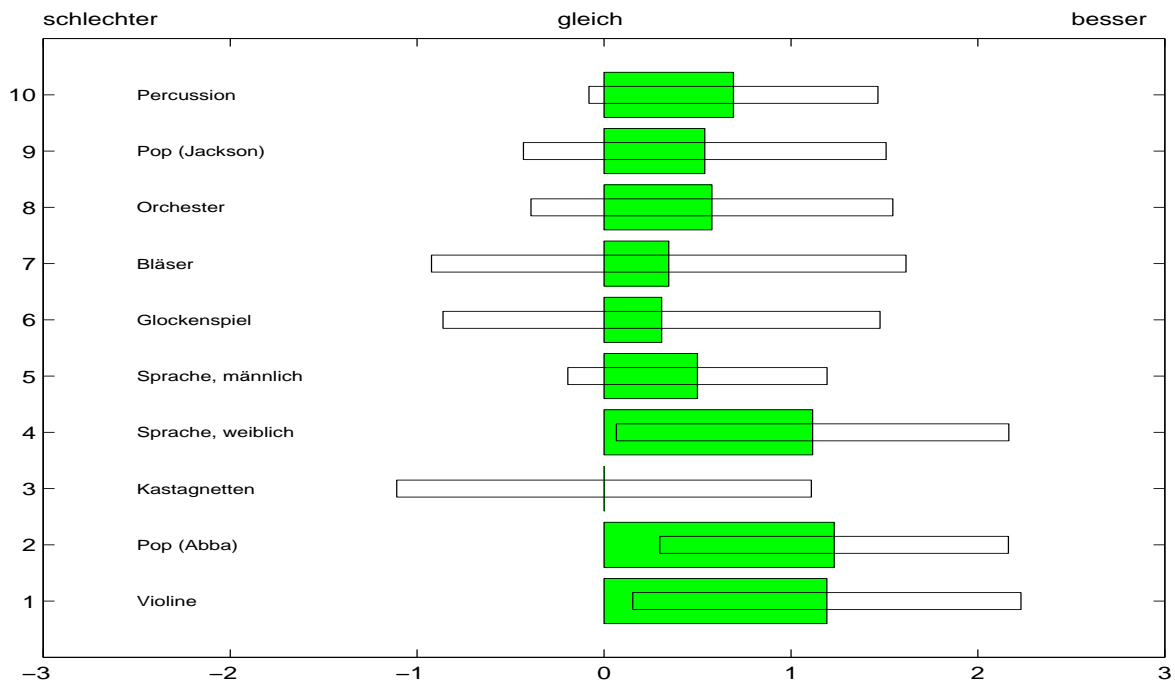


Abbildung 6.16: Relativer Vergleich der 5. Generation des Standardverfahrens und der 25. Generation des Direktinbettungscodecs. Bei positiven Werten wurde die 25. Generation als besser bewertet.

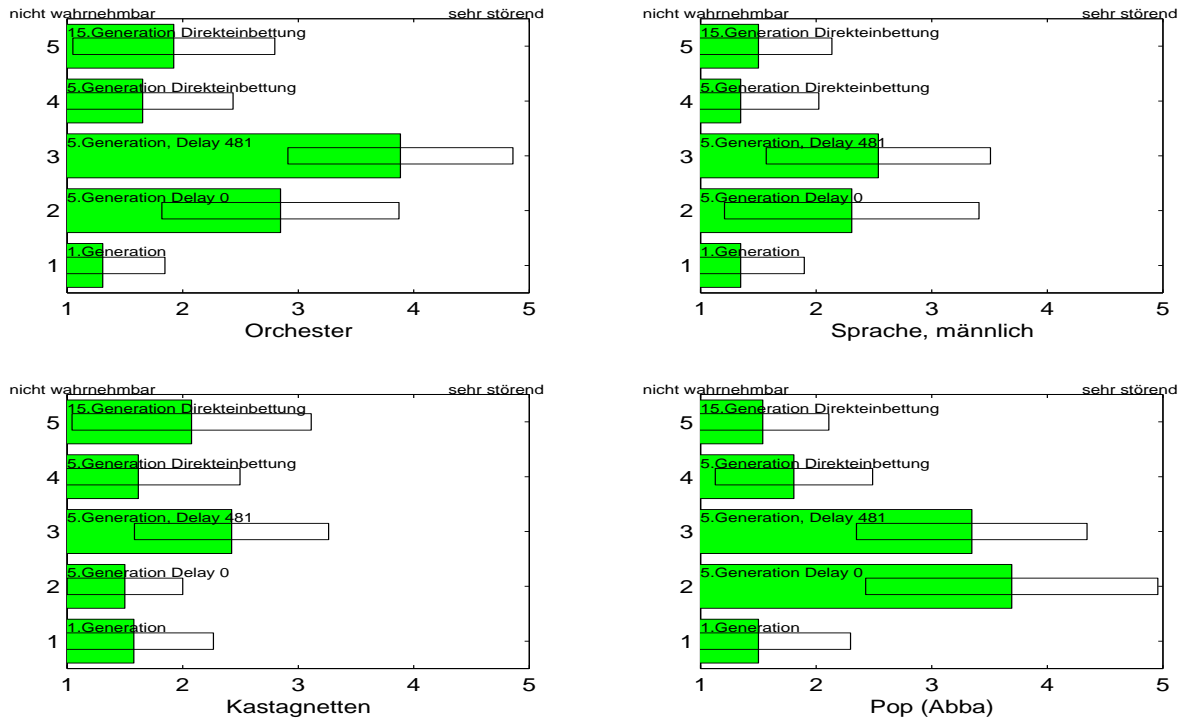


Abbildung 6.17: Absoluter Höreindruck auf der MOS-Skala für fünf Versionen der Teststücke, Teil 1.

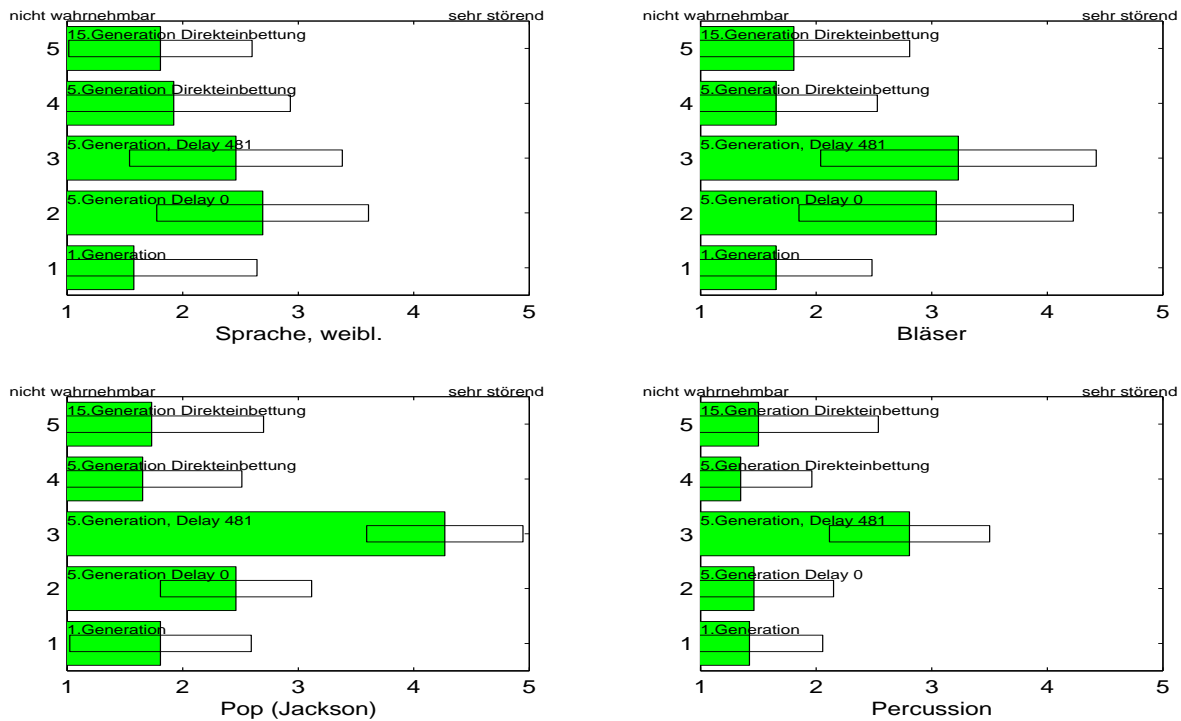


Abbildung 6.18: Absoluter Höreindruck auf der MOS-Skala für fünf Versionen der Teststücke, Teil 2.

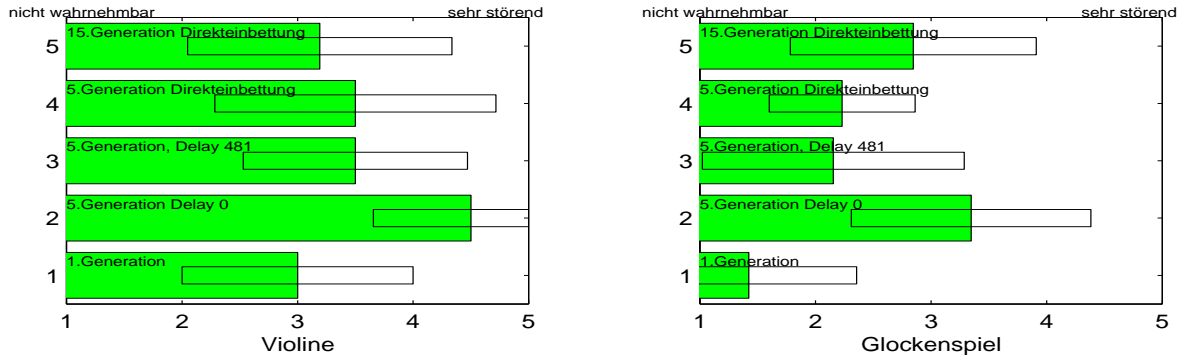


Abbildung 6.19: Absoluter Höreindruck auf der MOS-Skala für fünf Versionen der Teststücke, Teil 3. Hier sind die kritischen Stücke dargestellt. Das Violinenstück weist dabei eine schlechte Qualität in der ersten Generation auf, wohingegen das Glockenspielstück insgesamt nur wenig degeneriert.

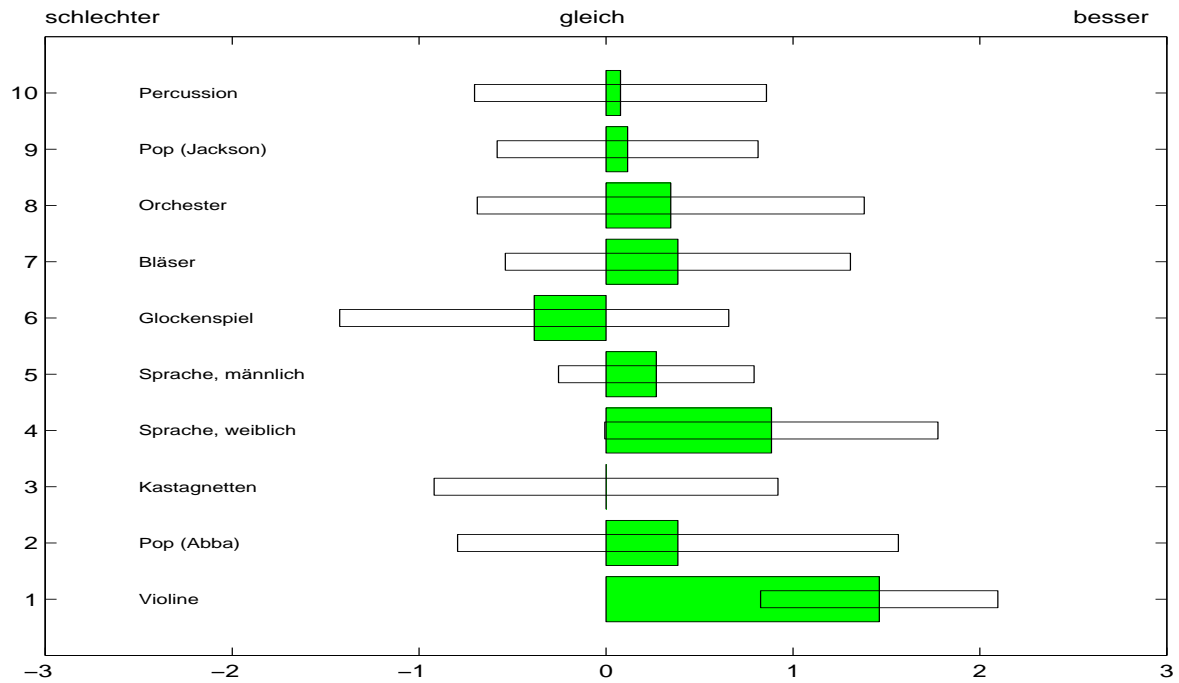


Abbildung 6.20: Relativer Vergleich der 3. Generation des Standardverfahrens und der 5. Generation des Direkteinbettungscodecs. Bei positiven Werten wurde die 5. Generation als besser bewertet.

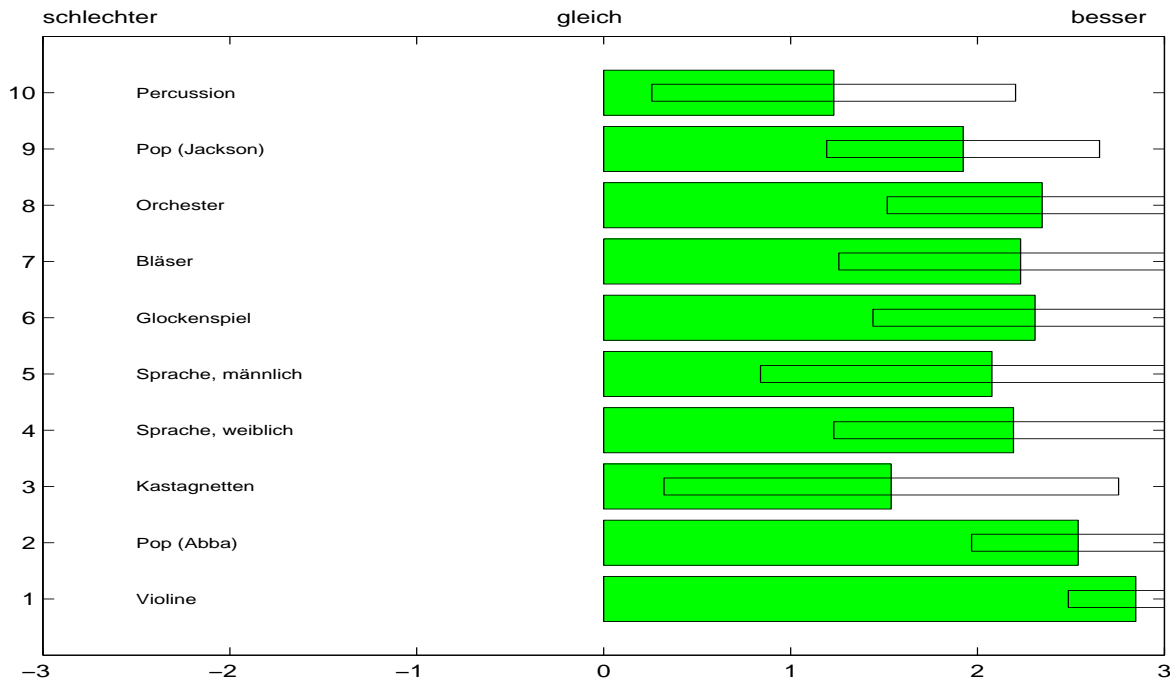


Abbildung 6.21: Relativer Vergleich der 10. Generation des Standardverfahrens und der 25. Generation des Direkteinbettungscodecs. Bei positiven Werten wurde die 25. Generation als besser bewertet.

- Die Quantifizierung von Klangeindrücken, die qualitativ eingeschätzt wurden, ist auch rein subjektiv. So gab es z.B. jeweils einige Hörer, die die relativen Skalen systematisch im Intervall  $[-1 : 1]$  nutzten und auch solche, die durchgehend das Intervall  $[-3 : 3]$  voll ausschöpften.
- Aufgrund der vorherigen Bemerkung sind die zum Teil hohen Standardabweichungen der Testergebnisse erklärbar.
- Einigen Testhörern war unklar, wie Stücke verglichen werden sollten, bei denen beide Versionen verschieden jedoch gleich stark im Klang beeinträchtigt waren.

Vor diesem Hintergrund sind die folgenden Testauswertungen zu sehen.

Die Abb. 6.14 bis 6.21 zeigen die Ergebnisse der Tests als Balkendiagramm. Da hier tatsächlich nur Tendenzen betrachtet werden sollen, sind keinen numerischen Werte angegeben. Die Balken enthalten jeweils die Mittelwerte  $E(x)$  der Testergebnisse  $x \in [-3 : 3]^{26}$  bzw.  $x \in [1 : 5]^{26}$  und die Standardabweichungen  $\sqrt{E((x - E(x))^2)}$  eingezeichnet als Abweichung vom Mittelwert nach unten und oben. Die Ergebnisse wurden im Sinne einer einheitlichen Darstellungsweise so normalisiert, daß die dargestellten Werte einer einheitlichen Abfolge der verschiedenen Versionen (also z.B. immer zuerst die ersten Generationen und dann die 5.) entsprechen.

Es lassen sich folgende Tendenzen und Beobachtungen zusammenfassen:

- Für sechs der Stücke konnten im Test vom Abb. 6.15 die erste Generation nicht von der 25. Generation mit dem Direkteinbettungscodec unterschieden werden.

- Beim Violinen- und Kastagnettenstück trägt die vorhandene Beeinträchtigung der ersten Generation dazu bei, daß schon die Grundqualität von den Hörern sehr verschieden eingeschätzt wird. Besonders extrem zeigt dies die absolute Einschätzung des Violinenstücks, die bereits relativ schlecht ausfällt.
- Das Glockenspielstück ist aufgrund seiner recht stabilen Natur nur schwer zu bewerten. Klarere Tendenzen treten hier nur im letzten Test zutage.
- Der zweite Test zeigt, daß die 5. Generationen mit dem Direkteinbettungscodec deutlich besser als die 5. Generationen des Standardverfahrens eingeschätzt wurden. Der sechste Test verschärft dies im wesentlichen auf die 3. Generation des Standardverfahrens. Damit ist die Beobachtung der ungefähren Einschätzung der allgemeinen Qualität mit mindestens der zweiten bis dritten Generation des Standardverfahrens bestärkt.
- Test acht zeigt die ganz eindeutige Verschlechterung einer hohen Generation des Standardverfahrens (10.) im Gegensatz zur 25. Generation des Direkteinbettungscodecs. Dies bestätigt die Vorgehensweise der Beschränkung auf ansonsten höchstens die 5. Generation des Standardverfahrens.
- Die durch die Einbettung erzeugten „Plop“-Geräusche wurden von den Testpersonen nicht erkannt. Die Kritikpunkte bei der Befragung bezogen sich bei einigen Stücken auf „Kratzen“, „Verzerrung“, „Dumpfheit“ oder „Scheppern“, was auch zu einem gewissen Teil mit der initialen Qualität der ersten Generationen zu tun hat. Alleine im Violinenstück wurde von drei Personen ein leichtes „Plätschern“ wahrgenommen.

Die Testergebnisse bestätigen somit die Analyseergebnisse des vorherigen Abschnitts.

### 6.5.7 Objektive Messungen

Auch objektive Abstandsmessungen bestätigen die beschriebenen Ergebnisse. Die in den Abbildungen 6.22 bis 6.28 dargestellten segmentellen SNR-Werte wurden jeweils für aufeinanderfolgende Blöcke der Länge 1152 (entsprechend einer MPEG-Rahmenlänge) der Signale berechnet. Die Abbildungen 6.23 und 6.25 geben hier eine *punktweise* Analyse der Differenzen zwischen verschiedenen Generationen.

In Abb. 6.22 sind für den Fall des Kastagnettenstücks 044kasta segmentelle SNR-Werte zwischen verschiedenen Generationen dargestellt. Die vier Teile der Abbildung zeigen, von oben nach unten, jeweils den segmentellen SNR zwischen der 1. und 3., der 3. und 5., der 5. und 10. sowie der 10. und 25. Generation. Die eingezeichneten Kreise bei jeweils 80 dB zeigen dabei an, daß die betreffenden Segmente übereinstimmen (der SNR somit unendlich groß ist). Dies bedeutet, daß durch die Einbettung die exakte Reproduktion des Signals für diese Segmente erreicht wird. Deutlich ist zu erkennen, daß sich mit wachsender Generation immer weniger Segmente ändern. Diese Segmente stimmen mit denjenigen überein, in die eingebettet werden kann (vgl. Abb. 6.23 oben). Die SNR-Werte an den Stellen, an denen nicht eingebettet werden konnte, sind analog zu denen, die aus der Anwendung des Standardverfahrens mit synchronisierendem Delay resultieren. Es fällt auf, daß bereits im Bereich der dritten Generation eine große Anzahl stabiler Segmente existiert.

Abb. 6.23 zeigt im unteren Teil die Wellenform zum Kastagnettenstück. Darüber ist punktweise das Differenzsignal zwischen der 5. und der 10. Generation angegeben. Die horizontalen Balken im oberen Teil der Abbildung zeigen an, ob sich die betreffenden Samples in einem

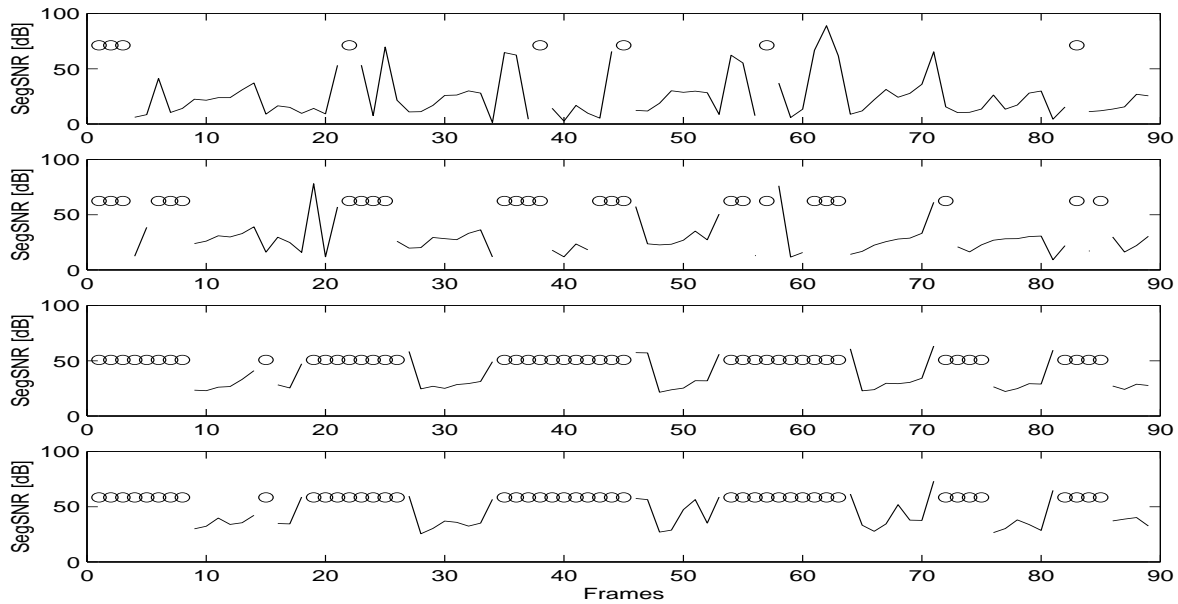


Abbildung 6.22: Die vier Teile der Abbildung zeigen jeweils den segmentellen Signal-Rauschabstand zwischen 1. und 3. (oben), 3. und 5. (2. von oben), 5. und 10. (2. von unten) bzw. 10. und 25. Generation (unten) für das Kastagnettenstück 044kasta. Die eingezeichneten Kreise bei jeweils 80 dB zeigen dabei an, daß die betreffenden Segmente übereinstimmen.

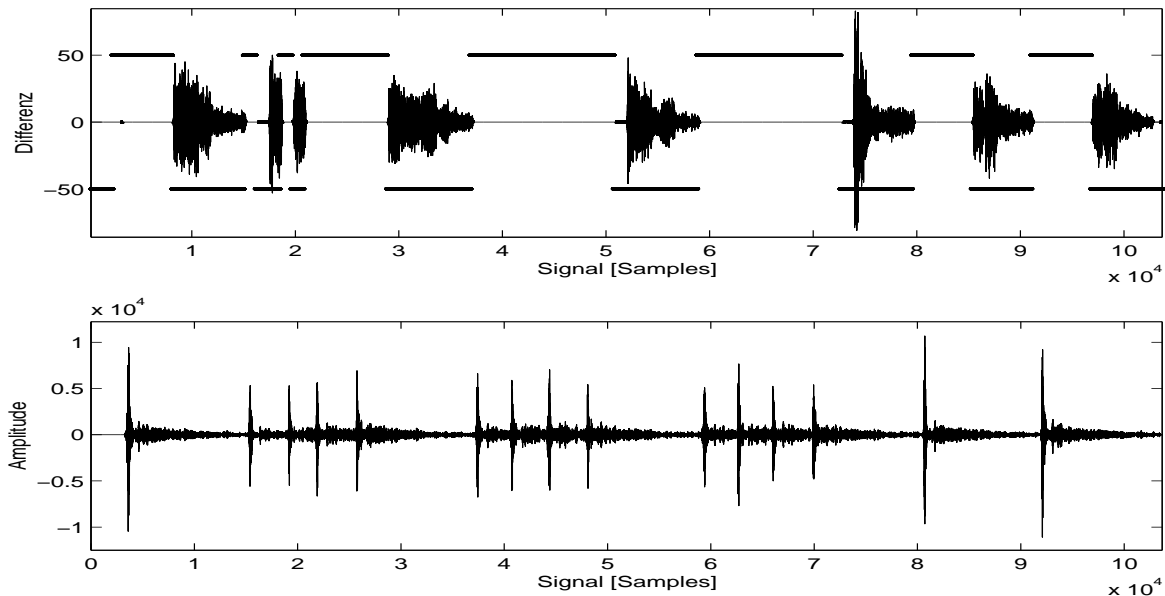


Abbildung 6.23: Die Abbildung zeigt im unteren Teil die Wellenform zum Kastagnettenstück. Darüber ist das Differenzsignal (punktweise) zwischen der 5. und der 10. Generation angegeben. Die oberen horizontalen Balken (bei 50) zeigen an, daß an der betreffenden Stelle eingebettet werden konnte, die unteren, daß nicht eingebettet werden konnte. Es ist deutlich zu erkennen, daß das Differenzsignal an den Einbettungsstellen verschwindet.

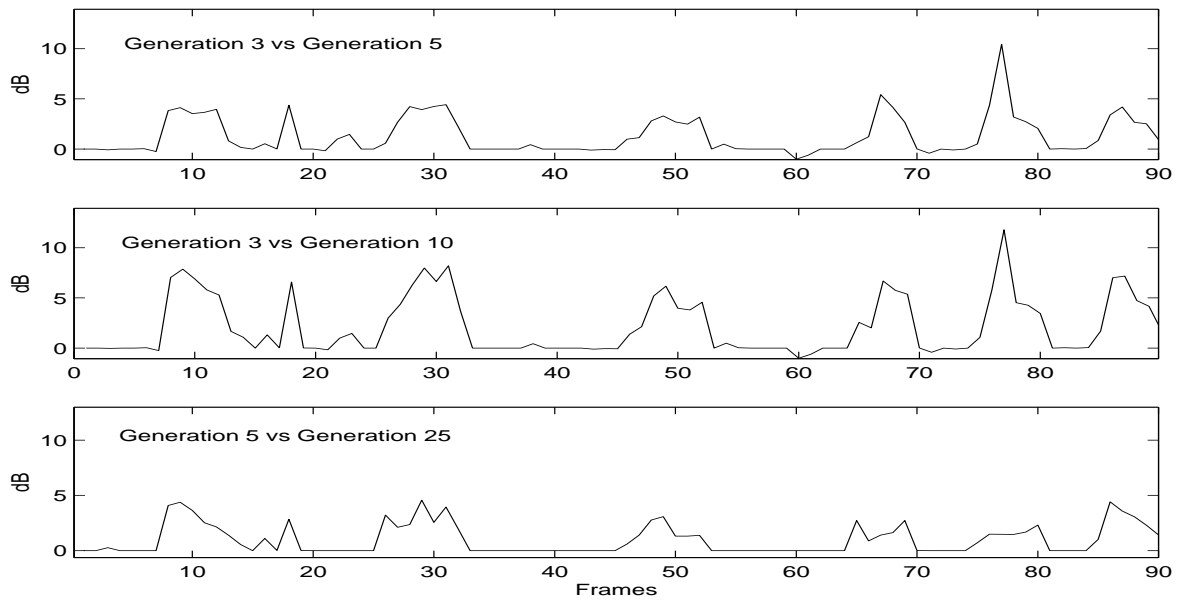


Abbildung 6.24: Die Abbildung zeigt einen relativen SNR zwischen den Generationen 3 und 5 (oben), 3 und 10 (mitte) bzw. 5 und 25 (unten). Werte um Null zeigen, daß sich der Rauschabstand nicht ändert, Werte größer Null, daß die höhere Generation relativ einen höheren SNR liefert.

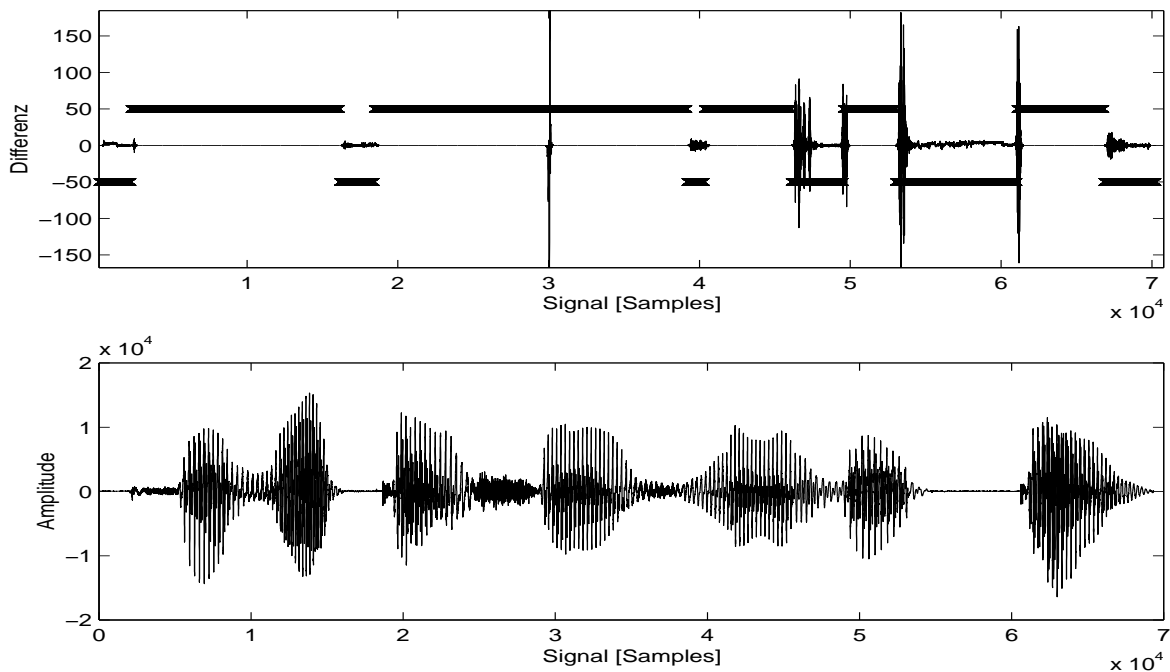


Abbildung 6.25: Die Abbildung zeigt analog zu Abbildung 6.23 die Beziehung zwischen exakter punktweiser Rekonstruktion und möglicher Einbettung im Falle des Sprachstücks 087spec.

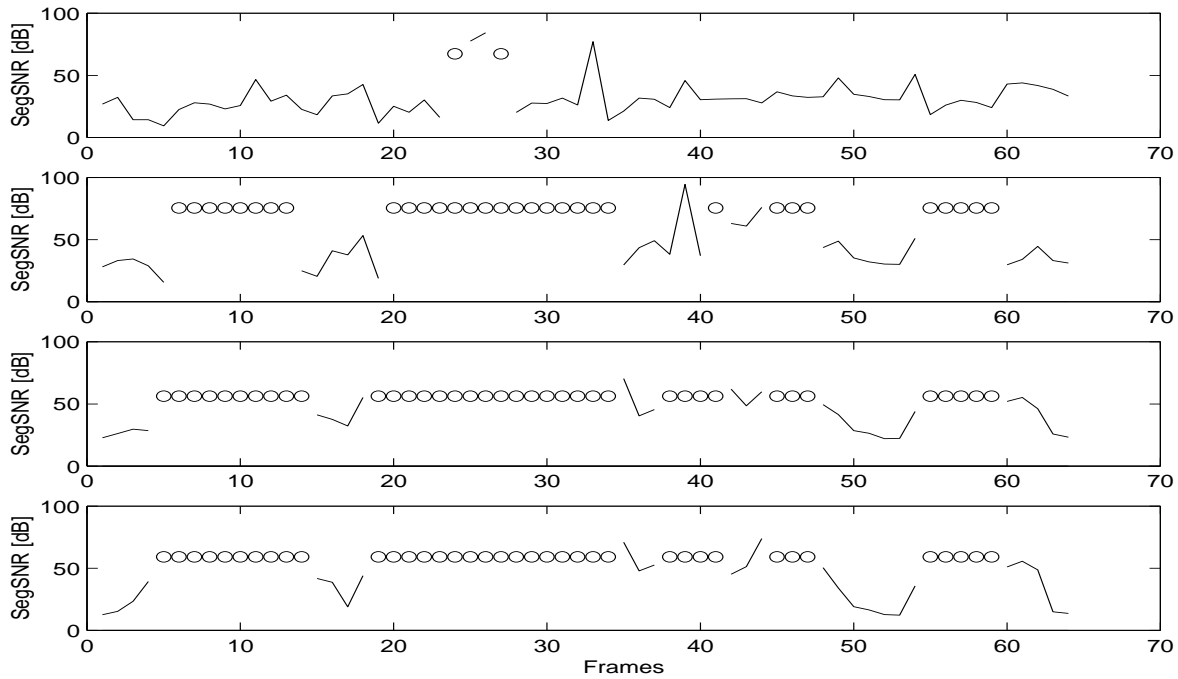


Abbildung 6.26: Die Abbildung zeigt analog zur Abbildung 6.22 den segmentellen SNR zwischen verschiedenen Generationen des Sprachstücks.

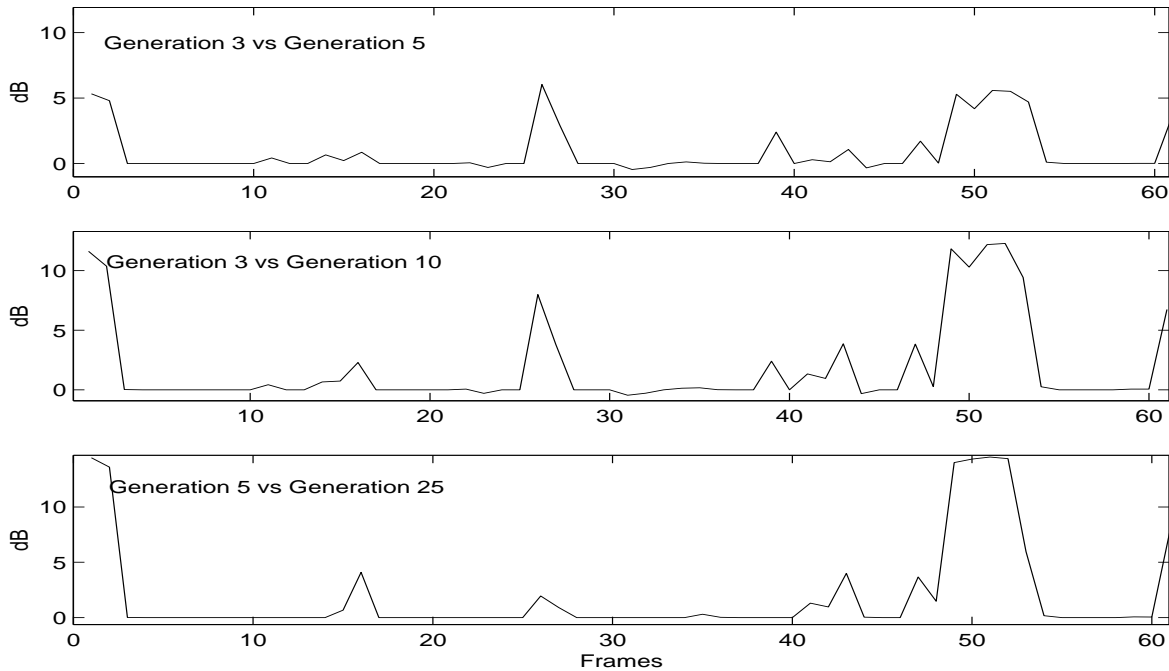


Abbildung 6.27: Die Abbildung zeigt analog zur Abbildung 6.24 den relativen segmentellen SNR zwischen verschiedenen Generationen des Sprachstücks.

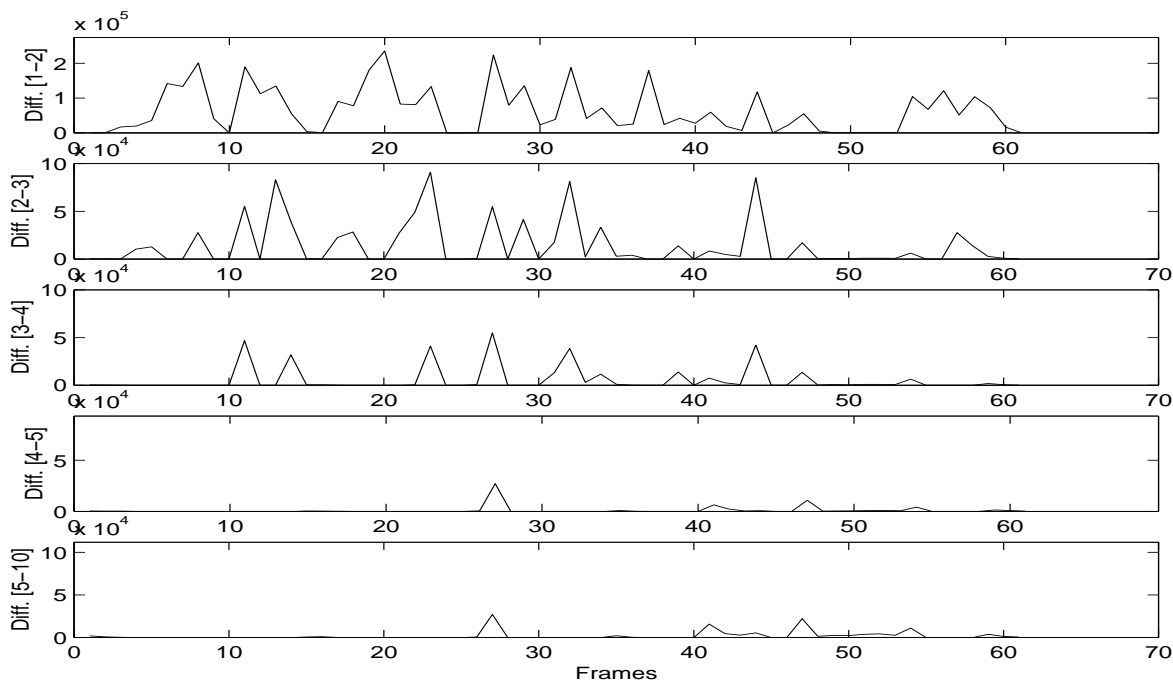


Abbildung 6.28: Die Abbildung zeigt segmentelle  $\ell_1$ -Abstände für verschiedene Generationen. Von oben nach unten sind dies die 1.-2., 2.-3., 3.-4., 4.-5. sowie die 5.-10. Generation. Es ist deutlich zu erkennen, daß die Abstände immer geringer werden.

Rahmen befinden, in den eingebettet werden konnte. Die oberen Balken (bei Amplitude 50) zeigen an, daß an der betreffenden Stelle eingebettet werden konnte, die unteren, daß nicht eingebettet werden konnte. Es ist deutlich zu erkennen, daß das Differenzsignal an den Einbettungsstellen verschwindet. Dies ist auch rechnerisch nachzuweisen: an den betreffenden Stellen sind die Signale tatsächlich punktweise identisch.

Die Abb. 6.24 zeigt einen relativen segmentellen SNR zwischen den Generationen 3 und 5 (oben), 3 und 10 (mitte) bzw. 5 und 25 (unten). Hier ist der relative SNR zwischen Signalen  $g_i$  und  $g_j$  der Generationen  $i$  und  $j$  als

$$\text{SNR}(g_i, g_j) := 10 \log_{10} \frac{\sum_k (g_1(k) - g_i(k))^2}{\sum_k (g_1(k) - g_j(k))^2}$$

definiert. Das Signal erster Generation,  $g_1$ , dient dabei als Referenz. Der segmentelle relative SNR ergibt sich entsprechend. Für die hier angegebenen Beispiele ist stets  $i > j$ . Werte um Null zeigen, daß sich der Rauschabstand nicht ändert, Werte größer Null, daß die höhere Generation einen höheren SNR liefert. Es ist wiederum klar ersichtlich, daß das Signal an den Einbettungsstellen stabil bleibt, wohingegen der relative SNR an den anderen Stellen, je nach Größe der Generationendifferenz, ansteigt.

Die Abbildungen 6.25 bis 6.27 zeigen analoge Ergebnisse für das Sprachstück 087speec. In Abb. 6.28 sind zusätzlich die segmentellen  $\ell_1$ -Abstände für verschiedene Generationen aufgetragen. Von oben nach unten sind dies die 1.-2., 2.-3., 3.-4., 4.-5. sowie die 5.-10. Generation. Es ist deutlich zu erkennen, daß die Abstände für steigende Generationen immer geringer werden.

Ähnliche Messungen zu den anderen Teststücken bestätigen die exemplarisch dargestellten Ergebnisse. Zusammenfassend ist auch beim Direkteinbettungscodec an den Einbettungsstellen eine exakte Signalreproduktion möglich. Diese ist allerdings noch nicht bezüglich der ersten sondern erst ab der dritten bis fünften Generation erzielbar. Es ist bemerkenswert, daß der durch die Direkteinbettung modifizierte „Codec-Operator“ *DC* im Falle des MPEG-1 Layer II-Codecs offensichtlich durch wiederholte Anwendung lokale Fixpunkte im codierten Signal liefert. Die Hörtests zeigen hier eine gewisse „Gutartigkeit“ auf: an den Übergangsstellen zwischen möglicher und nicht möglicher Einbettung sowie dort, wo in keiner Generation eine Einbettung möglich ist, bleibt die perzeptuelle Signalqualität trotzdem erhalten. Somit bleibt die Qualität des gesamten Signals stabil.

# Kapitel 7

## MPEG-1 Layer III - Codec

In diesem Kapitel wird eine Erweiterung des im letzten Kapitel angegebenen Codecs auf den Layer III des MPEG-Standards [18] entworfen. Hierauf aufbauend wird in [50] eine Codec-Implementierung vorgenommen.

Zunächst wird kurz die Funktionsweise eines Layer III-Codecs nach ISO/IEC 11172-3 vorgestellt. Es wird hier insbesondere auf die Unterschiede und Erweiterungen zu Layer II-Codecs eingegangen. Im zweiten und dritten Abschnitt werden die Berechnung der zur Verfügung stehenden Einbettungskapazität und die relevanten Encoderinformationen behandelt. Die Einbettung, die in diesem Fall mehrstufig vorgenommen werden kann, wird im vierten Abschnitt behandelt. Abschließend geht der fünfte Abschnitt auf Problematiken in Zusammenhang mit Implementationen unter Direkteinbettungscodern ein.

### 7.1 Funktionsweise von Layer III-Codecs

Wie im letzten Kapitel ist der beschriebene Encoder nicht standardisiert sondern nur durch den Standard vorgeschlagen. Freiheiten liegen insbesondere bei der Wahl der Quantisierungsaufösung, der Bitallokierung und beim psychoakustischen Modell. Hierbei können prinzipiell beide im Standard vorgeschlagenen Modelle mit leichten Modifikationen verwendet werden. Die Modelle liefern jeweils den SMR als Ausgabe und werden in der folgenden Beschreibung nicht weiter berücksichtigt. Zu bemerken ist jedoch, daß in Layer III pro Frame zwei psychoakustische Modelle ausgewertet werden. Dabei wird die zeitliche Auflösung der Analyse-DFT in Abhängigkeit der MDCT-Blocklänge zwischen 256 (kurze Blocks, s.u.) und 1024 (lange Blocks) variiert.

#### 7.1.1 Encoder

Die wesentlichen Unterschiede von Layer II gegenüber Layer III sind

- die Verwendung einer zusätzlichen Spektraltransformation auf den Subbandsignalen,
- die Verwendung einer Bitreservoirtechnik, die eine variable Bitrate innerhalb gewisser Grenzen erlaubt,
- adaptive Anpassung der Transformation auf den Signalcharakter (lokal stationär oder nicht),

- Verwendung logarithmischer Quantisierung,
- Hinzunahme einer Huffman-Codierung zur Redundanzminimierung,
- (notwendigerweise) komplexere Bitallokierungsstrategie in Form eines Analysis-by-Synthesis Verfahrens,
- Verzicht auf die formale Trennung zwischen Bitallokierung und Skalierung.

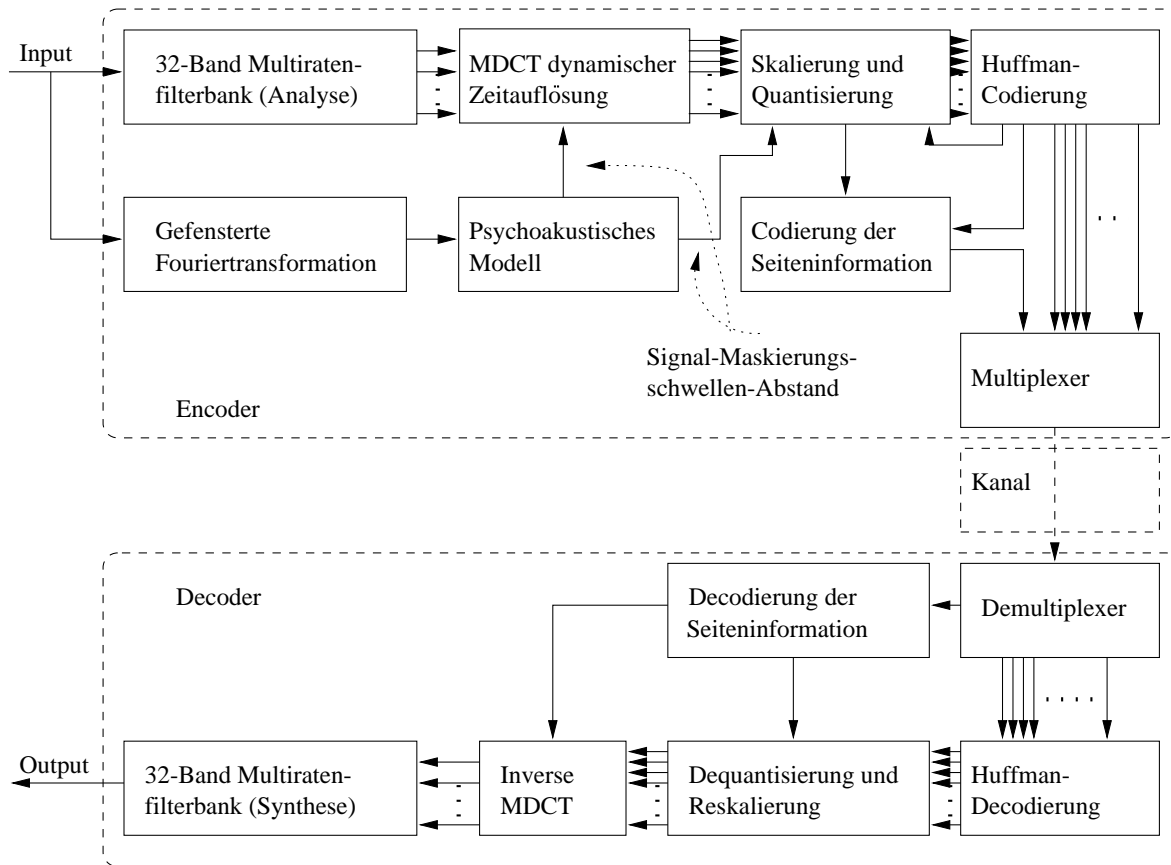


Abbildung 7.1: Aufbau eines Codecs nach ISO/IEC 11172-3, Layer III.

Abb. 7.1 gibt einen Überblick über die Struktur eines Layer III-Codecs.

### 7.1.1.1 Analysetransformation und Blockgrößen

Wie in Layer II werden blockweise Rahmen aus 1152 Abtastwerten pro Kanal verarbeitet. Nach der Filterung mit der 32-Band Filterbank werden die resultierenden 36 Werte jedes Subbands jeweils in zwei zeitliche Blöcke von 18 Werten aufgeteilt. Die so aus der Zusammenfassung jeweils aller Subbandblöcke entstehenden zwei Blöcke der Größe  $32 \cdot 18 = 576$  werden als *Granules* bezeichnet.

Für jedes Granule wird mit Hilfe des psychoakustischen Modells die Art der zu verwendenden MDCT (*Modified Discrete Cosine Transform*) bestimmt. Hierbei gibt es die Typen *long*, *start*, *short* und *stop*, wobei der Typ *long* eine MDCT mit Fensterlänge 36 und der Typ

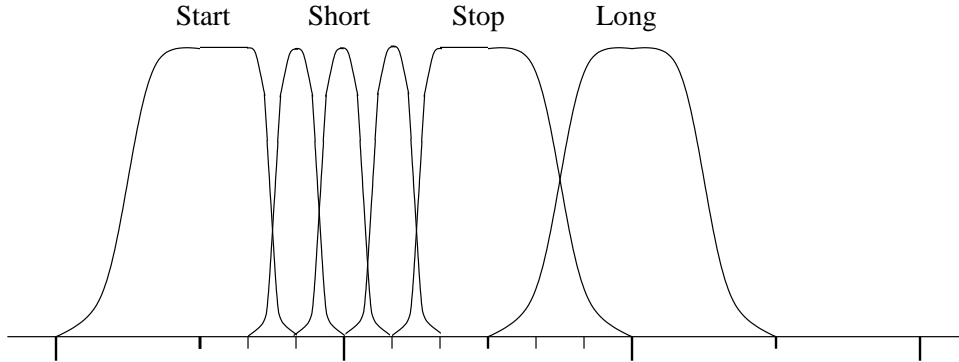


Abbildung 7.2: Die vier verschiedenen möglichen Fensterarten der MDCT-Blöcke.

*short* drei MDCTs der Fensterlänge 12 pro Granule bedeutet. Abb. 7.2 zeigt beispielhaft die Fensterfolge für ein Subband bei der Blockabfolge *start-short-stop-long*. Die Typen *start* bzw. *stop* implizieren die Verwendung von Übergangstransformationen zwischen Blöcken der Typen *long* und *short* bzw. *short* und *long*. Die Übergangstransformationen sind jeweils aus einer Hälfte einer Transformation vom Typ *long* und einer von Typ *short* zusammengesetzt und haben die Fensterlänge 30. Die hier zu verwendenden 36 Subbandwerte setzen sich stets aus den 18 Werten des aktuellen und den 18 Werten des vorherigen Granules zusammen. Bei den MDCTs handelt es sich um gefensterte Cosinustransformationen, wobei die Fensterfunktion von Typ eine Hann-Fensters ist. Sie ist für  $N \in \{12, 36\}$  und  $0 \leq i < N$  durch

$$f_i := \sin\left(\frac{\pi}{N}\left(i + \frac{1}{2}\right)\right)$$

gegeben ([18], S.37). Gefensterte trigonometrische Transformationen werden im Detail z.B. in [31] oder [38] behandelt. Sämtliche hier verwendeten MDCTs sind orthogonale Transformationen.

Mit Hilfe der variablen Blockbreiten werden im Layer III-Codec Preechos, die durch zu lange Blockbreiten verursacht werden können, vermieden. Die mögliche Abfolge der Blockbreiten wird durch einen endlichen Automaten ([18], S.95) beschrieben. Zwischen kurzen und langen Blocks muß hiernach jeweils ein Übergangsblock (*start* bzw. *stop*) verwendet werden.

Als Ausnahme können die unteren Subbänder vom Umschalten auf kurze Blocklängen ausgenommen werden. In diesem Fall wird die Variable `window_switching_flag` gleich eins gesetzt.

Im Falle der MDCT-Länge 36 wird auf den resultierenden 576 Spektrallinien eines Granules eine weitere Orthogonaltransformation durchgeführt um das Aliasing der 32-Band Filterbank zu unterdrücken (Butterflystruktur [18], S.42). Schreibt man die 576 Spektrallinien in einen

Vektor  $s$ , so ist dies als Matrixmultiplikation  $Ts$  mit  $T = I_9 \oplus (I_{31} \otimes B) \oplus I_9$  und

$$B = \begin{pmatrix} s_9 & & & & & & & & -a_9 \\ & s_8 & & & & & & & -a_8 \\ & & \ddots & & & & & & \\ & & & s_1 & -a_1 & & & & \\ & & & a_1 & s_1 & & & & \\ & & & & & \ddots & & & \\ & & & & & & & & \\ a_8 & & & & & & & s_8 & \\ a_9 & & & & & & & & s_9 \end{pmatrix}$$

darstellbar, wobei die  $s_i$  und  $a_i$  entsprechend [18], S. 65 gewählt werden mit  $s_9 := 1$  und  $a_9 := 0$ .

### 7.1.1.2 Skalierungsfaktoren

Die Skalierungsfaktoren ergeben sich für die Spektralkomponenten in Layer III nach folgenden Regeln:

- Für lange MDCT-Blöcke gibt es 21 Skalierungsfaktoren, die in vier *Skalierungsfaktorblöcke* eingeteilt sind (0 – 5, 6 – 10, 11 – 15 und 16 – 20). Die Zuordnung eines jeden Skalierungsfaktors zu einer entsprechenden Teilmenge der 576 Spektrallinien wird durch eine Tabelle, abhängig alleine von der Abtastrate, geregelt ([18], B.8). Für kurze MDCT-Blöcke gibt es 11 Skalierungsfaktoren. Für die Ausnahme des Mixed-Block-Modus enthalten die unteren Bänder (lange Blöcke) 8 und die oberen (kurze Blöcke) 9 Skalierungsfaktoren.
- Die Skalierungsfaktorauswahl (`scfsi`) gibt für die vier Gruppen von Skalierungsfaktorbändern jeweils an, ob die zugehörigen Skalierungsfaktoren des ersten Granules auch für das zweite Granule verwendet werden können.

Die Skalierungsfaktoren werden während der Bitallokierung bestimmt.

### 7.1.1.3 Bitallokierung

Die Bitallokierung wird in drei geschachtelten Schleifen vorgenommen. Dieser Vorgang ist wesentlich komplexer als bei den niedrigeren Layern, da aufgrund der Verwendung des nachgeschalteten Huffman-Codierers die durch die bei einer Iteration festgelegten Parameter erzielte Bitrate nicht a priori feststeht.

Im Unterschied zu Layer II wird in der innersten Bitallokierungsschleife inkrementell die Quantisierungsschrittweite erhöht, bis eine gegebene Bitrate eingehalten werden kann. Hier wird also Rauschen „allokiert“ (Bits werden weggelassen), wohingegen in Layer II genau die umgekehrte Strategie verfolgt wird (Bits werden inkrementell hinzugenommen). In der Skalierungsfaktor-Schleife wird daraufhin überprüft, ob die durch die momentanen Skalierungsfaktoren induzierte Verzerrung sich innerhalb akzeptabler Grenzen bewegt. Skalierungsfaktorbänder, die eine zu starke Verzerrung aufweisen, werden verstärkt (d.h. deren Darstellungsgenauigkeit wird erhöht). Durch wechselweise Ausführung beider Schleifen wird so versucht, unter Einhaltung der Bitrate möglichst wenige verzerrte Skalierungsbänder zu erzeugen. Zur Ausnutzung globaler Redundanzen können hierbei noch einige spezielle Skalierungs- und Gainfaktoren verwendet werden (s.u.).

### 7.1.1.4 Huffman-Codierung

Die skalierten Spektralwerte werden in einen unteren und einen oberen Bereich aufgeteilt, wobei der obere Bereich nur Werte mit Betrag kleiner eins enthält. Die Werte dieses Bereichs werden mit Hilfe eines eigenen Huffman-Codes in Viererpaaren codiert. Der untere Bereich wird in drei (lange Blöcke) bzw. zwei (kurze Blöcke) Teile aufgeteilt. Die Werte innerhalb dieser Bereiche werden jeweils mit einem gemeinsamen Huffman-Code codiert. Dazu werden aus aufeinanderfolgenden Werten Paare gebildet. Insgesamt stehen hierzu seitens des Standards 30 Codebücher zur Auswahl, zwei Codebuchnummern bleiben ungenutzt.

## 7.2 Verfügbare Einbettungskapazität

Die Decodierung läuft entsprechend in umgekehrter Weise ab. Für die Feststellung der zur Verfügung stehenden Einbettungskapazität ist hierbei insbesondere die Rückquantisierung von Interesse. Betrachtet man die im Standard angegebene Rückquantisierungsvorschrift, so ergibt sich zusammengefaßt

$$s(q) = \text{sign}(q) \cdot |q|^{4/3} \cdot \underbrace{2^{\frac{1}{4}(g_{\text{global}} - 210 - 8g_{\text{subblock}} - \lambda(s_{\ell} + \delta_{\text{preflag}}\lambda_{\text{pretab}}))}}_{=:F} \quad (7.1)$$

für ein Codewort  $q$  und den rekonstruierten Wert  $s(q)$ .

Dabei tragen die folgenden Parameter (Bezeichnungen wie im Standard) zur globalen bzw. blockweisen Verstärkung innerhalb eines Granules bei. In Klammern ist jeweils der Gültigkeitsbereich angegeben:

`global_gain[granule][channel]` (in (7.1)  $g_{\text{global}}$ ): Globaler Gain, auch als Quantisierungsschrittweite interpretierbar. Es ist wichtig zu bemerken, daß im globalen Gain eine vorher zu bestimmende Systemkonstante enthalten ist.

`subblock_gain[granule][channel][window]` (in (7.1)  $g_{\text{subblock}}$ ): Bei kurzen Blöcken wird ein zusätzlicher Gainfaktor für jeden der drei Blöcke verwendet.

`scalefac_scale[granule][channel]`: Dieses Bit legt fest, ob die Skalierungsfaktoren eines Rahmens logarithmisch zur Basis 2 oder zur Basis  $\sqrt{2}$  interpretiert werden. Entsprechend wird in (7.1) der Multiplikator  $\lambda$  auf 1 oder 0.5 gesetzt.

`preflag[granule][channel]` (in (7.1)  $\delta_{\text{preflag}}$ ): Legt fest, ob die Skalierungsfaktoren der höheren Subbänder zusätzlich verstärkt werden. Der Verstärkungsfaktor  $\lambda_{\text{pretab}}$  wird in Abhängigkeit des Skalierungsfaktorbands gemäß Tabelle B.6 aus [18] bestimmt.

Der Wert  $s_{\ell}$  in (7.1) ist der Skalierungsfaktor des  $\ell$ -ten Skalierungsfaktorbands.

Zunächst bemerkt man, daß (7.1) eine feste Bitallokierung für den Gültigkeitsbereich eines Skalierungsfaktors impliziert. Bei kurzen Blöcken wird dies verfeinert zu einer gemeinsamen Bitallokierung pro Skalierungsfaktorband und Subblock. Die zur Verfügung stehende Bitbreite ergibt sich nun für jeden dieser Blöcke aus  $F$  und der Größenordnung von  $q$ . Hierzu kann in Analogie zu Layer II zunächst die durch  $F$  gewonnene Darstellungsgenauigkeit in Bits in Abhängigkeit der globalen Parameter und der Skalierung bestimmt werden. Die Größenordnung von  $q$  bestimmt dann die genaue Einbettungskapazität. Zur Verwendung von Einbettungsregionen kann hier die allgemeine Einbettungsform aus Kapitel 4 verwendet werden.

Wird Direkteinbettung verwendet, so kann z.B. eine Einbettungskapazität abhängig von der Größenordnung von  $s(q)$  vorher in Tabellen festgelegt werden. In jedem Fall ist die genaue zur Verfügung stehende Kapazität erst nach der Rückquantisierung bekannt.

### 7.3 Einzubettende Coderinformationen

Die zur erneuten Encodierung benötigten Coderparameter teilen sich auf in solche, die *vor* dem MDCT-Schritt benötigt werden und solche, die erst *danach* benötigt werden.

Die ersteren sind:

**window\_switching\_flag:** Gibt an, ob kurze oder lange Blöcke verwendet werden.

**block\_type:** Gibt an, ob es sich um einen *start-*, *stop-* oder *short-*Block handelt.

**mixed\_block\_flag:** Gibt an, ob die unteren und oberen Subbänder verschiedene Blocklängen verwenden (siehe obige Ausnahme).

Insgesamt werden lediglich 4 Bits benötigt, um diese Informationen zu codieren.

Nach dem MDCT-Schritt werden die restlichen Seiteninformationen benötigt:

**part2\_2\_length:** Anzahl Bits, die für Skalierungsfaktoren und Huffman-codierte Worte verwendet werden. Die Information kann bei einer wiederholten Encodierung als Anhaltspunkt für die tatsächliche Bitrate pro Granule bei früheren Generationen dienen.

**big\_values:** Bereich der Spektrallinien mit Betrag größer eins.

**global\_gain:** Globale Bitrate.

**scalefac\_compress:** Bitbreite der Skalierungsfaktoren.

**table\_select:** Ausgewählte Huffman-Tabellen - zwei bei kurzen Blocks, drei bei langen.

**subblock\_gain:** Verstärkungsfaktor eines Subblocks bei kurzen Blöcken (s.o.).

**region0\_count:** Breite des Spektrallinienbereichs (angegeben in Skalierungsfaktorbändern), der mit dem ersten Huffman-Code codiert wird. Gerechnet wird beginnend mit dem ersten Skalierungsband.

**region1\_count:** Wie oben für den Bereich des zweiten Huffman-Codes.

**preflag:** Verstärkung der höheren Skalierungsfaktoren (s.o.).

**scalefac\_scale:** Logarithmische Basis der Skalierungsfaktoren (s.o.).

**count1table\_select:** Auswahl der Huffman-Tabelle für den Bereich der Spektrallinien mit Betrag kleiner eins.

Insgesamt werden zur Codierung dieser Informationen 122 Bits im Monomodus und 244 Bits im Stereomodus benötigt. Addiert man hierzu den Bitbedarf für die **private\_bits** und den Pointer auf **main\_data\_begin**, die beide für unsere Zwecke nicht benötigt werden, erhält man den im Standard angegebenen Bitbedarf von 136 Bits im Monomodus und 256 Bits im Stereomodus.

In der obigen Aufzählung sind noch nicht die Skalierungsfaktoren enthalten, die ebenfalls encoderseitig benötigt werden und somit eingebettet werden müssen. Die Anzahl der zu verwendenden Skalierungsfaktoren bestimmt sich wie oben beschrieben abhängig von der Skalierungsfaktorauswahl und der Blocklänge. Die Anzahl Bits, die pro Skalierungsfaktor aufgewendet werden, bestimmt sich aus dem oben aufgeführten Parameter `scalefac_compress`.

## 7.4 Mehrstufige Einbettung

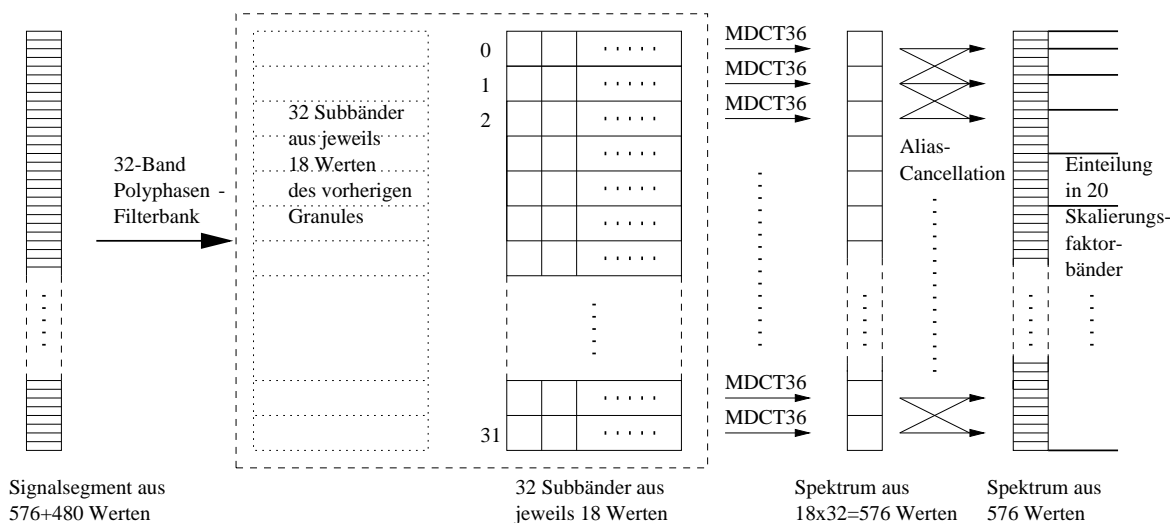


Abbildung 7.3: Überblick über die Transformation innerhalb eines Granules im Falle langer MDCT-Blöcke.

Die beiden wesentlich verschiedenen auftretenden Transformationsarten sind in Abb. 7.3 und Abb. 7.4 zusammengefasst. Die einzelnen Transformationsschritte sind in Pfeilrichtung von links nach rechts auszuführen. Zu beachten ist, daß die Subbandblöcke des vorherigen Granules zeitlich *vor* die Werte des aktuellen Granules gesetzt werden. Der Alias-Cancellation Schritt entspricht der obigen Transformation  $T$ . Die in Abb. 7.3 angegebenen Butterflies geben hier deren grobe Wirkungsweise wieder bei der jeweils zwei Spektrallinien benachbarter Blöcke kombiniert werden.

Im Fall langer Blöcke gibt es prinzipiell drei mögliche Stellen in der Transformationskette  $TMS$ , an denen eine Einbettung vorgenommen werden kann ( $T$  ist hier die Alias-Cancellation Matrix,  $M$  eine Matrix, die die MDCT-Operationen zusammenfaßt und  $S$  die 32-Band Filterbank auf dem entsprechenden Block. Die Matrizen  $S$  und  $M$  sind theoretisch jeweils zweiseitig unendlich und aus Produkten von Transformations- und Permutationsmatrizen gebildet. Um die Darstellung nicht unnötig zu verkomplizieren wird hier keine explizite Form angegeben.):

1. In das Spektrum  $TMSx$ . Die Einbettung auf dieser Ebene ermöglicht die präziseste Einhaltung psychoakustischer Kriterien, da die aliaskorrigierte Spektraldarstellung am genauesten der in der psychoakustischen Analyse verwendeten Fourier-Spektraldarstellung entspricht. Naheliegende Einbettungsblöcke sind in diesem Fall die Skalierungsfaktorbänder.

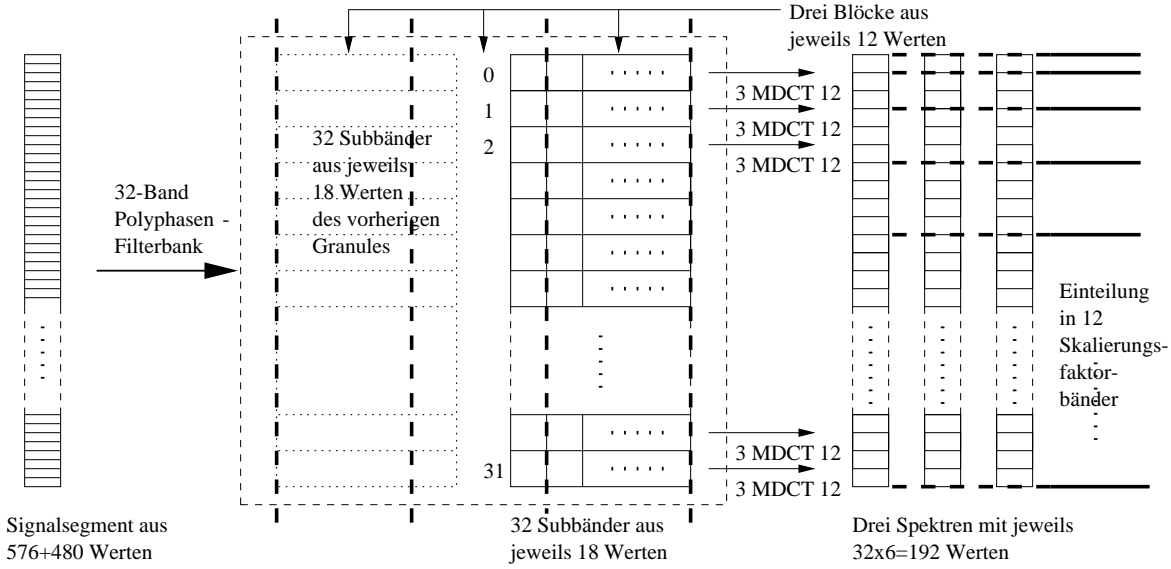


Abbildung 7.4: Überblick über die Transformation innerhalb eines Granules im Falle kurzer MDCT-Blöcke.

- In die Spektren  $T^{-1}TMSx$ . Einbettungsblöcke wären in diesem Fall die MDCT-Blöcke. Die Funktionsweise der MDCT läßt sich über Block-Toeplitzmatrizen beschreiben. Die MDCT-Operation für ein Subband und aufeinanderfolgende Granules wird beschrieben durch eine Block-Toeplitzmatrix  $\tilde{M}$  mit Toeplitzblöcken  $A_{i,i} =: A$  und  $A_{i,i+1} =: B$  und  $A, B \in M_{18}(\mathbb{R})$ . Alle anderen Einträge von  $\tilde{M}$  verschwinden. Das Subbandsignal  $s = (\dots s_0, s_1, s_2, \dots)^\top$  ist in die den Granules entsprechende Blöcke  $s_i \in \mathbb{R}^{18}$  aufgeteilt. Die MDCT-Transformation ist dann  $\tilde{M}s = (\dots \tilde{s}_0, \tilde{s}_1, \tilde{s}_2, \dots)^\top$  wo  $\tilde{s}_i = As_i + Bs_{i+1}$ . Setzt man  $B^\top A = A^\top B = 0$  und  $A^\top A + B^\top B = I_{18}$  voraus (für vorliegende MDCT ist dies bis auf einen Faktor erfüllt) und berechnet  $\tilde{M}^\top \tilde{s} = (\dots B^\top \tilde{s}_{i-1} + A^\top \tilde{s}_i \dots)^\top$ , so erhält man

$$\begin{aligned} B^\top \tilde{s}_{i-1} + A^\top \tilde{s}_i &= B^\top As_{i-1} + B^\top Bs_i + A^\top As_i + A^\top Bs_{i+1} \\ &= (A^\top A + B^\top B)s_i = s_i \end{aligned}$$

also  $\tilde{M}^\top \tilde{s} = s$ . Die Einbettung funktioniert aufgrund der Tatsache, daß für die verwendeten MDCTs zusätzlich  $AB^\top = BA^\top = 0$  gilt. Bettet man z.B. durch

$$\tilde{s}_i \mapsto \tilde{s}_i + \varepsilon =: x \quad \tilde{s}_{i+1} \mapsto \tilde{s}_{i+1} + \delta =: y$$

in zwei aufeinanderfolgende Granules eines Subbands ein, und transformiert via  $\tilde{x} := \tilde{M}^{-1}(\dots \tilde{s}_{i-1}, x, y, \tilde{s}_{i+2}, \dots)^\top$  zurück, so erhält man wegen  $AB^\top = BA^\top = 0$  und damit  $AA^\top A = A$  und  $BB^\top B = B$  sofort  $\tilde{M}\tilde{M}^\top = I$  und somit

$$\tilde{M}\tilde{x} = (\dots \tilde{s}_{i-1}, x, y, \tilde{s}_{i+2}, \dots)^\top.$$

Bei dieser Einbettungsart ist vorab eine Umrechnung der Skalierungsbänder auf die jeweiligen MDCT-Blöcke vorzunehmen.

3. Einbettung in die Subbänder. Dies stellt die Standardvorgehensweise der hier vorgestellten Layer II-Realisierung dar. Die Umrechnung der Skalierungsblöcke auf die 32 Subbänder verläuft analog zur MDCT-Einbettung.

Soll die Einbettung mehrstufig angewendet werden, also an zwei oder mehr der obigen Stellen erfolgen, so ist darauf zu achten, daß nachfolgende Einbettungen gemäß den Anforderungen von Kapitel 5 nur auf solchen Teile oder Koordinaten des Spektrums bzw. nur auf solchen Subbänder erfolgen, die vorher noch nicht zur Einbettung verwendet wurden. Somit ist, insbesondere wenn als Einbettungsstelle die erste Alternative gewählt wird, zu bestimmen, welche Subbänder die Teile des Bildes von  $M^{-1}T^{-1}$  enthalten, in die eingebettet wurde.

Im Falle kurzer Blöcke entfällt die Transformation  $T$  zur Alias-Cancellation. Daher kommen als Einbettungsstellen nur die beiden letztgenannten in Frage. Hier ist allerdings zu beachten, daß sich die Skalierungsfaktorbänder nun aus elf Frequenzbereichen für jeden der drei Blöcke zusammensetzen.

Insgesamt sind z.B. folgende zwei Einbettungsstrategien denkbar. Dabei sei  $s := TMSx$  der Spektralblock eines Frames.

1. Einbettung der Information über den Blocktyp und den Mixed-Block Modus in die Subbänder, d.h. in  $M^{-1}T^{-1}s$ , und Einbettung aller restlichen Parameter in  $s$ . Diese Vorgehensweise hat den Vorteil, daß ein encoderseitiger Detektor zunächst die korrekte Blocklänge feststellen kann.
2. Einbettung aller Parameter in  $s$ . Dies hat die oben erwähnten Vorteile, bringt aber den Nachteil mit sich, daß der Detektor zunächst probeweise die Transformationen für die verschiedenen Blocktypen durchführen muß.

## 7.5 Problematiken der Direkteinbettung

Bei Verwendung der Direkteinbettung kommt es wie in Kapitel 6 gesehen häufig dazu, daß sich die Subbandsignale und somit gegebenenfalls die Codeworte nach der Quantisierung bzw. Skalierung leicht ändern. Dies führt bei der hier verwendeten Huffman-Codierung dazu, daß bei nachfolgenden Generationen nicht mehr exakt dieselbe Bitrate erreicht werden kann. Kleine Schwankungen können hier aufgrund der variablen Bitrate zwar ausgeglichen werden, jedoch unter Umständen keine starken. So kann es dazu kommen, daß ein ständiger Mehrbedarf an Bits in einigen aufeinanderfolgenden Granules dazu führt, daß die extrahierten Encoderparameter nicht unter Einhaltung der Bitratentoleranz des Bitreservoirs verwertbar sind. Zur Vermeidung diese Problems sind mehrere Vorgehensweisen denkbar:

- In den Granules, in denen dieses Problem auftritt, kann nach der Standardmethode komprimiert werden.
- Hybrid-Ansatz: Es wird versucht, die wichtigsten Parameter (z.B. die Gainwerte der Skalierungsfaktorbänder) beizubehalten, und durch leichte Abänderungen anderer Parameter die zulässige Bitrate zu erreichen. Z.B. kann versucht werden, das Granule mit einem der anderen Codebücher zu codieren.
- Der Arithmetikfehler bzw. die Veränderung der Codeworte macht sich in der Größenordnung der Codeworte (interpretiert als Ganzzahlwert) nur wenig bemerkbar. Gegebenenfalls kann aufgrund dieser Feststellung ein passendes alternatives Codebuch automatisch ermittelt werden.



# Kapitel 8

## Anwendungen und Erweiterungen der vorgestellten Verfahren

Die vorgestellten Verfahren sind im Zusammenhang mit einer Vielzahl von Anwendungen einsetzbar. In diesem Kapitel werden sowohl einige unmittelbar verwandte als auch einige neue Anwendungen aus dem Bereich der Audiosignalverarbeitung aufgezeigt und Lösungsansätze unter Verwendung der vorgestellten Verfahren skizziert. Weiterhin wird auf Erweiterungen der vorgestellten Codecs im Hinblick auf deren Einsatz im Zusammenhang mit anderen gebräuchlichen Audiocodecs eingegangen.

### 8.1 Erweiterungen der Generationen-Codecs

Zu Verbesserungen und Erweiterungen der Generationencodecs beginnen momentan mehrere Diplomarbeiten. Neben der in Kapitel 7 beschriebenen Umsetzung des Direkteinbettungscodecs auf den Layer III von MPEG-1 und damit der Erstellung einer „High-Quality“-Variante [50], geht es dabei um die im folgenden beschriebenen Projekte.

#### 8.1.1 Bitreservoitechnik und Fehlerregionencodec

Die Konzipierung und Integration einer Bitreservoir-Technik in den Direkteinbettungscodec [12] ist eine wichtige Aufgabe. Die in Kapitel 6 untersuchten Analyseergebnisse zeigen deutlich, daß die potentielle Einbettungskapazität an vielen Stellen des Signals höher als der tatsächliche Bitbedarf ist. Abb. 8.1 zeigt das Konzept der Bitreservoirtechnik. In dieser

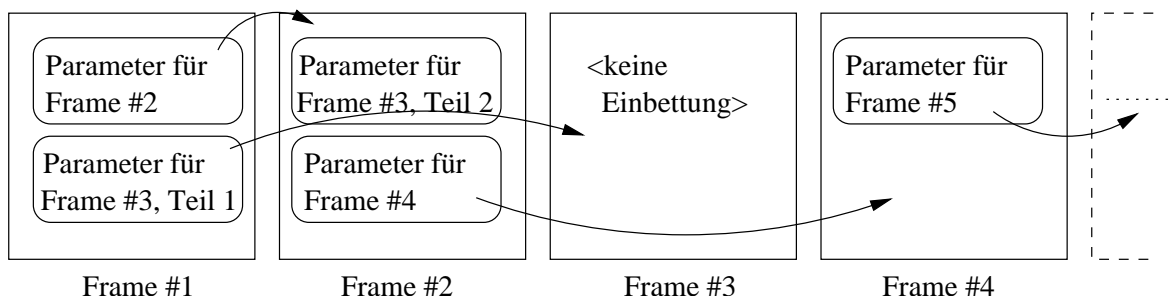


Abbildung 8.1: Bitreservoirtechnik mit Mindestversatz um einen Datenrahmen.

Variante besteht ein Mindestversatz von einem Datenrahmen zwischen eingebetteten Encoderinformationen und dem jeweiligen diese Informationen benötigenden Datenrahmen. Auf diese Weise kann encoderseitig, unter Verwendung des Algorithmus zur Einbettungsbitallokierung, aus den vorhandenen Daten bestimmt werden, wo die Einbettungsstellen sind und welche Einbettungsbreite der Decoder verwendet hat. So kann die Verwendung von Einbettungsmarkierungen auf bestimmte Startrahmen reduziert werden. Solche Startrahmen werden zur Initialisierung des Verfahrens benötigt.

Mit Hilfe der Bitreservoirtechnik kann ein Einbettungsregionencodex besonders elegant realisiert werden. Bei Verwendung der Standardtechnik der blockweisen Verarbeitung ohne Bitreservoir existieren durch die große Anzahl möglicher Rekonstruktionslevels (Kombination aus Bitallokierung und Skalierungsfaktoren) stark erschwerte Bedingungen auf Seiten des Einbettungsdetektors. Da bei Verwendung der Bitreservoirtechnik mit einem Mindestversatz von einem Datenrahmen die Rekonstruktionslevels zum Zeitpunkt der Verarbeitung eines Rahmens bekannt sind, wird die Aufgabe des Detektors stark vereinfacht.

### 8.1.2 Reduktion von Arithmetik- und NPR-Fehlern

Ähnlich dem durch die Einbettungsmarkierungen entstehenden Overhead ist der durch die Rekonstruktionsfehler entstehende Bedarf einer fehlerkorrigierenden Codierung der eingebetteten Werte beträchtlich. Darum geht es in einem weiteren Projekt um den Entwurf eines dem MPEG-1, Layer II, ähnlichen Codex unter Wegfall bzw. deutlicher Reduktion der Rekonstruktionsfehler [41]. Hierzu wird der Einsatz neuer PR-Filterbänke bzw. NPR Filterbänke mit viel geringerem Rekonstruktionsfehler [43] in Verbindung mit MDCT- und MLT-Filterbänken untersucht. Neben den MLTs erlauben auch ELTs (Extended Lapped Transforms) nach Malvar perfekte Rekonstruktion [39].

Rekonstruktionsfehler, wie sie innerhalb des MPEG-Verfahrens auftreten, lassen sich, da sie im Prinzip decoderseitig vorhersagbar sind, durch einen einfachen Trick unter Verwendung einer geometrischen Reihe reduzieren. Für die Rücktransformation  $T$  und die Hintransformation  $\tilde{T}$  gilt  $\tilde{T}T = I + E$  wo  $|E| < 1$ . Daher gilt

$$(I + E)^{-1} = I - E + E^2 - E^3 + \dots$$

und man kann  $\tilde{T}T(I + E)^{-1} = I$  schreiben. Die Approximation  $(I + E)^{-1} = I - E + O(E^2)$  liefert mit  $S := I - E$  die neue Rücktransformation  $TS$  und  $\tilde{T}(TS) \approx I$  mit einer im Vergleich zu  $\tilde{T}T$  quadratisch verbesserten Genauigkeit. Der Nachteil dieses Vorgehens ist, daß über die rekonstruierten Zeitsignale  $TS$ s keine Aussage getroffen und die Rekonstruktionsfilterbank  $T$  verändert wird.

### 8.1.3 Stabilität gegenüber Signalverarbeitungsoperationen

Die vorgestellten Verfahren bieten der eingebetteten Information fast keinen Schutz gegen Zerstörung durch Signalverarbeitungsoperationen wie Filterung, Abstratenänderung oder auch nur Veränderung der Lautstärke. Bei bestimmten Operationen ist dies auch zu rechtfertigen, da Signalveränderungen leicht dazu führen können, daß die eingebetteten Coderparameter nicht mehr zum (veränderten) Signal passen. Bei einigen Operationen, wie Abstratenänderung, leichter Lautstärkeänderung, dem Zusammenschneiden mehrere Stücke durch Aneinanderfügen bzw. Überblenden oder Tiefpaßfilterung sind jedoch Teile der eingebetteten Coderparameter sinnvoll auf das modifizierte Signal anwendbar. Im Falle des Zusam-

menschneidens durch Aneinanderfügen bzw. Überblenden geht die eingebettete Information aufgrund ihrer Lokalität bzw. Rahmenbezogenheit nur an den Übergangsstellen verloren. Verfahren zur Einbringung digitaler Wasserzeichen bieten einen großen Schutz der Wasserzeichen gegen eine Vielzahl denkbarer Signalveränderungen bis hin zur Komprimierung [3, 11, 63]. Dies wird allerdings auf Kosten großer Redundanz erreicht und ist mit der von den vorgestellten Verfahren benötigten großen Einbettungskapazität nicht oder nur schwer vereinbar. Eine mögliche Lösung besteht in der Reduktion der einzubettenden Information. Anhand der MPEG-1-Realisierung wurde versucht, lediglich die Allokierungsinformation und die Skalierungsfaktorauswahl zu übertragen. Die Skalierungsfaktoren wurden encoderseitig neu berechnet. Bei diesem Vorgehen war allerdings eine deutliche Qualitätsverschlechterung der betrachteten Signale festzustellen, was bestätigt, daß die durch die Skalierungsfaktoren gewonnene Darstellungsgenauigkeit für das Funktionieren des MPEG-Verfahrens sehr wichtig ist.

Eine Möglichkeit zur Umgehung der Übertragung der Skalierungsfaktoren wäre eine decoderseitige Anpassung der Signalamplituden innerhalb der Skalierungsbänder auf einen Pegel, der encoderseitig den gewünschten Skalierungsfaktor liefert. Da die Rekonstruktionslevels aufgrund der Quantisierung jedoch bereits im Vergleich zum Original geändert wurden, ist die Funktionsfähigkeit dieser Methode nicht garantiert.

#### 8.1.4 Vorkomprimierung

Liegt ein Zeitsignal mit eingebetteter Encoderinformation vor, so müssen diese Parameter bei wiederholter Encodierung nicht erneut berechnet werden. Erste Tests zeigen, daß im Falle des Layer II-Codex bis zu 50% schneller komprimiert werden kann. Bei der vorliegenden Implementation ist dieser Geschwindigkeitsgewinn allerdings davon abhängig, ob eingebettet werden konnte oder nicht. Bei wesentlich komplexeren Encodervorgängen wie etwa im Falle eines Layer III-Codex könnte mit Hilfe eines „vorkomprimierten“ Signals die erneute Kompression sogar auf langsameren Rechnern in Echtzeit erfolgen. Sogar wesentlich komplexere Kompressionsalgorithmen könnten somit praktikabel werden.

## 8.2 Adaptation auf andere Audiocodecs

Es soll kurz auf die mögliche Anpassung der vorgestellten Verfahren auf andere Audiocodecs eingegangen werden.

### 8.2.1 ATRAC

Das ATRAC-Verfahren von Sony wird im MiniDisc-System verwendet und erlaubt eine Kompressionsrate von 1 : 5. Die verwendete Hybrid-Filterbankstruktur ist in Abb. 8.2 dargestellt. Die zweistufige QMF-Filterbank führt eine Zerlegung in zwei niedrige und ein hohes Subband durch. Das hohe Subband wird zwecks Zeitsynchronizität geeignet verzögert. Nachgeschaltet ist eine MDCT pro Subband, wobei ein psychoakustisches Modell über die Zeitauflösung entscheidet und zwischen langen und kurzen Blocklängen wählt. Es entstehen insgesamt 512 Spektralwerte, die in 52 Block floating units (BFU) gruppiert, pro BFU skaliert und schließlich mit einem Quantisierer pro BFU codiert werden. Die einzubettende Information pro Datenrahmen besteht aus

- MDCT Blocklänge,

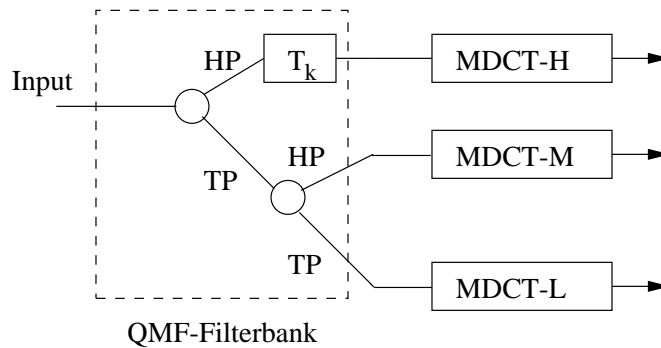


Abbildung 8.2: Im ATRAC-Verfahren verwendete Filterbankstruktur.

- Bitallokierung pro BFU und
- Skalierungsfaktoren pro BFU.

### 8.2.2 PASC

Precision Adaptive Subband Coding (PASC) wird in DCC-Recordern verwendet. DCC-Recorder arbeiten mit einem analogen und einem digitalen Lesekopf und sind kompatibel zur analogen Kompaktcassette. Um mit der verfügbaren Bitkapazität zur digitalen Speicherung auszukommen, wird eine Datenreduktion um den Faktor 1 : 4 vorgenommen. Hierzu wird eine vereinfachte Variante eines MPEG-1 Layer I Codecs verwendet, die perzeptuell transparente Qualität liefert. Der erzeugte Code ist MPEG-1 Layer I kompatibel, daher kann das Verfahren zur Vermeidung von Generationseffekten entsprechend angewendet werden. DCC-Recorder stellen innerhalb eines digitalen Subcodes eine Datenrate von ca. 400 Zeichen pro Sekunde für Sekundärinformationen wie Texte oder Musiktitel zur Verfügung. Diese Subcodeinformationen können mit Hilfe der vorgestellten Einbettungsverfahren zusätzlich zu den Encoderparametern eingebettet werden.

### 8.2.3 AC-2/AC-3

AC-2 Coder von Dolby erreichen unter Verwendung einer MDCT/MDST-Blocktransformation mit 50% Überlappung Kompressionsraten von 1 : 5.5 bis 1 : 6.1. Die Blocklängen werden wiederum adaptiv gewählt, die zugehörigen Fensterfunktionen sind proprietär. AC-3 ist eine mehrkanaltaugliche Erweiterung, die 5.1 Surround-Signale auf eine Datenrate von 320 kbps komprimieren kann. AC-3 wird unter anderem innerhalb des amerikanischen High-Definition Television (HDTV) verwendet.

Ähnlich der Skalierung im MPEG-Verfahren wird das MDCT-Spektrum bei AC-3 in Exponenten und Mantissen aufgeteilt, die getrennt behandelt werden. Die codierten Exponenten beschreiben die spektrale Hüllkurve und werden differentiell, abhängig vom Signalcharakter (stationär/transient) übertragen. Bei Bedarf werden in hochfrequenten Signalanteilen Redundanzen innerhalb der verschiedenen Kanäle ausgenutzt (Coupling). Die relevante einzubettende Encoderinformation besteht aus der Codierung der spektralen Hüllkurve, der Coupling-Information und der Information über den Signalcharakter. Weiterhin werden wie gewohnt die Codeworte übertragen.

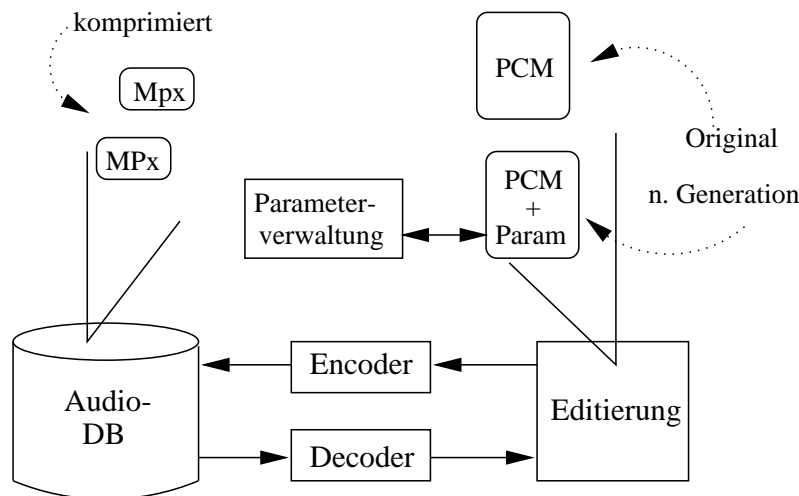


Abbildung 8.3: Editiersystem, Dateitypen und Organisation der Coderparameter.

#### 8.2.4 MPEG-2 AAC

Das *Main Profile* des MPEG-2 AAC verwendet eine MDCT mit variabler Blocklänge von 256 oder 2048 Werten. Zwischen den verschiedenen Blocklängen wird signaladaptiv gewechselt. Weiterhin kann das zur MDCT gehörige Fenster parametrisch angepaßt werden. Das Fenster kann dabei zwischen einem Sinusfenster und einem Kaiser-Bessel-Fenster variiert werden. Letzteres liefert eine bessere Stopbanddämpfung zu höheren Bändern hin. Bei Verwendung langer Transformationsblöcke werden die Werte des MDCT-Spektrums rückwärtsadaptiv mit Hilfe eines Prädiktors zweiter Ordnung übertragen. Zur Quantisierung wird die nichtlineare Quantisierung aus dem MPEG-1 Standard (Layer III) verwendet. Innerhalb der Bitallokierungsschleife wird das Quantisierungsrauschen durch die Wahl entsprechender Skalierungsfaktoren gesteuert (Noise shaping). Als letzter Codierschritt wird eine verlustfreie Komprimierung durch geschicktes Umordnen der Codeworte mit anschließender Huffman-Codierung durchgeführt. Als Erweiterung des Grundverfahrens sind zusätzlich Joint-Stereo Coding und Temporal Noise Shaping (TNS) möglich [4].

Relevante Encoderparameter sind somit die MDCT-Blocklänge, Quantisierer- bzw. Skalierungsinformation sowie gegebenenfalls die Codebuchauswahl.

### 8.3 Editiersysteme

Die bereits einleitend erwähnten Editiersysteme mit komprimiert gespeicherten bzw. verwalteten Audiodateien sind eine attraktive Anwendungsmöglichkeit im Hinblick auf Arbeitsplätze mit minderer Netzanbindung oder geringer lokaler Speicherkapazität. Abb. 8.3 zeigt das Editierszenario ergänzt um anfallende Datensätze. Auf der Seite der Datenbank sind dies komprimierte Dateien (angedeutet als Dateien im MPEG-Format), auf Seiten der Editierung sind dies (dekomprimierte) PCM-Dateien. Werden dekomprimierte Datensätze lokal verwaltet, ist eine Einbettung der Coderparameter prinzipiell nicht notwendig, und die Verwaltung dieser Parameter kann von einer externen Einheit übernommen werden. Es stellen sich einige fundamentale Fragen:

- Bei welchen Editieroperationen entstehen Artefakte bzw. Generationseffekte?
- Wie ist das (einfache) Konzept der Generationen auf *Editiergenerationen* übertragbar? Hier ist die Abhängigkeit der Generationenbildung von u.U. *mehreren* Dateien und verschiedenen Editieroperationen zu berücksichtigen.
- Wie können die Coderparameter geschickt kombiniert werden, wenn zwei oder mehrere Signale zusammengemischt oder -geschnitten werden, so daß keine Artefakte entstehen?
- Für welche Editieroperationen ist eine solche Kombination überhaupt möglich bzw. nötig und für welche nicht?

Die Modifikation der Coderparameter in Abhängigkeit von einer gegebenen Editierung steht dabei in engem Zusammenhang mit der Stabilität der Einbettung gegenüber Signalveränderungen. Insgesamt sind zur Erstellung von Editiersystemen in obigem Stil somit noch zahlreiche Aufgaben zu bewältigen.

## 8.4 Audio-Vertextung und Partiturintegration

Aufbauend auf die vorgestellten Einbettungsverfahren soll in einem weiteren Projekt ein Verfahren zur Einbettung, Übermittlung und Decodierung bestimmter kontextbasierter Informationen innerhalb von PCM-Audiodaten entwickelt werden [45]. Speziell geht es darum,

1. innerhalb PCM-codierter Musiksignale mit Vokalanteil (meist Gesang) die Textinformation bzw. die Liedtexte,
2. innerhalb PCM-codierter Musiksignale die Partiturninformation, d.h. Noten und gegebenenfalls die Spielweise und
3. Kombinationen von beidem

zu übertragen. Eine wesentliche Eigenschaft soll hierbei die *Synchronizität* von textuellen Informationen und digitalen Audiodaten sein. Schwerpunkte sind die geeignete Anpassung der Einbettungsverfahren und die Entwicklung sowie Realisierung einer Plattform zur synchronen Wiedergabe von Audio- und Sekundärdaten.

Anwendungsbereiche sind überall dort anzusiedeln, wo digitale Audiodaten in unkomprimierter Form verwendet werden, weiterhin dort wo es auf eine synchronisierte Übertragung oder Wiedergabe von Audiodaten und Sekundärinformationen ankommt.

Zunächst wird hierbei davon ausgegangen, daß bei der Ersteinbettung der Sekundärinformation die Synchronisation manuell durchgeführt wird. Tatsächlich ist eine vollautomatische Synchronisation, wenn überhaupt, so zumindest nach dem heutigen Stand der Technik nicht realisierbar. Ein weiterführendes Problem ist daher die halbautomatisierte Synchronisation. Innerhalb des Projekts sollen daher außerdem erste Methoden und Ideen in Richtung einer halbautomatisierten Synchronisierung im Falle der textuellen Sekundärinformation untersucht werden.

Ein großer Vorteil der Einbettung in PCM-Dateien ist die Unabhängigkeit von Hardwareformaten. Die praktische Relevanz einer solchen Datenübermittlung wird durch die in den verschiedenen Standards bereitgestellten (in diesem Falle formatabhängigen) Möglichkeiten für eine Übermittlung von Sekundärinformationen deutlich. Das Beispiel der DCC wurde

oben bereits erwähnt. Ein weiteres Beispiel ist die Codierung von Information innerhalb der freien Subcode-Bits von Audio-CDs. Ansätze wie CD+G oder CD+MIDI [48] haben sich zwar kaum durchgesetzt, jedoch sind in letzter Zeit bei einigen größeren Firmen Bestrebungen zu erkennen, ein CD+Text-Format zu etablieren [2]. Die Subcodes bieten eine theoretische Kapazität von ca. 5400 Bytes/Sek., was ca. 141 Bytes pro MPEG-Rahmen entspricht. Die in den Einbettungsverfahren dieser Arbeit festgestellten Kapazitäten sind im Mittel ähnlich, jedoch naturgemäß stark variierend. Die bisherigen CD+Text Entwicklungen bieten noch nicht die in der rahmenbasierten Methode implizit vorhandene Synchronizität, die zur simultanen Wiedergabe von Audio- und Textinformation wünschenswert ist. Die Firma Sony bietet jedoch bereits einen auf CD+Text aufgesetzten Mechanismus zur synchronisierten Speicherung von Informationen an [2].

## 8.5 Weitere Anwendungsgebiete

### 8.5.1 Bildkompression

Zum Zwecke des Watermarkings wurde Varianten der Einbettungstechnik auf Bilddaten angewendet [11, 63]. Direkt- und Fehlerregioneneinbettung sind leicht innerhalb gängiger Kompressionsverfahren anwendbar. Z.B. arbeitet JPEG-basierte Bildkompression [15], wie sie auch in den I-Frames des MPEG-1 Video-Standards [17] verwendet wird, ebenfalls transformationsbasiert. Ein Bild wird dabei in  $8 \times 8$ -Blöcke aufgeteilt, mit einer zweidimensionalen DCT transformiert, via Zig-Zag-Scan in einen eindimensionalen Datensatz überführt, quantisiert und anschließend lauffängen- und entropiecodiert. Die Quantisierung ist hierbei wiederum der einzige verlustbehaftete Verarbeitungsschritt und die Requantisierung die entsprechende Einbettungsstelle.

Die Einbettungstechnik ist ebenfalls auf die sehr erfolgreiche Zerotree- bzw. Embedded Zerotree Codierung (EZW-Coder von *Embedded Zerotree Wavelet-Coder*) anwendbar. Am Einbettungskonzept ändert sich dabei nichts, da das decoderseitige Resultat der EZW-Codierung wie üblich eine quantisierte Darstellung der Bilddaten ist. In seiner ursprünglichen Version [55] besteht die EZW-Codierung grob aus folgenden Komponenten:

- Diskrete Wavelettransformation oder andere hierarchische Transformation zur Dekorrelation der Bilddaten.
- Geschickte Codierung der signifikanten und insignifikanten Daten. Dabei werden die insignifikanten Daten pyramidenartig über die verschiedenen Hierarchiestufen organisiert (in den sogenannten *Zerotrees*), ausgehend von der Annahme, daß diese sich aufgrund von Selbstähnlichkeiten innerhalb von Bilddaten relativ häufig hierarchieübergreifend oder skalenübergreifend vorhersagen lassen.
- Übertragung der Codeworte mit inkrementeller Genauigkeit. Für eine Approximationsstufe werden dabei jeweils Signifikanzdaten und Bits für die signifikanten Koeffizienten übertragen. Somit kann ein Encoder nach jeder Approximationsstufe ein sinnvolles Bildsignal rekonstruieren. Die erhaltene Genauigkeit erhöht sich von Stufe zu Stufe.
- Adaptive arithmetische Codierung der Signifikanzinformation und Codeworte.

Da nach vollständiger Übertragung der Daten einer Approximationsstufe alle Koeffizienten denselben maximalen Rekonstruktionsfehler aufweisen, ist eine Anwendung des Einbettungs-

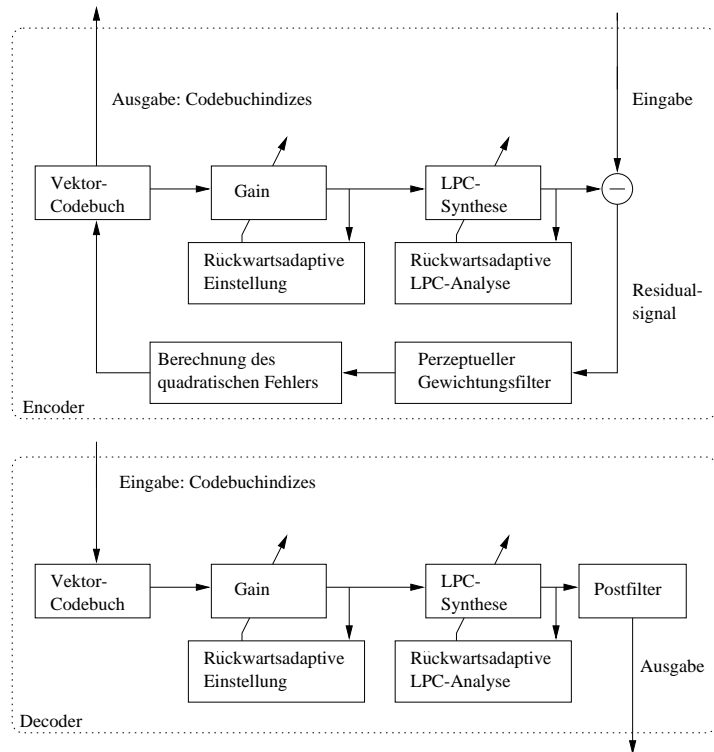


Abbildung 8.4: Schema eines LD-CELP Codecs nach ITU-T G.728.

verfahrens sogar besonders einfach. Für das EZW-Verfahren wurden in letzter Zeit zahlreiche Verbesserungsvorschläge gemacht, siehe z.B. [51].

### 8.5.2 Sprachverarbeitung und prädiktive Einbettung

Eine Erweiterung der Einbettungsmechanismen auf die z.B. in der Sprachcodierung sehr häufig verwendete linear prädiktive Codierung ist ebenfalls sehr interessant, insbesondere da hier das Anwendungsgebiet der Tandemcodierung angesiedelt ist. Es soll hier kurz auf das Beispiel des Low-Delay CELP Codecs G.728 [14] eingegangen werden. Abb. 8.4 zeigt schematisch den Datenfluß und die Verarbeitungseinheiten des LD-Celp. Als Eingabe werden linear quantisierte PCM-Signale verarbeitet. Die Ausgabe besteht aus Indizes eines im voraus bekannten Codebuchs. Die Verarbeitung geschieht in Rahmen der Länge von jeweils 20 Abtastwerten. Jeder Rahmen ist in vier Vektoren der Länge fünf unterteilt, und für jeden der Vektoren wird ein Codebuchindex ermittelt. Die Koeffizienten eines linearen Prädiktors sowie der Verstärkungsfaktor einer Gain-Einheit werden jeweils für die Verarbeitung eines Datenrahmens bestimmt. Dies geschieht *rückwärtsadaptiv* basierend auf dem Ergebnis der LPC-Synthese vorheriger Daten. Somit kann der Decoder unter Verwendung derselben Berechnungsvorschrift dieselben LPC- und Gain-Parameter berechnen. Diese müssen somit nicht zum Decoder übertragen werden. Anfangs sind beide Parameter geeignet zu initialisieren. Die Auswahl eines Codebuchvektors geschieht mittels eines Analyse-durch-Synthese Verfahrens unter Minimierung des quadratischen Fehlers des perzeptuell gewichteten Residualsignals. Das verwendete perzeptuelle Gewichtungsfilter wird aus Effizienzgründen in Implementierungen jeweils *vor* die Differenzbildung zwischen Original und prädiziertem Signal geschaltet.

Der Decoder arbeitet analog zur Syntheseeinheit des Encoders. Das nachgeschaltete Postfilter dient der Verstärkung bestimmter Frequenzbereiche und ermöglicht somit eine ausreichend gute Signalqualität bei bis zu dreimaliger Tandemcodierung.

Als Anregung für ein Verfahren zur vollständigen Vermeidung von Generationseffekten soll nun idealisierend angenommen werden, daß

- ein Eingangssignal  $n$ -ter Generation dem Encoder synchron zum Signal  $n - 1$ -ter Generation (falls vorhanden) vorliegt,
- Prädiktor und Gain-Einheit des Encoders beim Codieren des Signals  $n$ -ter Generation mit denselben Werten initialisiert werden, wie die vorherige Decoderstufe und
- das perzeptuelle Gewichtungsfiler so wie in Abb. 8.4 dargestellt realisiert ist.

Unter der weiteren Annahme, daß die LPC-Syntheseabbildung aus der Menge der Codebuchvektoren auf ein prädiertes Signal injektiv ist, gibt es dann genau einen Codebuchvektor, der das Residualsignal verschwinden läßt. Dieser Codebuchvektor ist dann genau der Codebuchvektor, den der CELP-Codec der vorherigen Generation für das codierte Signal  $n - 1$ -ter Generation berechnet hat. Eine Fortführung dieses Vorgehens liefert für alle Generationen denselben Code und somit eine Vermeidung der Generationseffekte. In höheren Generationen kann somit das perzeptuelle Gewichtungsfiler konzeptuell weggelassen werden. Zur Realisierung des vorgestellten Konzepts sind obige „Laborbedingungen“ in Verbindung mit der jeweiligen konkreten Implementierung abzustimmen.

Eine Dateneinbettung in der prädiktiven Codierung hängt prinzipiell vom Abstand der Bilder der Codebuchvektoren unter der jeweiligen LPC-Syntheseabbildung ab. Ein großer Abstand (im Sinne einer Minimaldistanz) bewirkt in diesem Fall eine große Einbettungskapazität. Ob eine Einbettung perzeptuell sinnvoll ist, kann wieder mit Hilfe von Einbettungsregioneninformation übermittelt werden. Die Einbettungsregionen können dabei als Regionen mit hinreichend kleinem perzeptuell gewichteten quadratischen Fehler gewählt werden. Da dieses ad hoc-Vorgehen vom Berechnungsaufwand jedoch sehr komplex ist, müssen hier noch stark vereinfachende Mechanismen, wie etwa die schnelle Bestimmung einer geeigneten Teilmenge von Einbettungsvektoren, gefunden werden.



## Kapitel 9

# Zusammenfassung und Diskussion

In der vorliegenden Arbeit wurden Generationseffekte, wie sie bei der wiederholten Komprimierung und Dekomprimierung digitaler Audiosignale entstehen, untersucht. Zur Vermeidung dieser Generationseffekte wurden zwei generische Varianten eines psychoakustischen Audiocodex vorgestellt, der Direkteinbettungscodec und der Fehlerregionencodex. Das Grundprinzip der Verfahren resultiert aus der Annahme, daß der durch die Quantisierung innerhalb eines Codex entstehende Informationsverlust dazu führt, daß nachfolgende Codexstufen die zur Encodierung benötigten Parameter nicht mehr sinnvoll aus dem Signal höherer Generation bestimmen können. Daher werden diesen Codexstufen die Decoderinformationen der jeweils vorherigen Codexstufen übermittelt. Die Übermittlung geschieht optional durch Einbettung innerhalb des dekomprimierten Audiosignals. Hierzu wurden in Kapitel 4 Einbettungsverfahren entwickelt, die nach psychoakustischen Prinzipien arbeiten und eine Einbettung ohne wahrnehmbaren Qualitätsverlust erlauben. Der Einbettungsregionencodex überträgt hierzu Tendenzbitinformationen vom Encoder zum Decoder. Hierdurch wird zwar die Kompressionsrate geringfügig gesenkt, im Gegenzug jedoch eine Einbettung ermöglicht, die unter gewissen Nebenbedingungen eine Reproduktion des komprimierten Audiosignals erster Generation in allen höheren Codexstufen erlaubt. Die Einbettung erfordert kein zusätzliches Datenformat, was insbesondere im Zusammenhang mit der Verwendung und Übermittlung der dekomprimierten Stücke auf Standardmedien (CD, DAT, PCM-Datei etc.) wichtig ist.

### Direkteinbettungscodec

Die Direkteinbettungsvariante nimmt mögliche kleinere Qualitätsverluste in Kauf, arbeitet jedoch ohne zusätzliche Tendenzbits, also mit der ursprünglichen Datenrate. Zu dieser Codexvariante wurde eine konkrete Realisierung innerhalb eines MPEG-1, Layer II Codex vorgestellt und untersucht. Hierzu wurde ein Hörtest mit 26 Personen herangezogen. Trotz der üblichen Kritikpunkte im Zusammenhang mit Hörtests zeigen die Ergebnisse deutliche Trends. So konnten von den Testhörer die Versionen erster Generation meist nicht von denen der 25. Generation unterschieden werden. Hingegen wurden die Versionen dritter Generation aller Stücke mit dem Standardverfahren schlechter bewertet, als die der ersten. Untersuchungen der Einbettungskapazität zeigen, daß die relevanten Informationen nicht immer eingebettet werden können. Es stellt sich jedoch heraus, daß an den Stellen, an denen keine Einbettung möglich ist, aufgrund der geringen Signalstärke bzw. dem tonalen Charakter des Signals keine signifikanten Generationseffekte zu verzeichnen sind. Dieses nichttriviale Resultat läßt sich aus der Funktionsweise des verwendeten MPEG-Coders erklären. Bitreservoirtechniken

können dafür sorgen, daß einzubettende Informationen aus Datenrahmen mit zu geringer Kapazität in Rahmen mit großer Kapazität eingebettet werden können.

### **Dateneinbettungsverfahren**

Die Einbettungsregionen beschreiben bei Verwendung eines geeigneten psychoakustischen Modells die psychoakustische Einbettungskapazität eines transformierten Signalblocks. Unter der Annahme eines scharfen psychoakustischen Modells, das exakt auf die verwendete Transformation angepaßt ist, wird hierdurch die maximale Einbettungskapazität beschrieben, die ohne eine perzeptuelle Störung des rekonstruierten Signals ausgenutzt werden kann. Natürlich ist die Kapazität abhängig von einer Maximierung des Volumens der Einbettungsregionen. Es soll nochmals betont werden, daß die Zielsetzung bei dieser Art der Einbettung deutlich verschieden von der Zielsetzung bei der Einbettung digitaler Wasserzeichen ist. Im Gegensatz dazu wird hier keine Art geheimen Schlüssels verwendet und auch nicht auf eine eventuelle Stabilität gegenüber schwerwiegenden Signalveränderungen (wie etwa Kompression) geachtet. Dafür wird eine sehr hohe Einbettungskapazität unter Vermeidung von Signalstörungen erreicht. Hier existieren, insbesondere aufgrund der möglichen zeitlichen Synchronizität, eine Reihe attraktiver Anwendungsbeispiele.

### **Kritikpunkte**

In der momentanen Version der Codecs sind nur unzureichende Mechanismen zur Behandlung von Übertragungsfehlern gegeben. Kleine Fehler können daher zum Versagen des Verfahrens führen. Ein prinzipieller Nachteil der Verfahrens ist die relativ große Menge einzubettender Information. Somit ist eine noch größere Anfälligkeit gegenüber Störungen gegeben.

Im Szenario des Editiersystems unter Verwendung einer Datenbank komprimierter Audiosignale gilt es auch, kritische Anmerkungen zu treffen. Vermutlich ist im Fall beliebiger Editieroperationen nicht immer eine Kombination oder Manipulation vorheriger Codeparameter möglich, so daß Generationseffekte in dieser Richtung nicht ausgeschlossen werden können. Hier gilt es also insbesondere, eine geeignete Untermenge von Operationen zu finden, für die dies nicht gilt.

### **Aufgeworfene Fragestellungen und Probleme**

Die systematische Vermeidung von Generationseffekten sowie die Verarbeitung von komprimierten Audiodaten in digitalen Editiersystemen steht noch in ihren Anfängen. Daher ergeben sich zahlreiche neue Fragestellungen und Probleme:

- Gibt es einen effizienten Algorithmus der, wie in Kapitel 5 angedeutet, ein Signal genau so rekonstruiert, daß ein nachfolgender Encoder automatisch die korrekten Encoderparameter bestimmen kann?
- Wie lassen sich die Degenerationseffekte der Coderparameter systematisch beschreiben und somit gegebenenfalls abschwächen?
- Gibt es eine geeignete Untermenge von Encoderparametern, so daß es ausreicht, nur diese Untermenge zur wiederholten Encodierung zu berücksichtigen?

- Wie lassen sich die vorgestellten Verfahren zur Verwendung in heterogenen Codec-kaskaden erweitern? Hier müßten geeignete Umrechnungsvorschriften der eingebetteten Encoderparameter gefunden werden.
- Wie läßt sich eine Robustheit der Einbettung gegenüber Übertragungsfehlern und leichten Verfälschungen erreichen? Dies ist auch im Hinblick auf eine Speicherung der Träger-signale auf Audio-CDs wichtig.
- Wie können für bestimmte Editier- oder Mischoperationen zweier (oder mehrerer) vormals komprimierter Audiostücke die Encoderparameter so kombiniert werden, daß das resultierende Signal möglichst keine Artefakte enthält? Einfachere Operationen wären z.B. eine Überlagerung von oder Überblendung zwischen zwei Stücken.

### Ausblick

Bezüglich weiterer Arbeiten sind zwei Schwerpunkte zu setzen. Im ersten geht es um Erweiterungen und Verfeinerungen des Codec-Systems. Neben ausführlichen Studien im Zusammenhang mit weiteren möglichen Kompressionsraten wird innerhalb der im letzten Kapitel beschriebenen Diplomarbeiten

- eine Layer III-Variante des Direkteinbettungscodecs realisiert,
- eine Codecvariante mit wesentlich geringeren Rekonstruktionsfehlern untersucht und
- ein Fehlerregionencodec in Verbindung mit einer Bitreservoirtechnik implementiert.

Weiterhin ist eine bezüglich Übertragungsfehlern robuste Codecversion ein wichtiges Ziel. Der zweite Schwerpunkt wird sich mit der Erstellung eines Editiersystems auf einem Datenbestand komprimierter Audiodateien beschäftigen. Hier werden zahlreiche Tests bezüglich Editieroperationen nötig sein. Erste Werkzeuge in dieser Richtung wurden bereits im Rahmen von Projektgruppen erstellt.

Zu den im Rahmen dieser Arbeit entwickelten Methoden wurden unter Förderung der DFG zwei Verfahren zur Patentierung eingereicht [34, 33]. Aufgaben im Umfeld dieser Patentierungen werden zukünftige Arbeiten ebenfalls beeinflussen.



# Literaturverzeichnis

- [1] H. BABOVSKY, T. BETH, H. NEUNZERT, AND M. SCHULZ-REESE, *Mathematische Methoden der Systemtheorie: Fourieranalysis*, no. 5 in *Mathematische Methoden in der Technik*, Teubner, 1987.
- [2] B. BEHR AND C. WIEDENHOFF, *Kuck mal, was da spielt*, c't Magazin für Computertechnik, 4 (1999), pp. 218–222.
- [3] L. BONEY, A. H. TEWFIK, AND K. N. HAMDY, *Digital watermarks for audio signals*, in *IEEE Int. Conf. on Multimedia Computing and Systems*, June 1996.
- [4] M. BOSI ET AL., *ISO/IEC MPEG-2 advanced audio coding*, *Journal of the Audio Engineering Society*, 45 (1997), pp. 789–814.
- [5] K. BRANDENBURG AND G. STOLL, *ISO-MPEG-1 Audio: A Generic Standard for Coding of High-Quality Digital Audio*, in *Collected Papers on Digital Audio Bit-Rate Reduction*, N. Gilchrist and C. Grewin, eds., *Audio Engineering Society*, Sept. 1996, pp. 31–42.
- [6] R. COIFMAN AND Y. MEYER, *Remarques sur l'analyse de Fourier à fenêtre*, *Comptes Rendus de l'Académie des Sciences*, 312 (1991), pp. 259–261. Série I.
- [7] I. J. COX, J. KILIAN, T. LEIGHTON, AND T. SHAMOON, *Secure Spread Spectrum Watermarking for Multimedia*, tech. rep., *NEC Research Institute*, 4 Independence Way, Princeton, NJ 08540, 1995.
- [8] I. DAUBECHIES, *Orthonormal bases of compactly supported wavelets*, *Comm. Pure Applied Math.*, XLI (1988), pp. 909–996.
- [9] J. R. DELLER, J. G. PROAKIS, AND J. H. L. HANSEN, *Discrete-Time Processing of Speech Signals*, *Macmillan*, 1993.
- [10] G. FOLLAND, *Real Analysis*, *Wiley and Sons*, 1984.
- [11] F. HARTUNG AND B. GIROD, *Digital watermarking of raw and compressed video*, in *Proc. SPIE*, 1996, pp. 205–213.
- [12] V. HASSENRIK, *Ein Audiocodierer zur Vermeidung von Generationseffekten aufbauend auf dem MPEG-1 Standard (ISO/IEC 11172-3)*. Diplomarbeit, *Institut für Informatik V*, *Universität Bonn*, 1999.
- [13] ———, *persönliche Mitteilung*, 1999.

- [14] INTERNATIONAL TELECOMMUNICATION UNION, *Coding of Speech at 16 kbit/s Using Low-Delay Code Excited Linear Prediction*, 1992. Recommendation G.728.
- [15] —, *Information Technology - Digital Compression and Coding of Continuous-Tone Still Images - Requirements and Guidelines*, 1992. Recommendation T.81.
- [16] ISO/IEC 11172, *Coding of moving pictures and associated audio for digital storage media at up to 1.5 mbits/s*, 1992. International Standard.
- [17] ISO/IEC 11172-1, *Coding of moving pictures and associated audio for digital storage media at up to 1.5 mbits/s- video part*, 1992. International Standard.
- [18] ISO/IEC 11172-3, *Coding of moving pictures and associated audio for digital storage media at up to 1.5 mbits/s- audio part*, 1992. International Standard.
- [19] ISO/IEC 13818, *Information technology - Generic coding of moving pictures and associated audio information*, 1994. International Standard.
- [20] ISO/IEC 13818-3, *Information technology - generic coding of moving pictures and associated audio, part 3: Audio*, 1994. International Standard.
- [21] ISO/IEC 13818-7, *Information technology - generic coding of moving pictures and associated audio, part 7: Advanced audio coding*, 1997. International Standard.
- [22] ISO/IEC 14496, *Information technology - Coding of moving pictures and audio*, Feb. 1999. International Standard.
- [23] ISO/IEC JTC1/SC29/WG11, *MPEG-7 Context and Objectives*, 1999. <http://drogo.cselt.stet.it/mpeg/standards/mpeg-7/mpeg-7.htm>.
- [24] —, *MPEG-7*, geplant für 2001.
- [25] N. IWAKAMI, T. MORIYA, AND S. MIKI, *High-Quality Audio-Coding at Less Than 64 kbit/s by Using Transform-Domain Weighted Interleave Vector Quantization (TWIN-VQ)*, in Proc. ICASSP, 1995, pp. 3095–3098.
- [26] N. JAYANT AND P. NOLL, *Digital Coding of Waveforms*, Prentice Hall, 1st ed., 1984.
- [27] J. D. JOHNSTON, *Estimation of Perceptual Entropy Using Noise Masking Criteria*, in Proc. ICASSP, 1988, pp. 2524–2527.
- [28] —, *Transform coding of audio signals using perceptual noise criteria*, IEEE Journal on Selected Areas in Communications, 6 (1988), pp. 314–323.
- [29] G. KAISER, *A Friendly Guide to Wavelets*, Birkhäuser, Boston, 1994.
- [30] M. KEYHL, H. POPP, E. EBERLEIN, K.-H. BRANDENBURG, H. GERHÄUSER, AND C. SCHMIDMER, *Verfahren zum kaskadierten Codieren und Decodieren von Audiodaten*, 1994. Patentschrift beim Deutschen Patentamt DE 4405659 C1.
- [31] F. KURTH, *Adaptive Wavelet Packets und Anwendungen in der Audiosignalverarbeitung*. Diplomarbeit, Institut für Informatik V, Universität Bonn, 1997.

- [32] ———, *On Rational Multirate Filter Banks*, IEEE Transactions on Signal Processing (zur Publikation eingereicht), (1998).
- [33] ———, *Verfahren zum unbemerkten Übertragen und/oder Speichern von Zusatzinformationen innerhalb eines Audiosignals*, 1999. Submitted to the German Patent Office.
- [34] F. KURTH AND M. CLAUSEN, *Ein Verfahren zum kaskadierten Codieren und Decodieren von Audiosignalen*, 1999. Submitted to the European Patent Office.
- [35] ———, *Filter bank tree and M-band wavelet packet algorithms in audio signal processing*, IEEE Trans. SP, 47 (1999), pp. 549–554.
- [36] E. LEHMANN, T. RIETHMÜLLER, AND H. STRASSBURG, *Das SoundBlaster-Profibuch*, Addison-Wesley, 1st ed., 1993.
- [37] I. LIM AND J. BAHN, *Fixed-point analysis and simulations of AC-3 algorithm*, in Proc. WASPAA 1997, Mohonk, N.Y., USA, 1997.
- [38] H. S. MALVAR, *Lapped transforms for efficient transform/subband coding*, IEEE Trans. ASSP, 38 (1990), pp. 969–978.
- [39] ———, *Signal Processing with Lapped Transforms*, Artech House, 1992.
- [40] MPEG/AUDIO SOFTWARE SIMULATION GROUP, *ISO/IEC 11172-3 Layer 1 and Layer 2 Software Package*, 1993.
- [41] I. NAGEL, *Ein Audiocodierer zur Vermeidung von Rekonstruktionsfehlern und Anwendungen auf robuste Kompressionsverfahren*. Diplomarbeit, Institut für Informatik V, Universität Bonn, 1999.
- [42] T. Q. NGUYEN, *Near Perfect-Reconstruction Pseudo-QMF Banks*, IEEE Trans. SP, 42 (1994), pp. 65–76.
- [43] ———, *Digital filter bank design quadratic-constrained formulation*, IEEE Trans. SP, 43 (1995), pp. 2103–2108.
- [44] P. NOLL, *MPEG Digital Audio Coding*, IEEE Signal Processing Magazine, (1997), pp. 59–81.
- [45] N. PACKHAM, *Transport kontextbasierter Informationen innerhalb digitaler Audiodaten und Anwendungen in digitalen Musikbibliotheken*. Diplomarbeit, Institut für Informatik V, Universität Bonn, 1999.
- [46] T. PAINTER AND A. SPANIAS, *A Review of Algorithms for Perceptual Coding of Digital Audio Signals*, 1997. <http://www.eas.asu.edu/~speech/ndtc/dsp97.ps>.
- [47] W. W. PETERSON AND E. WELDON, *Error-Correcting Codes*, MIT Press, 9th ed., 1988.
- [48] K. C. POHLMANN, *Principles of Digital Audio*, McGraw-Hill, 1st ed., 1995.
- [49] K. R. RAO AND P. YIP, *Discrete Cosine Transform - Algorithms, Advantages and Applications*, Academic Press, 1990.

- [50] A. RIBBROCK, *Ein Audiocodierer zur Vermeidung von Generationseffekten aufbauend auf dem MPEG-1 Standard (ISO/IEC 11172-3)*. Diplomarbeit, Institut für Informatik V, Universität Bonn, 1999.
- [51] A. SAID AND W. A. PEARLMAN, *A New Fast and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees*, IEEE Transactions on Circuits and Systems for Video Technology, 6 (1996).
- [52] E. D. SCHEIRER, *SAOL: The MPEG-4 Structured Audio Orchestra Language*, in International Computer Music Conference, Ann Arbor, MI., 1998.
- [53] ———, *The MPEG-4 Structured Audio Standard*, in Proc. ICASSP, 1998.
- [54] M. R. SCHROEDER, B. S. ATAL, AND J. L. HALL, *Optimizing digital speech coders by exploiting masking properties of the human ear*, The Journal of the Acoustical Society of America, 66 (1979), pp. 1647–1652.
- [55] J. M. SHAPIRO, *Embedded Image Coding Using Zerotrees of Wavelet Coefficients*, IEEE Trans. SP, 41 (1993), pp. 3445–3462.
- [56] D. SINHA AND J. D. JOHNSTON, *Audio Compression at Low Bit Rates Using a Signal Adaptive Switched Filter Bank*, in Proc. ICASSP, 1996, pp. 1053–1056.
- [57] B. TANG, A. SHEN, G. POTTIE, AND A. ALWAN, *A perceptually based embedded subband speech coder*, IEEE Trans. on Speech and Audio Processing, 5 (1997), pp. 131–140.
- [58] TECHNICAL CENTRE OF THE EUROPEAN BROADCASTING UNION, *Sound Quality Assessment Material Recordings for Subjective Tests*, 1988.
- [59] P. P. VAIDYANATHAN, *Multirate digital filters, filter banks, polyphase networks, and applications: A tutorial*, Proc. IEEE, 78 (1990), pp. 56–93.
- [60] ———, *Multirate Systems and Filter Banks*, Prentice Hall, 1993.
- [61] R. VELDHUIS AND A. KOHLRAUSCH, *Waveform Coding and Auditory Masking*, Elsevier, 1995, ch. 11, pp. 397–431.
- [62] D. WIESE, *Verfahren zum unbemerkten Übertragen und/oder Speichern von Zusatzinformationen innerhalb eines quellencodierten, datenreduzierten Audiosignals*, 1996. Offenlegungsschrift beim Deutschen Patentamt DE 4430864 A1.
- [63] J. ZHAO AND E. KOCH, *Embedding robust labels into images for copyright protection*.
- [64] U. ZÖLZER, *Digitale Audiosignalverarbeitung*, Teubner, 1996.
- [65] E. ZWICKER AND H. FASTL, *Psychoacoustics*, Springer, 1990.

