(19) Deutsches Patent- und Markenamt

(10) **DE 699 35 811 T3** 2012.08.16

# (12) Übersetzung der geänderten europäischen Patentschrift

(97) **EP 1 330 038 B2**
(21) Deutsches Aktenzeichen: **699 35 811.6**
(96) Europäisches Aktenzeichen: **03 00 9177.1**
(96) Europäischer Anmeldetag: **07.12.1999**
(97) Erstveröffentlichung durch das EPA: **23.07.2003**
(97) Veröffentlichungstag der Patenterteilung beim EPA: **11.04.2007**
(97) Veröffentlichungstag des geänderten Patents beim EPA: **13.06.2012**
(47) Veröffentlichungstag im Patentblatt: **16.08.2012**

(51) Int Cl.: ***H03M 7/48*** (2006.01)
***H03M 7/40*** (2006.01)
***H03M 7/46*** (2006.01)
***H03M 5/14*** (2006.01)

**Patentschrift wurde im Einspruchsverfahren geändert**

(30) Unionspriorität:
**211531** **14.12.1998** **US**

(84) Benannte Vertragsstaaten:
**AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LI, LU, MC, NL, PT, SE**

(73) Patentinhaber:
**Microsoft Corp., Redmond, Wash., US**

(74) Vertreter:
**Grünecker, Kinkeldey, Stockmair & Schwanhäusser, 80802, München, DE**

(72) Erfinder:
**Chen, Wei-ge, Issaquah,WA 98029, US; Lee, Ming-Chieh, Bellevue,WA 98006, US**

(54) Bezeichnung: **Frequenzbereichsaudiodekodierung mit Entropie-code Moduswechsel**

## Beschreibung

### Gebiet der Erfindung

**[0001]** Die Erfindung betrifft allgemein gesagt Frequenzdomänen-Audiocodieren und im Besonderen Entropiecodierverfahren, die bei Frequenzdomänen-Audiocodierern und -decodierem eingesetzt werden.

### Hintergrund

**[0002]** In einer typischen Audiocodierumgebung werden Daten, wenn nötig (beispielsweise aus einem analogen Format) in eine lange Sequenz von Symbolen, die als Eingabefolge für den Codierer dienen, formatiert. Die Eingangsdaten werden durch einen Codierer codiert, über einen Kommunikationskanal übermittelt (oder einfach gespeichert) und von einem Decodierer decodiert. Während des Codierens wird die Eingabefolge vorverarbeitet, gesampelt, konvertiert, komprimiert oder auf andere Weise in eine für die Übertragung oder für die Speicherung geeignete Form umgewandelt. Nach der Übertragung Oder der Speicherung hat der Codierer die Aufgabe, die ursprüngliche Eingabefolge zu rekonstruieren.

**[0003]** Audiocodiertechniken können in zwei Klassen eingeteilt werden, nämlich in das Zeitdomänenverfahren und das Frequenzdomänenverfahren. Zeitdomänenverfahren, wie beispielsweise ADPCM, LPC arbeiten unmittelbar in der Zeitdomäne, während das Frequenzdomänenverfahren die Audiosignale in Frequenzdomänen überführt, um dort die Kompression auszuführen. Frequenzdomänen-Kompressoren/Dekompressoren (codecs) können weiterhin unterteilt werden in Teilband- oder Transformcodierer, obwohl die Unterscheidung zwischen diesen beiden nicht immer klar ist. Das bedeutet, dass Teilbandcodierer typischerweise Bandpassfilter benutzen, um das Eingangssignal in eine kleine Zahl (beispielsweise 4) von Teilbändern zu teilen, wobei Transformcodierer typischerweise mehrere Teilbänder (und deshalb eine entsprechend große Zahl von Transformkoeffizienten) haben.

**[0004]** Die Verarbeitung eines Audiosignals in der Frequenzdomäne ist hergeleitet sowohl aus der klassischen Signalverarbeitungstheorie als auch aus einem psychoakustischen Modell des Menschen. Psychoakustik übernimmt die Vorteile der bekannten Eigenschaften des Hörers, um den Informationsinhalt zu verringern. Zum Beispiel verhält sich das Innenohr, im Besonderen die Basilarmembran, wie ein Spektralanalysator und überfuhrt die Audiosignale in spektrale Daten, bevor eine weitere neuronale Verarbeitung stattfindet. Frequenzdomänenaudiocodecs nutzen meistens den Vorteil der auditiven Maskierung, die im menschlichen Hörsystem vorhanden ist, während der Modifikation des Originalsignals, um Redundanzen in der Information zu eliminieren. Da das menschliche Ohr diese Veränderungen nicht hören kann, kann man eine effiziente Kompression ohne Verzerrung entwickeln.

**[0005]** Die Maskierungsanalyse wird gewöhnlich in Verbindung mit der Quantisierung durchgeführt, so dass Quantisierungsgeräusche bequem „maskiert" werden können. Bei modernen Audiocodierverfahren werden gewöhnlich die quantisierten Spektraldaten durch die Anwendung der Entropiecodierung weiter komprimiert, beispielsweise durch Huffman-Codierung. Die Kompression ist notwendig, weil die Kommunikationskanäle gewöhnlich eine begrenzte zulässige Kapazität oder Bandbreite haben. Es ist häufig notwendig, den Informationsinhalt der Eingangsdaten zu reduzieren, um sie zuverlässig, zumindest jedoch über den Kommunikationskanal zu übermitteln.

**[0006]** Außerordentlicher Aufwand wurde bisher aufgewendet, um verlustfreie und verlustbehaftete Kompressionsverfahren für die Reduktion des Umfangs der zu übertragenden oder zu speichernden Daten zu entwickeln. Ein weit verbreitetes verlustfreies Verfahren ist das Huffman-Codieren, welches eine Sonderform der Entropiecodierung ist. Die Entropiecodierung weist verschiedenen Eingabesequenzen Codewörtem zu und speichert alle Eingabesequenzen in einem Codebuch. Die Komplexität der Entropiecodierung hängt ab von der Zahl m möglicher Werte, die eine Eingabesequenz X haben kann. Für kleine m gibt es wenige mögliche Eingabekombinationen, deshalb kann das Codebuch für Nachrichten sehr klein sein (beispielsweise nur wenige Bits, um unzweideutig alle möglichen Eingangssequenzen wiederzugeben). Für digitale Anwendungen entspricht das Codealphabet typischerweise einer Folge von binaren Zahlen {0, 1}, die Codewortlänge wird in Bits gemessen.

**[0007]** Es ist bekannt, wenn Eingabefolgen aus Symbolen mit gleicher Ereigniswahrscheinlichkeit bestehen, dass eine optimale Codierung mit Codewörtern einheitlicher Lange vorzunehmen ist. Aber es ist nicht typisch, dass ein Eingabestrom eine einheitliche Wahrscheinlichkeit für das Empfangen einer bestimmten Nachricht hat. In der Praxis sind bestimmte Nachrichteninhalte wahrscheinlicher als andere und Entropiecodierer benutzen den Vorteil solcher Datenkorrelationen, um die mittlere Länge der Codewörter in den erwarteten Eingabefolgen zu minimieren. Üblicherweise werden Eingabesequenzen mit fester Länge Codes variabler Länge zugeordnet (oder es werden umgekehrt Eingabesequenzen mit variabler Länge Codes fester Länge zugeordnet).

**[0008]** EP-A-0 612 156 (Fraunhofer Gesellschaft für Forschung) vom 24. August 1994 beschreibt einen einzelnen Codierer, der aus verschiedenen Huffman-Code-Tabellen auswählt, die verschiedene Zahlen von Huffman-Codes enthalten (d. h. 8 Werte, 16 Werte oder 32 Werte), abhängig von der Maximalzahl der

mit der Tabelle zu codierenden Daten, wobei für verschiedene Spektralbereiche unterschiedliche Codetabellen verwendet werden können. Das Auswählen von Huffman-Code-Tabellen verschiedener Größe sorgt für eine gewisse Anpassbarkeit beim Decodieren, ist aber in vielen Szenarien des Audiodecodierens ineffizient. So können z. B. einige Formen der Run-Length-Codierung bzw. -Dekodierung bestimmte Arten von Wahrscheinlichkeitsverteilungen in Spektralbereichen sehr gut verarbeiten, wo die Huffman-Codierung (auch bei Tabellenauswahl) ineffizient ist (z. B. wo 90% der Werte in einem Spektralbereich Nullen sind). Die Tabellenauswahl für einen gegebenen Codierer/Decodierer kann eine gewisse Anpassbarkeit bewirken, jedoch nur innerhalb den Grenzen eines gegebenen Codierers/Decodierers selbst.

**[0009]** US-A-5 467 087 (Chu Ke-Chiang) vom 14. November 1995 (1995-11-14) bezieht sich auf gewöhnliche Datenverdichtung, nicht auf Audio-Dekomprimierung, und beschreibt die Verwendung verschiedener Komprimierungsverfahren, aber beschreibt nicht die Verwendung verschiedener Decoder zum Entschlüsseln verschiedener Unterbereiche eines gegebenen Datenbereichs. Vielmehr wird das Komprimieren derselben Daten beschrieben, indem mehr als ein Komprimierungsverfahren eingesetzt wird und die Komprimierungsverfahren abgestuft werden.

**[0010]** Tewfik A. H. et al: „Enhanced wavelet based audio coder", Signals, Systems and Computers, 1993, Conference Record of the TWenty-Seventh Asilomar Conference an Pacific Grove, CA, USA 1.–3. Nov. 1993, Los Alamitos CA, USA, IEEE Comput. Soc. 1. Nov. 1993 (1993-11-01), Seiten 896–900, XP010096271 ISBN: 0-8186-4120-7, bespricht die Echtzeit-Anwendung einer vereinfachten Version einer Wellen basierten Audiocodierung mit etwas höheren Bit-Raten (64 bis 78 Kbits/sec) und benutzt beim Verdecken von Audio-Segmenten Zeitdaten mit einem plötzlichen Wechsel der Amplituden darin und mit effizienter Quantisierung der Abweichungen und Run-Length-Codierung für Nullen.

## Übersicht

**[0011]** Die Erfindung betrifft ein Verfahren zur Auswahl einer Form der Entropiecodierung für die Frequenzdomänen-Audiocodierung. Im Besonderen wird ein bestimmter, Audioeingabefolgen darstellender Eingabestrom in Frequenzbereiche unterteilt, entsprechend einigen statistischen Kriterien, die aus einer statistischen Analyse von typischen oder aktuell zu codierenden Eingabefolgen abgeleitet werden. Jeder Bereich wird einem Entropiecodierer zugeordnet, der auf den Typ der in diesem Bereich zu codieren Daten optimiert ist. Während des Codierens und des Decodierens wendet ein Modenselektor auf verschiedene Frequenzbereiche das jeweils richtige Entropieverfahren an. Im Voraus können Unterteilungsgrenzen festgelegt werden, die dem Decodierer implizit zur Kenntnis bringen, welches Decodierverfahren für die zu codierenden Daten anzuwenden ist. Oder es kann ein sogenanntes forward adaptive Verfahren benutzt werden, bei dem die Grenzen im Ausgabestrom durch die Indizierung des Wechsels des Codiermodus für die nachfolgenden Daten gekennzeichnet (flagged) sind.

**[0012]** Für natürliche Geräusche, etwa solche wie Sprache oder Musik, ist der Informationsinhalt auf niedrige Frequenzbereiche konzentriert. Das bedeutet statistisch, dass die niedrigen Frequenzen mehr Energiewerte ungleich Null (nach der Quantisierung) haben, während die höheren Frequenzbereiche mehr Werte mit Nullen haben, worin sich das Fehlen höherer Frequenzen widerspiegelt. Die statistische Analyse kann benutzt werden, um eine oder mehrere Unterteilungsgrenzen zu definieren, die niedrige von höheren Frequenzbereichen trennt. Zum Beispiel kann eine einzelne Unterteilung so definiert werden, dass das untere Viertel der Frequenzkomponenten unterhalb der Teilung liegt. Alternativ dazu kann man die Teilung so setzen, dass ungefähr eine Hälfte der kritischen Bänder in jedem definierten Frequenzband enthalten ist. (Kritische Bänder sind Frequenzbänder mit ungleichförmiger Breite, die der Empfindlichkeit des menschlichen auditiven Systems für bestimmte Frequenzen entsprechen). Das Ergebnis solcher Unterteilung ist die Definition zweier Frequenzbereiche, von denen der eine vorherrschend Frequenzkoeffizienten ungleich Null, während der andere Frequenzkoeffizienten vorherrschend aus Nullen enthalt. Mit der Vorkenntnis, dass die Bereiche vorherrschend Nullen oder Nicht-Nullen enthalten, kann mit Codierern codiert werden, die auf Null- oder Nicht-Null-Werte optimiert sind.

**[0013]** In einer Ausführungsform kann ein vorherrschend Null-Werte enthaltender Bereich mit einem Vielfach-Lauflängencodierer (run-length encoder, RLE) codiert werden, beispielsweise ein Codierer, der Sequenzen von Null-Werten mit ein oder mehreren Nicht-Null-Symbolen statistisch korreliert und Codewörtern variabler Lange zu willkürlich langen Eingabesequenzen solcher Null- und Nicht-Null-Werte zuweist. Ähnlich wird der Bereich, der hauptsächlich Nicht-Null-Werte enthält, mit einem Variable-zu-Variable-Entropie-Codierer codiert, wobei ein Codewort mit variabler Lange einer willkürlich langen Eingabesequenz aus Quantisierungssymbolen zugeordnet wird. Ein im Ganzen effizienteres Verfahren wird erreicht, wenn die zugrunde gelegten Codierverfahren entsprechend den Eigenschaften der Eingabedaten gewählt werden. In der Praxis wird die Zahl der Unterteilungen und der Frequenzbereiche in Abhängigkeit von den Datentypen, die zu codieren oder zu decodieren sind, variieren.

## Kurze Beschreibung der Zeichnungen

**[0014]** Die Fig. 1 ist ein Blockdiagramm eines Computersystems, das zur Implementierung von Frequenzdomänenaudiocodierung und -decodierung benutzt werden kann, welches Entropiecode-Modenwechsel verwendet.

**[0015]** Die Fig. 2 ist ein Flussdiagramm, welches Codieren und Decodieren von Audiodaten durch einen Frequenzdomänen-Audiocodierer zeigt.

**[0016]** Die Fig. 3 stellt einen Frequenzbereich dar, der entsprechend der Akustik unterteilt ist.

**[0017]** Die Fig. 4 stellt ein Transmissionsmodell dar, welches Entropiecode-Modenwechsel verwendet.

**[0018]** Die Fig. 5 ist ein Flussdiagramm, das das Erstellen eines Codebuchs mit variablen Längeneintragen für Symbolgruppierungen mit variabler Lange zeigt.

**[0019]** Die Fig. 6–Fig. 12 zeigen das Erstellen eines Codebuchs gemäß Fig. 5 für ein Alphabet {A, B, C}.

**[0020]** Die Fig. 13 zeigt ein spektrales Schwellengitter, welches beim Codieren von Audiosequenzen mit sich wiederholenden Spektralkoeffizienten eingesetzt wird.

**[0021]** Die Fig. 14 zeigt die Implementierung des Entropiecodierers gemäß Fig. 2.

## Detaillierte Beschreibung

### Beispielhaftes Einsatzumfeld

**[0022]** Mit Fig. 1 und der folgenden Diskussion ist beabsichtigt, eine kurze allgemeine Beschreibung einer geeigneten Rechnerumgebung zu liefern, in die die Erfindung implementiert werden kann. Während die Erfindung allgemein im Kontext von computerausführbaren Befehlen eines Computerprogramms beschrieben wird, welches auf einem Personal-Computer läuft, werden Fachleute erkennen, dass die Erfindung auch in Kombination mit anderen Programm-Modulen implementiert werden kann. Allgemein gesagt, schließen Programm-Module Routinen, Programme, Komponenten, Datenstrukturen usw. ein, die spezielle Aufgaben ausführen oder die spezielle abstrakte Datentypen implementieren. Weiterhin werden Fachleute erkennen, dass die Erfindung mit anderen Computer-System-Konfigurationen ausgeführt werden kann, einschließlich Handheld-Geraten, Multiprozessor-Systemen, mit Mikroprozessoren ausgerüstete oder programmierbare Endkundenelektronik, Minicomputern, Mainframe-Computern und ähnlichen. Die vorgestellte Ausführungsform der Erfindung ist auch in verteilter Rechner-Umgebung ausführbar, in der Programme in Fernverarbeitungseinrichtungen ausgeführt werden, die durch ein Kommunikationsnetzwerk miteinander verknüpft sind. Es können aber auch Ausführungsformen der Erfindung auf einem Einzel-Rechner ausgeführt werden. In einer verteilten Rechner-Umgebung können Programm-Module in Datenspeichereinrichtungen abgelegt sein, die lokal als auch remote vorhanden sind.

**[0023]** Bezugnehmend auf Fig. 1 schließt ein beispielhaftes System zur Implementierung der Erfindung einen Computer **20** ein, umfassend eine Prozessoreinheit **21**, einen Systemspeicher **22** und einen Systembus **23**, der untereinander verschiedene Systemkomponenten einschließlich den Systemspeicher mit der Prozessoreinheit **21** verbindet. Die Prozessoreinheit kann eine von verschiedenen kommerziell verfügbaren Prozessoren sein, einschließlich Intel x86, Pentium oder kompatible Mikroprozessoren von Intel oder anderen, den Alpha-Prozessor von Digital, oder den PowerPC von IBM oder Motorola. Ebenfalls können Doppelmikroprozessoren oder andere Multiprozessor-Architekturen als Prozessoreinheit **21** verwendet werden.

**[0024]** Der Systembus kann einer der verschiedensten Typen von Bus-Strukturen sein, einschließlich solcher wie Speicher-Bus oder Speicher-Controller, ein Peripherbus, oder ein lokaler Bus, der von unterschiedlichsten konventionellen Busarchitekturen, wie etwa PCI, AGP, VESA, Microchannel, ISA oder ESA Gebrauch macht, um nur einige zu nennen. Der Systemspeicher schließt einen Nur-Lese-Speicher (ROM) **24** und einen Direktzugriffsspeicher (RAM) **25** ein. Eine BIOS-System-Routine ist im ROM **24** gespeichert, welche die grundlegenden Routinen enthalt, die den Informationsfluss zwischen den Elementen des Computers **20** unterstützen, etwa während der Anlaufphase.

**[0025]** Der Computer **20** umfasst ferner ein Festplattenlaufwerk **27**, ein Magnetplattenlaufwerk **28**, um von einer Wechselplatte **29**, einem optischen Plattenlaufwerk, oder einer CD-ROM-Scheibe **30** oder von anderen optischen Medien, beispielsweise zu lesen oder auf ihnen zu schreiben. Das Festplattenlaufwerk **27**, das Magnetplattenlaufwerk **28** und das optische Plattenlaufwerk **30** sind über ein Festplattenlaufwerk-Interface **32** mit dem Systembus **23** verbunden, beziehungsweise über ein Magnetplattenlaufwerk-Interface **33** und ein Interface des optischen Plattenlaufwerks **34**. Die Laufwerke und ihre zugehörigen computerlesbaren Medien ermöglichen dem Computer **20** die nichtflüchtige Speicherung der Daten, der Datenstrukturen, der computerausführbaren Befehle usw.. Obwohl sich die Beschreibung der computerlesbaren Medien auf ein Festplattenlaufwerk, eine magnetische Wechselplatte, oder eine CD bezieht, wird der Fachmann erkennen, dass andere Medien-

arten, die computerlesbar sind, wie etwa Magnetkassetten, Flashmemory-Karten, digitale Videoplatten, Bernoulli-Kassetten oder ähnliche, auch in dieser exemplarischen Computer-Umgebung benutzt werden können.

**[0026]** Eine Anzahl von Programm-Modulen können auf den Laufwerken und im RAM **25** gespeichert werden, einschließlich ein Betriebssystem **35**, ein oder mehrere Anwendungsprogramme **36** (beispielsweise ein Internet-Browserprogramm) oder andere Programm-Module **37** oder Programmdaten **38**.

**[0027]** Ein Anwender kann Befehle und Informationen über eine Tastatur **40** oder eine Zeigeeinrichtung, wie etwa eine Maus, in den Computer **20** eingeben. Andere (nicht gezeigte) Eingabeeinrichtungen können ein Mikrophon, ein Joystick, ein Gamepad, eine Satellitenschüssel, ein Scanner oder ähnliches sein. Diese und andere Eingabeeinrichtungen werden oft über ein serielles Interface **46** mit der Prozessoreinheit **21** verbunden, die an den Systembus gekoppelt ist; es können aber auch andere Schnittstellen, wie etwa eine parallele Schnittstelle, eine Spiele-Schnittstelle oder ein universeller serieller Bus (USB) vorhanden sein. Ein Monitor **47** Oder eine andere Art eines Displays ist ebenfalls mit dem Systembus **23** über eine Schnittstelle verbunden, wie etwa über einen Videoadapter **48**. Zusätzlich zum Monitor umfassen Personalcomputer typischerweise noch andere (nicht dargestellte) periphere Ausgabeeinrichtungen, wie etwa Lautsprecher oder Drucker.

**[0028]** Es wird erwartet, dass der Computer **20** in einer Netzwerk-Umgebung arbeitet, wobei er logische Verbindungen mit einem oder mehreren Fernrechnern, etwa dem Femrechner **49** benutzt. Der Fernrechner **49** kann ein Netz-Server, ein Router, eine Peereinrichtung oder ein anderer üblicher Netzwerkrechner sein und typischerweise viele oder alle Elemente umfassen, die in Bezug auf den Computer **20** beschrieben sind, obwohl nur eine Datenspeichereinrichtung **50** in der Fig. 1 dargestellt ist. Der Computer **20** kann den Fernrechner **49** über Internetverbindung kontaktieren, die durch ein Gateway **55** (beispielsweise einen Router, eine Festleitung oder andere Netzverbindungen), eine Modemverbindung **54**, ein lokales Intranet (LAN) **51** oder ein wide area network (WAN) **52** hergestellt ist. Es soll betont werden, dass die dargestellten Netzwerkverbindungen exemplarisch sind und dass andere Einrichtungen zur Herstellung von Kommunikationsverbindungen zwischen den Computern benutzt werden können.

**[0029]** Die vorliegende Erfindung soll für den mit Programmiertätigkeit vertrauten Fachmann anhand von Aktionen und symbolischen Repräsentationen der Operationen des Computers **20** beschrieben werden, soweit nicht anderweitig angegeben. Solche Aktionen und Operationen werden manchmal mit computerausführbar bezeichnet. Es wird hervorgehoben, dass Aktionen und symbolische Repräsentationen der Operationen folgendes einschließen sollen:: die Manipulation von elektrischen Signalen, die Datenbits repräsentieren, durch die Prozessoreinheit **21**, welche im Ergebnis eine Transformation oder Reduktion der elektrischen Signalrepräsentation liefert; die Behandlung von Datenbits auf Speicherorten im Speichersystem (einschließlich Systemspeicher **22**, Festplattenlaufwerk **27**, Floppyplatte **29**, oder CD-ROM **31**), wobei Computersystemoperationen rekonfiguriert oder anderweitig geändert werden, oder andersartige Verarbeitungen von Signalen. Die Speicherorte, an denen Datenbits bearbeitet werden, sind physikalische Orte, die bestimmte elektrische, magnetische oder optische Eigenschaften haben, die den Datenbits entsprechen.

**[0030]** Die Fig. 2 zeigt ein Transmissionsmodell für die Weiterleitung von Audiodaten über einen Kanal **210**. Die Transmissionsquelle kann eine Rundfunk-Livesendung, gespeicherte Daten oder Informationen sein, die über drahtgebundene/drahtlose Kommunikationsverbindungen (beispielsweise ein LAN oder das Internet) übermittelt werden. Es wird vorausgesetzt, dass der Kanal **210** eine begrenzte Bandbreite hat, und dass deshalb die Kompression der Quellendaten **200** erwünscht ist, bevor die Daten zuverlässig über den Kanal übermittelbar sind. Es soll angemerkt werden, dass die Erfindung, obwohl sich die Diskussion auf die Transmission von Audiodaten konzentriert, auf die Übertragung anderer Daten angewendet werden kann, etwa solche wie Audio/Video-Informationen mit eingebetteten Audiodaten (beispielsweise im MPEG-Datenstrom gemultiplext) oder andere Datenquellen mit komprimierbaren Datenstrukturen (beispielsweise kohärente Daten).

**[0031]** Wie dargestellt ist die Datenquelle **200** der Eingang für einen Transformationscodierer **202** von Zeit- auf Frequenzdomänen, wie etwa eine Filterbank oder ein Transformator vom Typ „discrete-cosine". Der Transformationscodierer **202** ist für das Konvertieren einer kontinuierlichen oder vereinzelten Zeitdomänen-Eingabefolge, wie etwa eine Audiodatenquelle, in Mehrfach-Frequenzbänder mit vorgegebenen (obwohl manchmal auch unterschiedlichen) Bandbreiten geeignet. Diese Bänder können daraufhin unter Verwendung eines menschlichen auditiven Wahrnehmungsmodells **204** analysiert werden (zum Beispiel mit einem psychoakustischen Modell), um die Signalkomponenten zu bestimmen, die ohne Hörverlust sicher reduziert werden können. Zum Beispiel ist gut bekannt, dass bestimmte Frequenzen unhörbar sind, wenn bestimmte andere Gerausche oder Frequenzen im Eingangssignal vorhanden sind (simultane Maskierung). Konsequenterweise können solche unhörbaren Signale sicher aus dem Eingangssignal entfernt werden. Der Gebrauch eines menschlichen auditiven Modells ist gut bekannt, beispielweise die

MPEG 1, 2, oder 4 Normen. (Es wird angemerkt, dass solche Modelle auch mit einer Quantisierungsoperation **206** verknüpft werden können).

**[0032]** Nach dem Ausfuhren der Transformation **202** von Zeit- auf Frequenzdomänen werden Frequenzkoeffizienten aus jedem Bereich quantisiert **206**, um jeden Koeffizienten (Amplitudenqgrößen) auf einen Wert zu konvertieren, der aus einer endlichen Menge von möglichen Werten genommen ist, in der jeder Wert eine Größe in Bits hat, die den zugewiesenen Frequenzbereich repräsentiert. Der Quantisierer kann ein konventioneller gleichförmiger oder ungleichförmiger Quantisierer sein, wie etwa ein midriser oder midtreader Quantisierer mit (oder ohne) Speicher. Das allgemeine Ziel der Quantisierung ist es, eine optimale Bitzuordnung, zur Repräsentation der Eingabesignaldaten, zu identifizieren, beispielsweise um die Verwendung von vorhandenen Codier-Bits zuzuteilen, um das Codieren von (akustischen) signifikanten Teilen der Quellendaten sicherzustellen. Verschiedene Quantisierungsmethoden können verwendet werden, wie etwa die der Quantisierung nach step size prediction, um eine gewünschte Bit-Rate zu treffen (angenommen seien konstante Bit-Raten). Nachdem die Quelle **200** quantisiert wurde, werden die resultierenden Daten entropiecodiert **208** (siehe die Diskussion zu Fig. 6 bis Fig. 13).

**[0033]** Die entropiecodierte Ausgabefolge wird über den Kommunikationskanal **210** übermittelt (oder für eine spätere Übertragung gespeichert). Die empfangende Endstation **216** implementiert einen Rück-Codierprozess, beispielsweise eine Reihe von Schritten zum Rückgängigmachen der Codierung der Quellendaten **200**. Das heißt, dass die über den Kommunikationskanal **210** empfangenen codierten Daten als Eingabefolge für einen Entropiedecodierer **212** genommen werden, der rückwärts das Codebuch-Nachschlagen ausführt, um die codierte Ausgangsfolge in eine Approximation der ursprünglich quantisierten Ausgabefolge für die Eingabesymbolfolge **200** zu konvertieren. Diese angenäherten Daten werden dann von einem Dequantisierer **214** und einen Zeit-/Frequenzdomänencodierer **218** abgearbeitet, um die ursprüngliche Codieroperation umzukehren. Im Ergebnis entstehen rekonstruierte Daten **220**, die den ursprünglichen Quellendaten ähnlich sind. Es soll bemerkt werden, dass die rekonstruierten Daten **220** nur den ursprünglichen Quellendaten angenähert sind, denn die angewendeten Schritte **204** bis **208** sind verlustbehaftete Prozesse.

**[0034]** Eine mögliche Implementierung für das Übertragungsmodell ist ein Anwenderprogramm (client application), welches das Ausführen, das Anzeigen oder das Echtzeitabspielen der Daten veranlasst, die über eine Netzwerkverbindung von einem Server/einer Server-Anwendung angefordert werden. Zum Beispiel kann der Clientrechner ein Streaming-Übertragungssystem verwenden, welches adaptive Bandbreitenreservierung bereitstellt. (Eines dieser Streamingformate ist das Advanced Streaming Format von Microsoft). Eine Streamingumgebung steht im Gegensatz zu traditionellen Netzwerkprogrammen, weil sie die optimierte Datenbereitstellung für die Bedürfnisse des Retrievals erlaubt, etwa wie die Beschränkung der Zeilengeschwindigkeit. Eine herausgehobene Eigenschaft der Streamingdaten ist, dass die Daten in Echtzeit fortschreitend betrachtet werden können, so wie sie vom Client empfangen werden. Es wird angemerkt, dass es beabsichtigt ist, dass die verarbeiteten Daten für späteres Retrieval durch einen Client gespeichert werden können und dass dieses Retrieval in einem Nicht-streaming-Format ausgeführt werden kann (beispielsweise über eine kleine Playback-Anwendung).

**[0035]** Das Streaming-Format definiert die Struktur der synchronisierten Objektdatenströme und erlaubt es, jedes Objekt, beispielsweise Audio- oder Videodatenobjekte, Scripts, ActiveX controls oder HTML-Dokumente, in den Datenstrom einzufügen. Eine Anwenderprogrammschnittstelle (eine solche wäre das API von Microsoft Audio Compression Manager) wird bereitgestellt, um die Anwenderunterstützung des Streaming-Formats zu erleichtern. Die Übertragung von Streaming-Formatdaten über den Kommunikationskanal **210** verlangt, dass die Quelleninformation in eine für das Netzwerk geeignete Form konvertiert werden muss. Im Unterschied zu den traditionellen Paketen, die Routinginformationen und Daten enthalten, enthalten Streamingpakete einen priorisierten Mix von Daten verschiedener Objekte im Strom, so dass die Bandbreite zuerst für die Objekte mit höherer Priorität eingerichtet werden kann. Am Empfangsende **216** werden die Objekte im priorisierten Datenstrom für die Verwendung durch den Empfänger rekonstruiert.

**[0036]** Weil Daten wahrscheinlich so verwendet werden, wie sie empfangen werden, ist der Streaminginhalt empfindlich gegen Übertragungsverzögerungen. Wenn Daten nicht zuverlässig ankommen, oder wenn die Übertragungsgeschwindigkeit unter ein akzeptables Minimum fällt, werden die Daten unbrauchbar (beispielsweise Playback oder die Videosequenz schlägt fehl). Konsequenterweise verlangen bandbreitenintensive Daten (etwa solche wie die des Audiovorlaufs) eine signifikante Kompression, um sicherzustellen, dass sie der Bandbreitenanforderung des Kommunikationskanals **210** entsprechen. Da der Grad einer verlustbehafteten Kompression notwendigerweise die Qualität des reproduzierten Signals beeinflusst, sollte der Server auswählbare Codierungen für verschiedene Client-Netzwerk-Verbindungsgeschwindigkeiten zur Verfügung stellen (oder er sollte ein adaptives Feedbacksystem verwenden, um den Durchsatz auf Echtzeit einzustellen).

**[0037]** Eine besonders effektive Methode des Codierens der Frequenzkoeffizienten der Quellendaten **200** ist das Entropiecodieren, um die verlässliche Übertragung über den Kornmunikationskanal **210** sicherzustellen. Wie noch diskutiert werden soll, kann man auf die Datenkohärenz durch die Anwendung verschiedener Codiermethoden abheben, die auf verschiedene Teile der Eingabedaten optimiert sind. Das Entropiecodieren ist dann effektiv, wenn die Symbole eine ungleichförmige Wahrscheinlichkeitsverteilung haben. Entropiecodiermethoden, die viele Eingangssymbole gruppieren, wie etwa Variable-zu-Variable- oder die weiter unten diskutierten RLE-Codierer, sind gut geeignet, um sich auf die Datenkohärenz zu konzentrieren. Der Gebrauch verschiedener Codiermethoden für verschiedene Frequenzbereiche erlaubt ein optimales Codieren, wenn die Codierer auf Wahrscheinlichkeitsverteilungen für jeden dieser Bereiche zugeschnitten sind.

**[0038]** Die Fig. 3 illustriert ein Zeitdomänensignal, welches in die Frequenzdomäne übersetzt ist. Auf der x-Achse 1st ein Frequenzbereich **300** von Null **302** bis zu einer Maximalfrequenz **304** aufgetragen. Im Bereich **300** wurde eine Unterteilung **306** definiert, wo die Unterteilung durch die statistische Analyse eines erwarteten Eingabestroms bestimmt ist (beispielsweise durch während des Trainings des Entropiecodebuchs erhaltene statistische Information oder durch adaptive Analyse der aktuellen Eingabefolge). Dieses statistische Modell wird für das Codieren der aktuellen Eingabefolge **308** angewendet.

**[0039]** Ein Ansatz zur Vornahme einer Teilung ist es, wie oben diskutiert, einen bestimmten Prozentsatz von Frequenzen oder kritischen Bändern unterhalb der Grenze anzuordnen.

**[0040]** Ein alternatives Verfahren ist es, statistische Grunddaten zu sammeln, etwa solche wie die Wahrscheinlichkeit von Nullen oder Nicht-Nullen entsprechend den Wahrscheinlichkeitsverteilungen für jede Frequenz. Die Untersuchung der Statistik dieser Frequenzen zeigt einen graduellen Wechsel über die Frequenzen und es kann eine Teilungsgrenze ausgewählt werden, so dass die Verteilungen zwischen den einzelnen Teilungen ähnlich sind. Es sei bemerkt, dass Frequenzunterteilungen gegen die Samplingrate der Eingabefolge und der erwarteten Bit-Rate empfindlich sind. Die Samplingrate und die Bit-Rate bestimmen die stochastische Eigenschaft der quantisierten Frequenzkoeffizienten. Diese Eigenschaft ist grundlegend verantwortlich für die Bestimmung der Lage der Unterteilungen.

**[0041]** Ein anderes optimales Verfahren ist die adaptive Lokalisierung einer Unterteilung durch das Ausführen einer ausgiebigen Suche zur Bestimmung einer optimalen Grenzlage. Das bedeutet, dass eine optimale Lösung darin besteht, zu versuchen, jede Frequenz (oder eine Untermenge davon) als Teilungsort zu nehmen, eine Entropiecodierung vorzunehmen und zu prüfen, welche potentielle Grenzposition ein Minimum an Bits für die Codierung liefert. Obwohl vom Rechenumfang aufwendiger, ist bei einer vollständigen Suche die Kompression (oder einer fast vollständigen, wenn Frequenzuntermengen benutzt werden) von Vorteil, auch wenn die Rechnerkosten von Bedeutung sein sollten, denn diese kann die Kosten aufwiegen, wenn mehrfache Unterteilungen benutzt werden.

**[0042]** Durch die Aufteilung des Frequenzspektrums **300** in separate Frequenzunterbereiche **310**, **312** kann ein Codierer verschiedene Codierschemata anwenden, die auf das Codieren verschiedener Frequenzbereiche optimiert sind. Dies unterscheidet sich von früheren Verfahren, wie etwa von Entropiecodierschemata, die verschiedene Entropiecodiertabellen entsprechend den Eigenschaften der zu codieren Daten ersetzen. Diese früheren Verfahren sind in ihrer Flexibilität wegen eines einzigen Entropiecodieralgorithmusses dadurch begrenzt, dass eine einzige Codiertabelle nicht auf unterschiedliche Arten von Eingabedaten eingehen kann und dass ein Overhead vorhanden ist, durch die Identifizierung verschiedener zu benutzender Tabellen. Ein Verfahren, welches für einen einzigen Typ von Daten optimiert ist, kann nicht effizient auf einen anderen Typ von Daten angewendet werden.

**[0043]** In der beschriebenen Ausführungsform ist das ausgewählte Teilungskriterium für den Bereich **300** die Wahrscheinlichkeit C(F) (Y-Achse), dass ein bestimmtes spektrales Ereignis eine Folge von Koeffizienten bei Oder nahe bei einer bestimmten Intensität (beispielsweise Null) ist. Unter Zuhilfenahme der Codebücherstellung kann für eine beispielhafte Eingabefolge die Wahrscheinlichkeit, Nullwerte zu erhalten, vorausberechnet werden. Wie dargestellt hat das Eingabesignal **308** bei der angezeigten Unterteilung **306** eine hohe Wahrscheinlichkeit, Null zu sein. (Die Position des Teilungstrenners **306** war so gewählt worden, dass 80% Oder 90% der Eingabefolge hinter dem Unterteiler Null Oder fast Null sein wurden.)

**[0044]** Es wird angenommen, dass am Minimum ein Eingabesignal **308** in zwei Bereiche unterteilt wird, wobei jeder Bereich eine Datenstruktur aufweist, die für die Kompression durch verschiedene Codierverfahren am besten geeignet ist. Bei der vorgestellten Ausführungsform hat ein Bereich ursprünglich Null-Werte, während der andere ursprünglich Werte ungleich Null hat. Daher werden zwei Codierer benutzt, die jeweils auf die Datentypen in den entsprechenden Bereichen optimiert sind. Während die vorgestellte Anwendung die Frequenzkoeffizienten in zwei Bereiche teilt, können auch mehr als zwei Bereiche definiert werden, von denen jeder einen eigenen optimierten Codierer hat oder es können verschiede-

ne Bereiche ähnliche Charakteristika haben und dadurch denselben Codierer beschäftigen.

**[0045]** Für das Codieren des fast nur aus Nicht-Nullen bestehenden Bereichs **310** kann ein Entropiecodierer verwendet werden, wie er in den Fig. 6 bis Fig. 13 beschrieben wird. Wie weiter unten diskutiert wird, ist das Codierverfahren der Fig. 6 bis Fig. 13 zum Codieren von Audioeingabedaten mit Werten ungleich Null besonders gut geeignet. Für den Bereich **312**, der fast nur aus Nullen besteht, wird ein Codierer verwendet, der auf solche Daten optimiert ist. In der vorgestellten Ausführungsform, weil er für das Codieren von Daten mit vorherrschenden Werten optimiert ist (beispielsweise Nullen) wird ein Lauflängencodierer verwendet. Fig. 13 zeigt einen RLE-basierten Entropiecodierer, der den Bereich **312**, der hauptsachlich Nullen enthalt, effektiv codieren kann.

**[0046]** Die Fig. 4 zeigt ein Transmissionsmodell für die Übertragung von Audiodaten über den Kanal (siehe Fig. 2), in dem vielfache Entropiecodier-/Decodierverfahren angewendet werden, um die Eingangsdaten **200** zu manipulieren. Es ist bekannt, dass die Audioquelldaten **200** Werte nur innerhalb weniger Frequenzbereiche haben. Wie oben zu Fig. 2 diskutiert, können Quelldaten **200** in Frequenzdomänen konvertiert **202**, aufgrund eines psychoakustischen Modells **204** reduziert und quantisiert **206** werden. Da die Quantisierung eine bedeutende Zahl von Fast-Null-Ausgabewerten liefern kann, kann ein Entropiecodierer **208** auf die Codierung dieser quantisierten Ausgangsfolge optimiert werden.

**[0047]** Nach der Quantisierung tendieren die Spektralkoeffizienten der quantisierten Daten auf die Wiedergabe des Informationsinhalts von typischen Audiodaten. Die Analyse der Quantisierungskoeffizienten zeigt, dass diese bei niedrigen Frequenzbereichen nahezu fast nur Nicht-Null- und bei höheren Frequenzen fast nur Null-Koeffizienten sind. Deshalb kann für Frequenzunterteilungen an bestimmten Frequenzorten ein Modenselektor **400** festlegen, welcher Codierer zu verwenden ist, entsprechend dem zu codierenden Frequenzbereich.

**[0048]** Die Bestimmung der Platzierung der Unterteilung kann auf eine statistische Analyse basiert werden, die denjenigen unter bekannten Codierern identifiziert, der für verschiedene Unterbereiche die jeweils bessere Codiereffizienz liefert. In einer Konfiguration wird die Analyse vor dem Codieren oder Decodieren bezüglich einer beispielhaften Eingabefolge durchgeführt. Dies erlaubt die Vorherbestimmung von Teilungsorten und entsprechenden Codierern für jeden Unterbereich, so dass kein Overhead für den Flag-Wechsel in verwendbaren Codierern eingebracht werden muss.

**[0049]** Alternativ dazu kann eine statistische Analyse der vorliegenden Daten (in Echtzeit oder offline) durchgeführt werden. In dieser Konfiguration muss, obwohl der Codierer/Decodierer im Voraus bekannt ist, ein Flag in den codierten Datenstrom eingebettet werden, um die Wechsel der verwendbaren Codierer anzuzeigen. Wie oben angesprochen können verschiedene mögliche Teilungsorte solange getestet werden, bis ein gewisser Grad von Codiereffizienz für jeden Unterbereich erreicht ist. Der Empfang eines Flags durch einen Codierer zeigt das Ende des Unterbereichs an und der Wert des Flags gibt an, welcher Codierer für die nachfolgenden Daten zu verwenden ist.

**[0050]** Obwohl das Einfugen von Markern einen gewissen Overhead in den Codier-/Decodierprozess einbringt, stellen solche Marker eine Verbesserung gegenüber den Codiermethoden nach dem Stand der Technik dar. Zum Beispiel seien die vorgestellten Ausführungsformen mit traditionellem Entropiecodieren (siehe beispielsweise MPEG 1, 2, Oder 4) von Audiodaten verglichen. Ein traditionelles System verwendet einen einzigen Entropiecodierer für alle Daten, wobei verschiedene Codebücher mit jedem der vielen kritischen Bänder der Frequenzbereiche der Eingangsdaten verknüpft werden (gewöhnlich 24 oder mehr Bänder, abhängig von der Samplingrate). An jedem kritischen Bandübergang, es seien 24 Bänder angenommen, ist ein 2-Bit-Flag (oder ein längeres) erforderlich, um anzuzeigen, welches von 24 Codebüchern verwendet werden soll, um die Daten des Bandes zu codieren. (5 Bits sind notwendig um 24 Zustände nachzuvollziehen (track). Das Flag selbst kann in weniger Bits wirksam codiert werden). Dies steht im scharfen Gegensatz zu den vorgestellten Ausführungsformen, welche entweder überhaupt keinen Flag benötigen oder bei dem Gebrauch von Flags gegenüber früheren Verfahren eine höhere Effizienz besitzen, es sei denn, die Zahl der Unterbereiche wird mit der Zahl der kritischen Bänder vergleichbar und die Zahl der Codierverfahren erreicht die Zahl der Tabellen. Das bedeutet, dass bei jeder Codierung, die kritische Bänder verwendet, 24 Unterbereiche vorhanden sind, die ein 2-5-Bit-Flag erfordern, um anzuzeigen, welche Codiertabelle zu verwenden ist. Im Gegensatz dazu können die vorgestellten Ausführungsformen nur 2 oder drei Unterbereiche haben und haben daher viel weniger Overhead.

**[0051]** Wie gezeigt, gibt es N vorausbestimmte Codierer **402**–**406**, wobei jeder darauf optimiert ist, einen Frequenzbereich mit Daten mit vorherrschender Charakteristik zu codieren. Das bedeutet nicht, dass es notwendigerweise N besondere Eingabebereiche gibt, weil verschiedene Frequenzbereiche ähnliche statistische Charakteristika ihrer Daten haben könnten, und dass deshalb nur derselbe Codierer verwendet werden konnte. In dem vorgestellten Beispiel gibt es nur zwei Bereiche (eine Unterteilung), die je-

weils einem niedrigen (hauptsächlich Nicht-Null-Koeffizienten) und einem hohen (hauptsachlich Null-Koeffizienten) Frequenzbereich entsprechen. Deshalb werden die meisten Null-Werte hinter der Unterteilung mit einem Codierer vom RLE-Typ (siehe beispielsweise Fig. 13) und die Daten vor der Unterteilung mit einem Entropiecodierer vom Typ Variable-zu-Variable-Entropie codiert.

**[0052]** Im allgemeinen Fall jedoch, wenn einmal die statistische Information für eine bestimmte Eingabefolge zur Verfügung steht, kann unter verschiedenen Codierern der am besten geeignete Codierer ausgewählt werden, um die Eingabefolge zu komprimieren. Zum Beispiel können Codierverfahren – wie etwa traditionelles Huffman-Codieren, Vektor-Huffman-Varianten, RLE-Codieren usw. – optimiert und deren Codebücher trainiert werden für Eingabefolgen mit speziellen Charakteristika, wie etwa hohe Spektralwerte, niedrige Spektralwerte, gemischte oder alternierende Spektralwerte oder einige andere gewünschte/wahrscheinliche Eigenschaften. Im Gegensatz zu früherem Gebrauch eines einzelnen Codierers für alle Eingabefolgen decken die dargestellten Konfigurationen verschiedene Codierverfahren ab, entsprechend der besten Übereinstimmung zwischen einem statistischen Profil für eine Eingabefolge und dem statistischen Profil für die Daten, auf die ein Codierbuch trainiert wurde.

**[0053]** Nach der Festlegung, welcher Codierer **402**–**406** benutzt werden soll, setzt sich die Bearbeitung wie diskutiert in Bezug auf Fig. 2 für die Obertragung der Daten zu einem Empfänger **216** für das Decodieren fort. Zu bemerken ist, dass ein Inverse-Mode-Selector nicht aufgezeigt ist. Ein Modenwechsler ist notwendig (beispielsweise als Teil des Codierers **212** in Fig. 2), um eine ordentliche Auswahl für einen Codierer zu treffen, der die Arbeit des Modenselektors **400** reversiert. Dennoch können – wie oben diskutiert, Bereichstrennorte im Voraus bestimmt werden, so dass ihre Identifikation während des Codierens anwendbar bleibt. Oder, für das dynamische adaptive Codieren/Decodieren können eingebettete Flags verwendet werden, um die Codiererauswahl anzustoßen. Der Gebrauch von Flags ist gleichbedeutend mit der Verwendung eines Modenselektors und der Modenselektor kann so beschaffen sein, dass er sowohl für vorbestimmte als auch für adaptiv gelegte Teilungen arbeitet.

**[0054]** Die Fig. 5 ist ein Flussdiagramm, welches ein bevorzugtes Verfahren für die Erstellung eines Codebuchs eines Entropiecodierers für Eingabefolgen mit hoher Wahrscheinlichkeit an Nicht-Null-Frequenzkoeffizienten hat. Im Besonderen und im Gegensatz zum früheren Stand der Technik illustriert Fig. 5 die Erstellung eines Codes, welcher Codezuordnungen mit variabler Lange zu Symbolgruppierungen mit variabler Länge hat. (Der altere Stand der Technik verlangt entweder Festlangencodes oder Festlangenblocke für Eingabefolgen). Die bevorzugten Implementationen überwinden die Ressourcenbedingung beim Vektorcodieren großer Dimensionen und die Unanwendbarkeit des Codierens in Wörter mit gleicher Lange dadurch, dass sie einen entropiebasierten Variable-zu-Variable-Code bereitstellen, bei dem Codewörter mit variabler Lange benutzt werden, um Sequenzen mit variabler Länge X zu codieren. Die Ressourcenbedingung kann durch das Setzen eines festen Maximalumfangs des Codebuchs willkürlich beschnitten werden. Dieses Codebuch wird wie folgt erstellt.

**[0055]** Man lässt jedes $y_i$ Menge von Quellensymbolen repräsentieren $\{x_j\}$, mit $1 <= j <= N_j$, bei einer Wahrscheinlichkeit $P_i$ des Ereignisses innerhalb des Eingabestroms. Jede Menge wird einem korrespondieren Codewort mit einer Länge von $L_i$ Bits zugewiesen. Unter der Annahme, dass jedes $x_i$ aus einem festen Alphabet mit vorbestimmtem Umfang entnommen ist, ist es die Aufgabe, die Gleichung

$$L = \Sigma_i[(L_i \cdot P_i)/N_i\}$$

zu minimieren.

**[0056]** Anstelle der Suche nach einer allgemeinen Lösung des Problems wird das Problem unterteilt in zwei verschiedene Aufgaben. Die erste Aufgabe ist die Identifizierung einer (suboptimalen) Gruppierung einer Menge von Eingabesymbolen $\{x_i\}$ über einen empirischen Ansatz, der unten beschrieben wird. Die zweite Aufgabe ist die Zuweisung eines Codes vom Entropietyp für die gruppierten Symbole $\{y_i\}$. Es sei bemerkt, dass es bekannt ist, dass wenn die Quelle, nicht kohärent ist (beispielsweise ist die Eingabefolge unabhängig oder ohne Speicher), jede Gruppierung, die dieselbe Konfiguration von $\{N_j\}$ hat, die gleiche Codiereffizienz erreichen kann. In dieser Situation wird die erste Aufgabe inkonsequent.

**[0057]** Um die erste Aufgabe zu lösen, wird eine triviale Gruppierung **500** vorbereitet, etwa so $\{y_i\} = \{x_i\}$. Diese Ursprungskonfiguration setzt voraus, dass ein beispielhafter Eingabestrom benutzt wird, um die Erstellung des Codebuchs zu trainieren. Es wird vorausgesetzt, dass ein Computer mit Softwarekonstruktionen so programmiert werden kann, dass Datenstrukturen den Empfang eines jeden Symbols aus der Eingabefolge nachvollziehen. Solche Datenstrukturen können als Binar-Typ mit Baumstruktur, als Hash-Tabelle oder als Kombinationen dieser beiden implementiert werden. Andere äquivalente Strukturen können ebenfalls benutzt werden.

**[0058]** Nach der Festlegung der trivialen Gruppierung wird die Ereigniswahrscheinlichkeit für jedes yi berechnet **502**. Diese Wahrscheinlichkeit wird in Bezug auf jede beispielhafte Eingabefolge bestimmt,

und wird verwendet, um die Codebücherstellung zu trainieren. Wenn weitere Symbole zu der Symboldatenstruktur hinzugefugt werden, werden die Wahrscheinlichkeiten dynamisch angepasst.

**[0059]** Als Nächstes wird die wahrscheinlichste Gruppierung $y_i$ identifiziert **504** (bezeichnet mit $y_{mp}$). Wenn **506** das Symbol mit der höchsten Wahrscheinlichkeit eine Gruppierung von vorhergehenden Symbolen mit geringerer Wahrscheinlichkeit ist, dann wird die Gruppierung aufgeteilt **508** in Symbolbestandteile und die Verarbeitung wird von Schritt **502** erneut gestartet. (Obwohl Symbole kombiniert werden können, behalten die Gruppen die Erinnerung an alle Symbole in ihnen, so dass die Symbole extrahiert werden können).

**[0060]** Wenn das Symbol keine Gruppierung ist, dann läuft die Verarbeitung in Schritt **510** weiter, in dem die höchst wahrscheinliche Gruppierung versuchsweise mit einzelnen Symbolerweiterungen $x_i$'s erweitert wird. Vorzugsweise wird $y_{mp}$ mit jedem Symbol aus dem X-Alphabet erweitert. Dennoch kann ein Prediktor nur benutzt werden, um eine Erweiterungsmenge zu generieren, die nur wahrscheinliche Erweiterungen enthalt, in dem Fall, dass das Alphabet sehr groß ist und es bekannt ist, dass viele Erweiterungen unwahrscheinlich sind. Zum Beispiel kann ein solcher Prediktor auf semantischer oder kontextueller Bedeutung basieren, so dass sehr unwahrscheinliche Erweiterungen von vornherein ignoriert werden können.

**[0061]** Die Wahrscheinlichkeit für jede versuchsweise Erweiterung von $y_{mp}$ wird dann berechnet **512** und nur die wahrscheinlichste Erweiterung einbehalten **514**. Der Rest der Erweiterungen mit geringer Wahrscheinlichkeit wird eingesammelt **516**, als eine kombinierte Gruppierung in dem Codebuch mit einem speziellen Symbol (Ereignis) gespeichert, um eine kombinierte Gruppierung anzuzeigen. Dieses Wild-Card-Symbol repräsentiert jede beliebige Symbolgruppierung, die $y_{mp}$ als Präfix hat, aber mit einer verschiedenen Erweiterung (Suffix) unterschiedlich von der wahrscheinlichsten Erweiterung. Das bedeutet, wenn $y_{mp} + x_{mp}$ die wahrscheinlichste Wurzel und Erweiterung ist, dann werden die anderen geringer wahrscheinlichen Erweiterungen repräsentiert durch $y_{mp}^*$, $* \neq x_{mp}$. (Es sei zur Klarstellung bemerkt, dass diese Diskussion davon ausgeht, dass eine serielle Verarbeitung der Einzelsymbolerweiterungen vorgenommen wird. Dennoch kann die Parallelverarbeitung von Vielfachsymbolerweiterungen erwogen werden.)

**[0062]** Der Fachmann wird verstehen, dass die Anwendung von Einzelsymbolerweiterungen und das Zurückhalten einer wahrscheinlichsten Gruppierung eine Einschränkung ist, die nur der Klarheit der Diskussion dient. Es wird weiter vorausgesetzt, dass die Codebücherstellung auch parallel erfolgen kann, obwohl sich die Diskussion auf serielle Verarbeitung bezieht.

**[0063]** Die Codebücherstellung wird vervollständigt durch Wiederholung **518** der Schritte **502** bis **516** bis alle möglichen Erweiterungen erstellt worden sind Oder die Zahl der Codebucheinträge eine vorbestimmte Grenze erreicht haben. Also: man berechnet wiederholt die Wahrscheinlichkeiten für die augenblicklichen $y_i$ **502**, bei dem die Codebuchmenge {Y} nunmehr $y_{mp} + x_{mp}$ umfasst, beziehungsweise man wählt aus **504** und gruppiert die wahrscheinlichsten und die unwahrscheinlichsten Erweiterungen. Der Effekt der wiederholten Anwendung der oben bezeichneten Operationen ist die automatische Zusammenfassung der Symbolgruppierungen, die eine hohe Korrelation haben, so dass die Korrelation zwischen den Gruppen minimiert ist. Dieses minimiert den Zahler von L, während die Lange der wahrscheinlichsten $y_i$ simultan maximiert wird, so dass der Nenner von L maximiert wird.

**[0064]** Die **Fig. 6** bis **Fig. 13** zeigen die Erstellung eines Codebuchs gemäß **Fig. 5** für ein beispielhaftes Alphabet {A, B, C}. Für die folgende Diskussion soll das Codebuch in Bezug auf einen beispielhaften Eingabestrom „AAABBAACABABBAB" definiert werden. Wie oben diskutiert, können eine oder mehrere beispielhafte Eingabefolgen verwendet werden, um ein Codebuch zu erstellen, welches dann von Decodierern und Codierern verwendet wird, um beliebige Eingabefolgen zu verarbeiten. Zum Zwecke der Klarheit wird das Codebuch durch eine Baumstruktur repräsentiert, obwohl es tatsachlich auch als lineare Tabelle, als hash-table, Datenbank usw. implementiert sein kann. Wie dargestellt, ist der Baum von links nach rechts orientiert, wobei die linke Spalte (beispielsweise "A" und „X0") die obere Zeile der baumartigen Struktur repräsentiert und nacheinander eingerückte Zeilen repräsentieren die „Kinder" der vorhergehenden Zeilenknoten (beispielsweise ist in einem top-down-Baum der **Fig. 7** der oberste Knoten „A" ein erstzeiliger Elternknoten für einen zweitzeiligen Zwischenkindknoten „B".).

**[0065]** Bei der Erstellung des Codebuchs gilt die generelle Regel, den wahrscheinlichsten Blattknoten zu nehmen, ihn zu expandieren, die Wahrscheinlichkeiten erneut zur Bestimmung des wahrscheinlichsten Blattknotens zu berechnen und dann die verbleibenden Geschwisterknoten in einen einzelnen Xn-Knoten (n = 0 ... N, wobei jedes Mai Knoten miteinander kombiniert werden) zu verdichten. Wenn sich herausstellt, dass der wahrscheinlichste Knoten ein Gruppenknoten ist, dann wird die Gruppe geteilt, die Wahrscheinlichkeiten erneut berechnet und der wahrscheinlichste Teilnehmerknoten beibehalten (beispielsweise werden die verbleibenden Gruppen umgruppiert). Die Verarbeitung wiederholt sich bis

ein Haltezustand erreicht ist, etwa wenn ein Codebuch die vorbestimmte Größe hat.

**[0066]** Die Fig. 6 zeigt eine ursprüngliche Gruppierung für den Eingabestrom „A A A B B A A C A BA B B A B". Das anfängliche Parsing der Eingabefolge liefert Wahrscheinlichkeiten für das Auftreten von A = 8/15, B = 6/15 und C = 1/15. Die ursprüngliche triviale Gruppierung kann basierend auf verschiedenen Kriterien erstellt werden, wobei das einfachste Kriterium das wäre, wenn für jeden Buchstaben im Alphabet ein Erstlevelknoten vorhanden ist. Dennoch kann, wenn das Eingabealphabet groß ist, die triviale Gruppierung durch wenige Untermengen von Symbolen begrenzt werden, die höchste Wahrscheinlichkeit haben, wobei die verbleibenden Symbole in einer X-Gruppierung kombiniert werden. Fig. 6 illustriert dieses Verfahren mit dem Beginn zweier Initialgruppen: Gruppe A **600** mit einer Wahrscheinlichkeit 8/15 und Gruppe X0 **602**, die die Wahrscheinlichkeit 7/15 hat, wobei X0 alle verbleibenden Symbole geringer Wahrscheinlichkeit im Alphabet repräsentiert, beispielsweise B oder C.

**[0067]** Nach der Bereitstellung einer trivialen Anfangsgruppierung wird der Blattknoten mit der höchsten Wahrscheinlichkeit für die Erweiterung ausgewählt (siehe auch Fig. 5 mit Diskussion bezüglich der Rechenreihenfolge). Danach wird, wie in Fig. 7 gezeigt, die Gruppe A **600** versuchsweise durch jeden Buchstaben im Alphabet expandiert (oder man begrenzt die Erweiterung auf einige Untermengen von ihnen, wie für die Erstellung der Ursprungsgruppierung beschrieben). Die Wahrscheinlichkeiten werden dann bezüglich des Eingabestroms „AAABBAACABABBAB" wiederholt berechnet, um die Werte für die versuchsweisen Erweiterungen A **606**, B **608** und C **610** zu bestimmen. Das Ergebnis sind neun Parsing-Gruppen, wobei "A A" mit 2/9 erscheint, "A B" mit 4/9 erscheint und „A C" mit 0/9 erscheint. Deshalb wird die wahrscheinlichste Erweiterung „A B" beibehalten und die anderen Erweiterungen fallen zusammen in X1 = A, C. Obwohl diese Beschreibung die wiederholte Berechnung aller Wahrscheinlichkeiten diskutiert, sei bemerkt, dass ein effizienterer Ansatz dann besteht, Wahrscheinlichkeiten und Symbolverknüpfungen für jeden Knoten innerhalb des Knotens zurückzubehalten, und die Information nur soweit zu berechnen, wie notwendig.

**[0068]** Die Fig. 8 zeigt das Zusammenfallen in X1 **612** für die Fig. 7. Die Berechnung wiederholt sich mit der Identifizierung des Knotens, der die höchste Wahrscheinlichkeit hat, beispielsweise des Knotens B **608** mit der Wahrscheinlichkeit 4/9.

**[0069]** Wie in Fig. 9 gezeigt, wird Knoten **608** versuchsweise erweitert mit den Symbolen A **614**, B **616**, C **618** und wie oben diskutiert, wird die versuchsweise Gruppierung mit der höchsten Wahrscheinlichkeit zurück behalten. Nach der wiederholten Berechnung der Wahrscheinlichkeiten besteht das Ergebnis in acht Parsing-Gruppen, in denen die Symbolsequenz „A B A" **614** einmal erscheint, „A B B" **616** einmal erscheint und „A B C" **618** überhaupt nicht erscheint. Da die versuchsweisen Erweiterungen A **614** und B **616** die gleiche Ereigniswahrscheinlichkeit haben, muss eine Regel für die Auswahl des zurückzubehaltenden Symbols definiert werden. Für diese Diskussion und immer dann, wenn gleiche Wahrscheinlichkeiten vorliegen, wird der höchste Zeilenknoten zurückbehalten, (beispielsweise der am weitesten links stehende Kindknoten in einem Baum mit top-down-Struktur). Gleichermaßen wird der am weitesten links stehende Zeilenknoten zurückbehalten (beispielsweise der der Wurzel des top-down-Baums nächststehende Knoten), wenn ein Konflikt zwischen Baumzeilen besteht.

**[0070]** Deshalb wird, wie in Fig. 10 gezeigt, der Knoten A **614** (Fig. 9) zurückbehalten und die Knoten B **616** und C **618** werden in den Knoten X2 = B, C **620** kombiniert, die eine kombinierte Wahrscheinlichkeit von 1/8 + 0/8 haben. Nun muss im nächsten Schritt der Knoten expandiert werden, der gegenwärtig die höchste Wahrscheinlichkeit in Bezug auf den Eingabestrom hat. Wie gezeigt haben Knoten X1 = A, C **612** und X0 = B, C **602** dieselbe Ereigniswahrscheinlichkeit (3/8). Wie oben diskutiert, wird eine Ruckfallregel benutzt, so dass der höchste Knoten im Baum (X0 **602**) expandiert wird. (Obwohl es nur wegen der Vertraglichkeit notwendig ist, 1st es von Vorzug, wenn Knoten höherer Stufe expandiert werden, denn dieses kann durch die Erhöhung der Anzahl langer Codewörter die Codiereffizienz erhöhen.) Da jedoch X0 **602** ein vereinter Knoten ist, muss dieser geteilt statt expandiert werden. Fig. 11 zeigt das Ergebnis der Teilung des Knotens X0 in seine Symbolbestandteile B **622** und C **624**. Das wiederholte Berechnen der Wahrscheinlichkeiten zeigt auf, dass die Symbolsequenz „A B A" mit 1/8 erscheint, „A B X2" mit 1/8 erscheint „A X1" mit 3/8 erscheint, „B" **422** mit 2/8 erscheint und „C" mit 1/8 erscheint. Da dies eine Teilungsoperation ist, wird der geteilte Knoten mit der höchsten Wahrscheinlichkeit, beispielsweise Knoten B **622**, zurückbehalten und der (die) verbleibende(n) Knoten wird (werden) in X0 **602** zurück rekombiniert.

**[0071]** Die Fig. 12 zeigt das Ergebnis des zurückbehaltenen Knotens B **622** mit hoher Wahrscheinlichkeit. Es sei bemerkt, dass die Gruppierung X0 **602** nunmehr nur ein einziges Symbol „C" repräsentiert. Nach der Durchsicht der Wahrscheinlichkeiten muss der Knoten mit der höchsten Wahrscheinlichkeit identifiziert werden, und geteilt oder expandiert werden. Wie gezeigt, erscheint Symbolsequenz „A B A" mit 1/8, „A B X2" erscheint mit 1/8, „A X1" erscheint mit 3/8, „B" erscheint mit 2/8 und „X0" erscheint mit 1/8. Des-

halb ist der Knoten X1 **612** ein kombinierter Knoten, der geteilt werden muss.

**[0072]** Die Teilung wird, wie oben diskutiert, weitergeführt und die Berechnung der Eingabefolge erfolgt in Verbindung mit Fig. 5 zyklisch, wo die Knoten mit höchster Wahrscheinlichkeit expandiert oder geteilt werden, bis ein Haltezustand erreicht ist (beispielsweise bis das Codebuch einen Maximalumfang erreicht). Wenn einmal das Codebuch einen Haltezustand erreicht hat, steht es für die Codierung der über einen Kommunikationskanal zu übermittelnden Daten zur Verfügung.

**[0073]** Die Fig. 13 zeigt ein Schwellengitter, das verwendet werden kann, das Verfahren gemäß Fig. 5 der Codebücherstellung abzuwandeln. Wie bei den Fig. 3 und Fig. 4 diskutiert, wird die Codierung effektiver, wenn die Codierer darauf zugeschnitten werden können, bestimmte Abschnitte der Eingabedaten zu bearbeiten. Wenn bekannt ist, dass das Codierverfahren eine signifikante Zahl von wiederholten Werten erzeugt, kann im Besonderen, ein Entropiecodierer mit einem Codierverfahren vom RLE-Typ kombiniert werden, um die Codiereffizienz für Daten mit sich wiederholenden Werten zu erhöhen.

**[0074]** In den vorgestellten Ausführungsformen fugt die Quantisierung der Eingangsdaten Null-Werte oder nahezu Null-Werte für Spektralkoeffizienten ein in signifikante Teile der Frequenzbereiche der Eingangsdaten. Konsequenterweise wird anstelle der Benutzung desselben Entropiecodierers (beispielsweise ein Codierer und ein Codebuch gemäß Fig. 5) für die nahezu fast Nicht-Null-Werte ein RLE-basierter Entropiecodierer benutzt.

**[0075]** Für die Erstellung eines Codebuchs für einen RLE-basierten Entropiecodierer bilde man aus den absoluten Werten der Nicht-Null-Spektralproben eine ganzzahlige Menge $L_i = \{1, 2, 3 \ldots L_n\}$, wobei $L_n$ für beliebige Werte steht, die größer oder gleich $L_n$ sind. Man bilde aus der Lauflange der Null-wertigen Spektralproben im Eingabestrom eine andere Menge $R_i = \{1, 2, 3 \ldots, R_m\}$, wobei $R_m$ für jeden Null-Lauf mit Lange größer als oder gleich $R_m$ steht. Bei der Benutzung dieser Notation können wir ein Eingabespektrum mit einem String von Eingabesymbolen repräsentieren, das als $(R_i, L_j)$ definiert wird, welches zu $R_i$ Null-Spektralproben korrespondiert gefolgt von $L_i$ (beispielsweise Symbole codiert mit dem Entropiecodierer).

**[0076]** Wie oben zu den Fig. 5 und folgenden beschrieben, ist der erste Schritt die Erstellung eines Codebuchs, um die Wahrscheinlichkeit für alle Eingabeereignisse zu sammeln. Hierbei ist die Eingabefolge angepasst in Bezug auf definierte Schwellen. Deshalb bestimmt sich die Wahrscheinlichkeit für $(R_i, L_j)$ für alle $1 <= i <= n$ und $1 <= j <= m$. Diese Wahrscheinlichkeiten sind in Fig. 13 grafisch dargestellt. Dort entsprechen dunklere Quadrate (beispielsweise **806**, **808**) Ereignissen, die eine höhere Wahrscheinlichkeit besitzen und hellere Quadrate (beispielsweise **810**, **812**) Ereignissen mit geringer oder nahezu Null-wertiger Wahrscheinlichkeit. Alle Eingabekonfigurationen mit hoher Wahrscheinlichkeit werden mit dem Bezugszeichen **800** zusammengefasst und alle Konfigurationen mit kleinen Wahrscheinlichkeiten als Bereich **802**. Alle Kombinationen mit geringer Wahrscheinlichkeit werden aus dem Codebuch ausgeschlossen. Es wird eine Wahrscheinlichkeitsschwelle **804** so definiert, dass jeder Wert unterhalb der Trennlinie auf Null gesetzt und aus dem Codebuch ausgeschlossen wird. Die über der Schwelle verbleibenden Konfigurationen, werden einem Code vom Entropietyp zugewiesen, der eine Lange hat, die umgekehrt proportional ist zu seiner Wahrscheinlichkeit. Im Falle quantisierter Audiodaten haben Eingabefolgen mit großer Amplitude eine kleine Wahrscheinlichkeit. Konsequenterweise fallen diese unter die Schwelle und werden aus dem Codebuch ausgeschlossen (dennoch können sie herausgetrennt und in den codierten Bitstrom eingebettet werden).

**[0077]** Um eine entropiecodierte Ausgabefolge unter Benutzung eines zweiten Codierers zu verschachteln, wird ein spezielles Codierbuch mit Entropiecodes reserviert, um die ausgeschlossenen Ereignisse zu markieren (beispielsweise RLE-codierte Daten). Während des Codierlaufs können spektrale Proben (Eingangssymbole) mit einer Liste von möglichen Ereignissen verglichen werden und wenn eine Obereinstimung gefunden wird (im Baum oder in der Tabelle, in der Hash-Struktur oder in einem Äquivalent, die jeweils verwendet werden, um das Codebuch zu repräsentieren, beispielsweise durch Gebrauch eines Variable-zu-Variable-Codierers), wird der Code vom zugehörigen Entropietyp ausgegeben gefolgt von einem Vorzeichen-Bit.

**[0078]** Wenn eine Obereinstimmung nicht gefunden wird, wird ein Escape-Code gesendet gefolgt von notwendiger Information zum Identifizieren des Ereignisses, beispielsweise eine Information zur RLE-Codierung der Daten. In dem Fall, wenn ein Eingabespektrum mit N Nullen endet, ist entweder ein explizites (spezielles) Endsignal oder ein spezielles Ereignis, etwa wie ein (N, 1)-Suffix notwendig, weil der Codierer in der Lage ist, die Gesamtzahl der Proben zu erkennen und das Codieren beendet, wenn diese Grenze überschritten wird.

**[0079]** Ein Schwellengitter ist zum Codieren nicht notwendig, da das Gitter nur gebraucht wird, um Codebucheintrage auszulesen. Die hier offenbarten Codierverfahren können mit einem Codebuch benutzt werden, das gemäß der Beschreibung in Fig. 13 erstellt wurde.

**[0080]** Die Fig. 14 zeigt ein Verfahren zur Implementierung des Entropiecodierers **208** aus Fig. 2, welches an quantisierten Daten unter Verwendung eines Codebuchs entsprechend Fig. 5 hergeleitet ist. (Es sei bemerkt, dass das Variable-zu-Variable-Codierverfahren allgemein für das Codieren jeder Art von Daten anwendbar ist). Wie dargestellt, werden die quantisierten Daten als Eingabefolge für den Entropiecodierer von Fig. 2 empfangen **900**. Es wird vorausgesetzt, dass die Eingabefolge aus einer Form von diskreten Signalen oder Datenpaketen besteht und dass zur Vereinfachung der Diskussion alle Eingabefolgen einfach als eine lange Folge von diskreten Symbolen angenommen werden. Die empfangene Eingabefolge **900** wird durchsucht **902**, um einen korrespondierenden Codebuchschlüssel im Codebuch der Fig. 5 zu lokalisieren. Solche Durchsuchung entspricht einem Datennachschlagen (data-look-up) und – abhängig von der verwendeten Datenstruktur für das Implementieren des Codebuchs – kann das exakte Verfahren des Nachschlagens variieren.

**[0081]** Es sei bemerkt, dass verschiedene Techniken zur Speicherung und Manipulation des Codebuchs eines Codierers zur Verfugung stehen. Zum Beispiel: Eine Struktur für ein Variable-zu-Variable-Codebuch ist traversal und die Speicherung erfolgt als N-knötiger (beispielsweise binärer, tertiärer usw.) Baum, wobei Symbolgruppierungen eine Traverse (traversal) durch die Baumstruktur leiten. Der Pfad zu einem Blattknoten des Baumes repräsentiert das Ende einer erkannten Symbolsequenz, wobei ein Code vom Entropietyp mit der Sequenz verbunden ist. (Es sei bemerkt, dass das Codebuch als Tabelle implementiert sein kann, wobei ein Tabelleneintrag die gesamte Eingabesequenz enthält, beispielsweise den Pfad zu einem Knoten). Knoten können softwaremäßig codiert werden als eine Struktur, eine Klassendefinition oder als andere Struktur, die das Speichern eines Symbols oder von mit Knoten verbundenen Symbolen und die Verknüpfung eines korrespondierenden Codes **905** vom Entropietyp **906** erlaubt.

**[0082]** Außerdem kann das Codebuch für den RLE-Codierer als ein zweidimensionales Gitter in einem Permanent-Speicher gespeichert werden, worin die Datenermittlung durch die Identifizierung zweier Indizes durchgeführt wird. Deshalb kann man Tabelleneintrage durch die Spezifikation einer Lauflange und eines bestimmten Symbolwertes ermitteln. Eine Codiertabelle kann als Huffman-Baum implementiert werden. Eine andere Codebuch-Implementation kann Rice-Golomb-Strukturen oder deren Äquivalente einschließen.

**[0083]** Obwohl nicht bei der Diskussion der Fig. 2 ausdrücklich gezeigt, funktioniert das Codieren als eine inverse Operation des Decodierens, wobei die codierten Daten **908** in einem Decodiercodebuch aufgesucht werden **906**, um eine Approximation der ursprünglichen Eingabefrequenzkoeffizienten **900** herzustellen.

**[0084]** Wenn die Erfindung auch mit Bezug auf eine erläuternde Ausführungsform beschrieben und bebildert wurde, sollte klar sein, dass die erläuternde Ausführungsform in Anordnung und Detail modifiziert werden kann, ohne die Grundlagen zu verlassen. Entsprechend sind alle diese Änderungen innerhalb des Umfangs der Erfindung Bestandteil der folgenden Ansprüche.

## Patentansprüche

1. Verfahren zum Decodieren einer Sequenz codierter Audiofrequenz-Koeffizienten, wobei das Verfahren umfasst:
Empfangen einer Sequenz codierter Audiofrequenz-Koeffizienten mit einem Frequenzdomänenbereich (**300**), der in wenigstens einen ersten und einen zweiten Frequenzdomänen-Teilbereich (**310**, **312**) unterteilt ist, für Entropie-Decodierung,
**dadurch gekennzeichnet**, dass:
der Frequenzdomänen-Bereich (**300**) für Entropie-Decodierung mit verschiedenen Entropie-Decodiereinrichtungen (**1**) unterteilt ist, wobei diese Unterteilung entsprechend statistischer Analyse durchgeführt worden ist, die separate Frequenzdomänen-Teilbereiche auf Basis von Entropie-Codierungseffizienz identifiziert,
wobei jeder von dem wenigstens ersten und zweiten Frequenzdomänen-Teilbereich (**310**, **312**) eine dazugehörige Entropie-Decodiereinrichtung aus der Vielzahl der verschiedenen Entropie-Decodiereinrichtungen aufweist und das Verfahren des Weiteren umfasst:
Decodieren von Koeffizienten des ersten Frequenzdomänen-Teilbereiches (**310**) mit einer ersten zugehörigen Entropie-Decodiereinrichtung, um erste decodierte Frequenz-Koeffizienten zu erzeugen;
Decodieren von Koeffizienten des zweiten Frequenzdomänen-Teilbereiches (**312**) mit einer zweiten zugehörigen Entropie-Decodiereinrichtung, um zweite decodierte Frequenzkoeffizienten zu erzeugen, wobei sich die zweite zugehörige Entropie-Decodiereinrichtung von der ersten zugehörigen Entropie-Decodiereinrichtung unterscheidet, wobei das Decodieren mit der zweiten zugehörigen Entropie-Decodiereinrichtung Lauflängen-Entropie-Decodieren umfasst und wobei das Decodieren mit der zweiten Entropie-Decodiereinrichtung ein Decodieren eines Entropie-Code umfasst, der eine Kombination darstellt von:
(a) einer Lauflänge eines Laufes von Null-Wert-Koeffizienten in dem zweiten Frequenzdomänen-Teilbereich (**312**), und
(b) einem Level eines Nicht-Null-Wert-Koeffizienten, der auf den Lauf von Null-Wert-Koeffizienten in dem zweiten Frequenzdomänen-Teilbereich (**312**) folgt;

Dequantisieren der ersten und der zweiten decodierten Frequenzkoeffizienten, um dequantisierte Frequenzkoeffizienten zu erzeugen; und
Umkehren von Frequenztransformation der dequantisierten Frequenzkoeffizienten.

2. Verfahren nach Anspruch 1, wobei Teilbereichs-Unterteilungen vorbestimmt sind.

3. Verfahren nach Anspruch 2, wobei Auswahl einer geeigneten zugehörigen Entropie-Decodiereinrichtung automatisch entsprechend dem Frequenzdomänen-Teilbereich der decodierten Koeffizienten stattfindet.

4. Verfahren nach Anspruch 1, wobei Teilbereichs-Unterteilungen adaptiv sind.

5. Verfahren nach Anspruch 4, wobei Auswahl einer geeigneten zugehörigen Entropie-Decodiereinrichtung entsprechend Flags in einem Datenstrom für die Sequenz stattfindet.

6. Verfahren nach einem der Ansprüche 1 bis 5, wobei das Decodieren mit der ersten zugehörigen Entropie-Decodiereinrichtung Entropie-Decodieren von variabel zu variabel umfasst.

7. Verfahren nach einem der Ansprüche 1 bis 5, wobei das Decodieren mit der ersten zugehörigen Entropie-Decodiereinrichtung Vektor-Huffman-Decodieren umfasst.

8. Verfahren nach einem der Ansprüche 1 bis 7, das des Weiteren umfasst: Umschalten zwischen den verschiedenen Entropie-Decodiereinrichtungen unter Verwendung einer Modus-Umschalteinrichtung.

9. Verfahren nach einem der Ansprüche 1 bis 8, wobei das Empfangen einschließt, dass die Sequenz von einem Streaming-Ausgabesystem empfangen wird, das die Sequenz als Streaming-Pakete über einen Kanal (**210**) zur Echtzeit-Wiedergabe ausgibt.

10. Verfahren nach Anspruch 9, wobei das Streaming-Ausgabesystem Bandbreiten-Anpassung gewährleistet.

11. Verfahren nach einem der Ansprüche 1 bis 10, wobei Koeffizienten jedes verschiedenen unterteilten Frequenzdomänen-Teilbereiches unter Verwendung einer anderen Entropie-Decodiereinrichtung decodiert werden.

12. Verfahren nach einem der Ansprüche 1 bis 11, wobei der erste Frequenzdomänen-Teilbereich (**310**) vorherrschend Nicht-Null-Koeffizienten enthält und der zweite Frequenzdomänen-Teilbereich (**312**) vorherrschend Null-Koeffizienten enthält.

13. Computerlesbares Medium, auf dem durch Computer ausführbare Befehle gespeichert sind, die ein Computersystem, das durch sie programmiert wird, veranlassen, das Verfahren nach einem der Ansprüche 1 bis 12 durchzuführen.

Es folgen 9 Blatt Zeichnungen

**Anhängende Zeichnungen**

FIG. 1

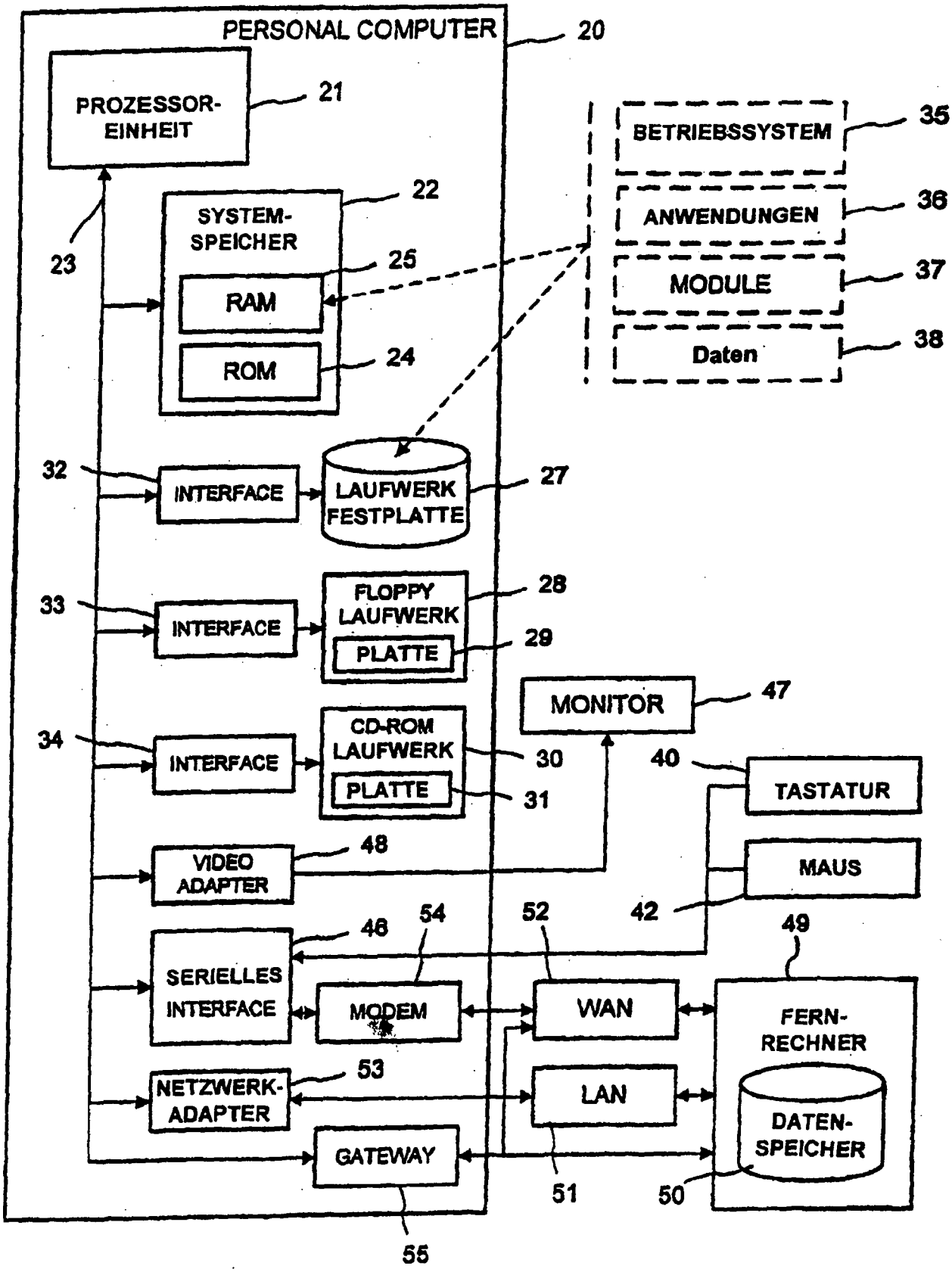


FIG. 2

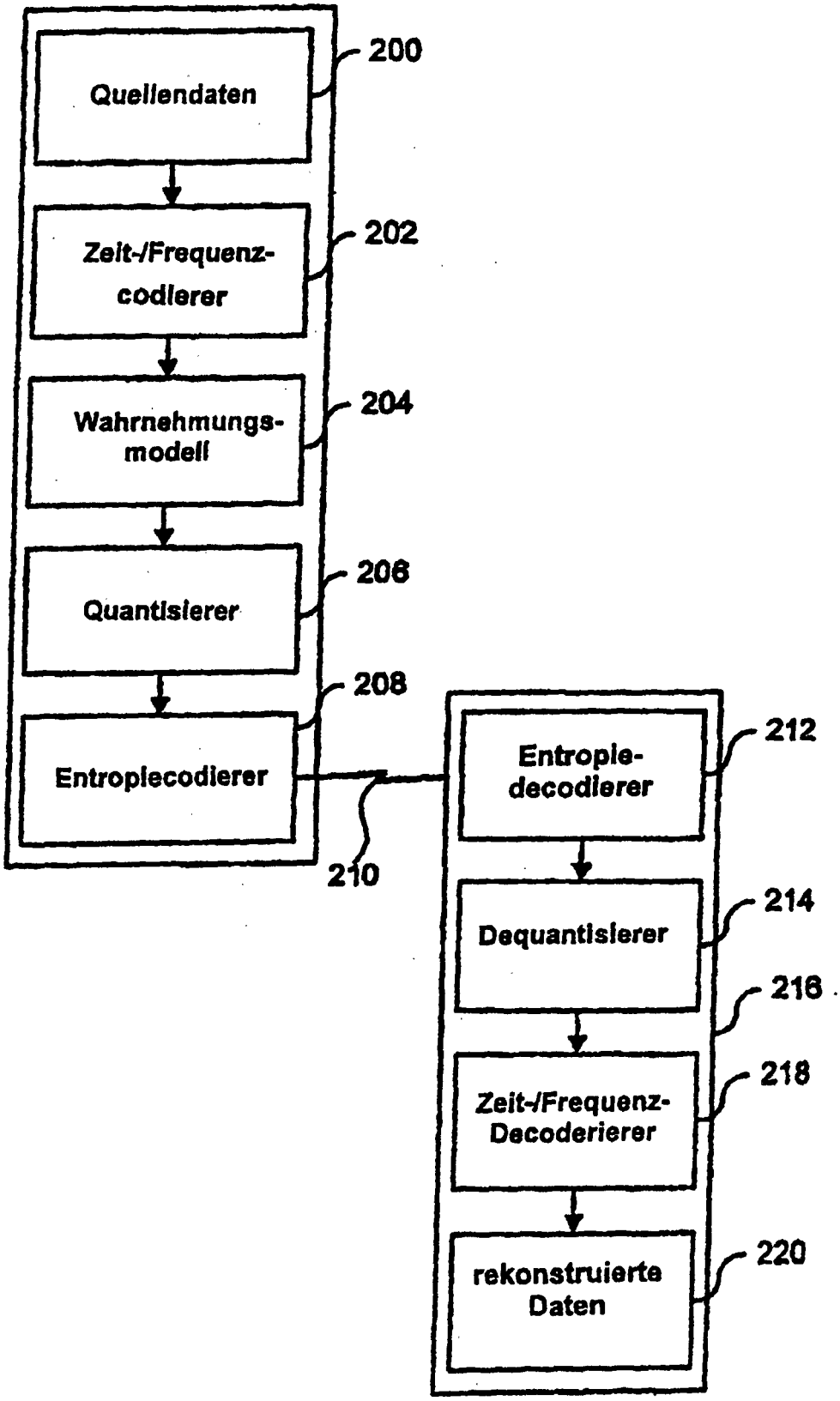

Quellendaten
200
Zeit-/Frequenz-codierer
202
Wahrnehmungs-modell
204
Quantisierer
206
Entropiecodierer
208
210
Entropie-decodierer
212
Dequantisierer
214
216
Zeit-/Frequenz-Decoderierer
218
rekonstruierte Daten
220

FIG. 3

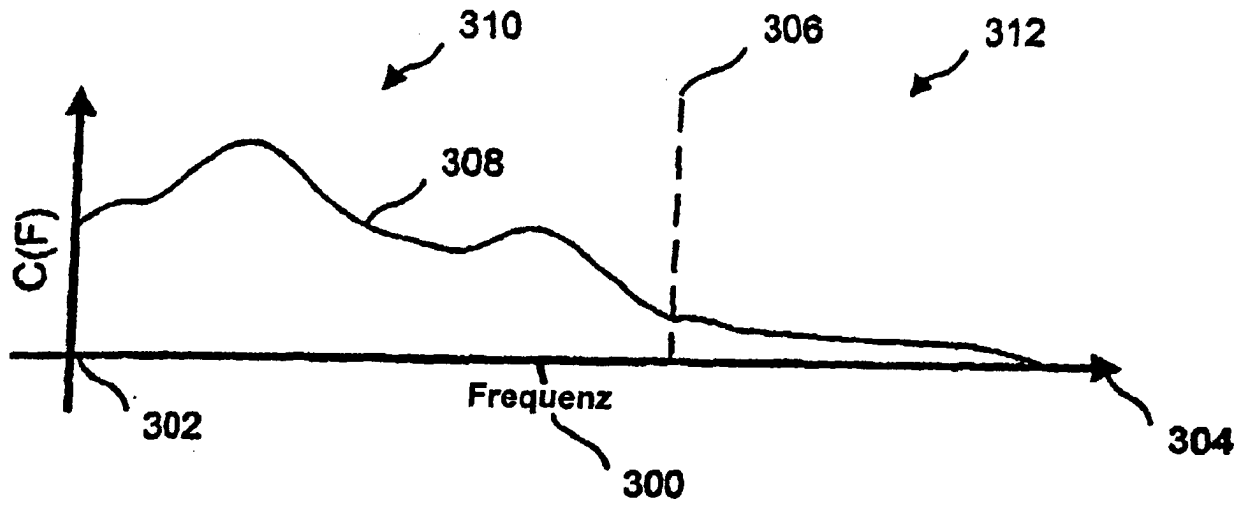


FIG. 4

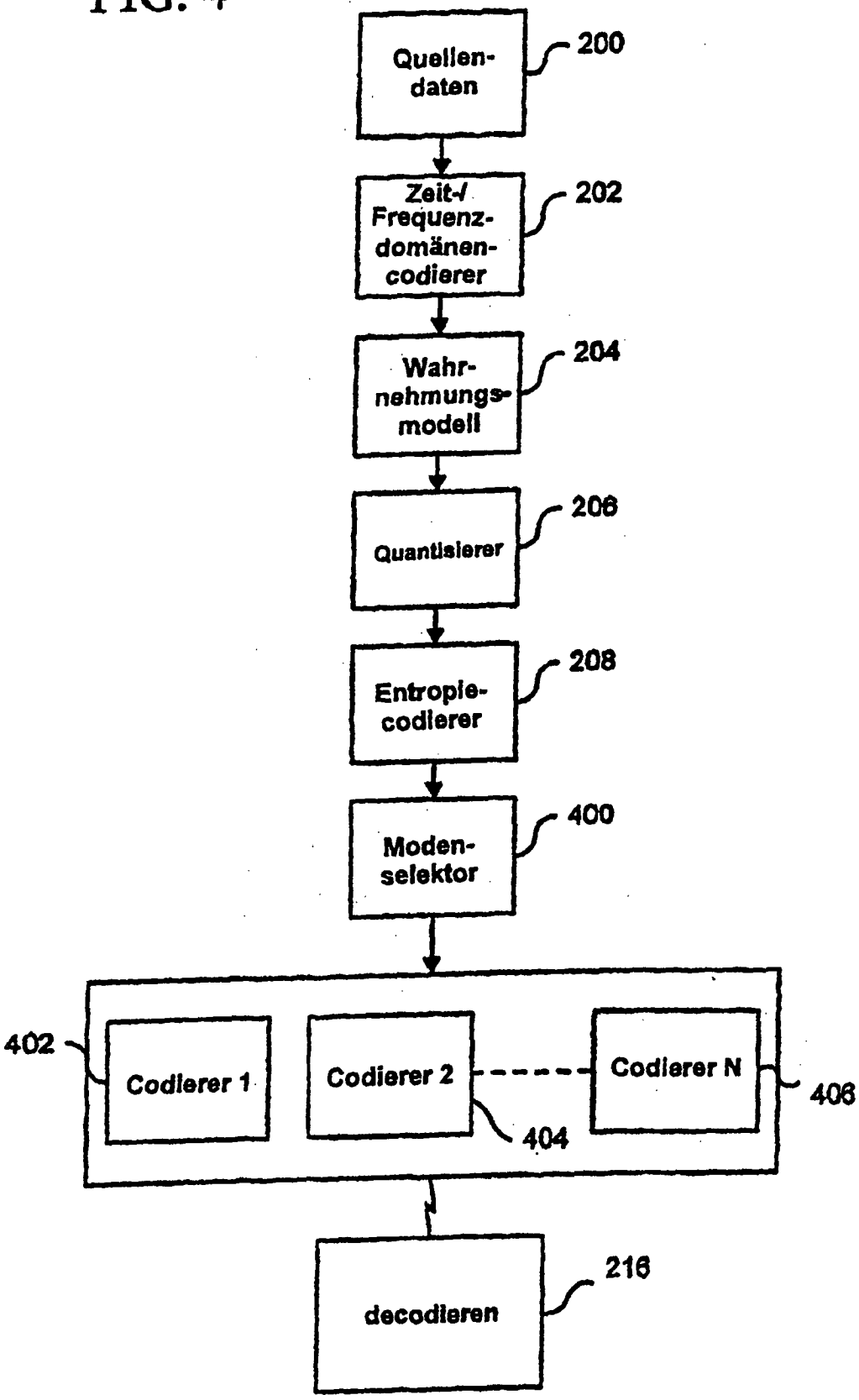

Quellen-daten
200
Zeit-/ Frequenz-domänen-codierer
202
Wahr-nehmungs-modell
204
Quantisierer
206
Entropie-codierer
208
Moden-selektor
400
402
Codierer 1
Codierer 2
Codierer N
406
404
decodieren
216

FIG. 5

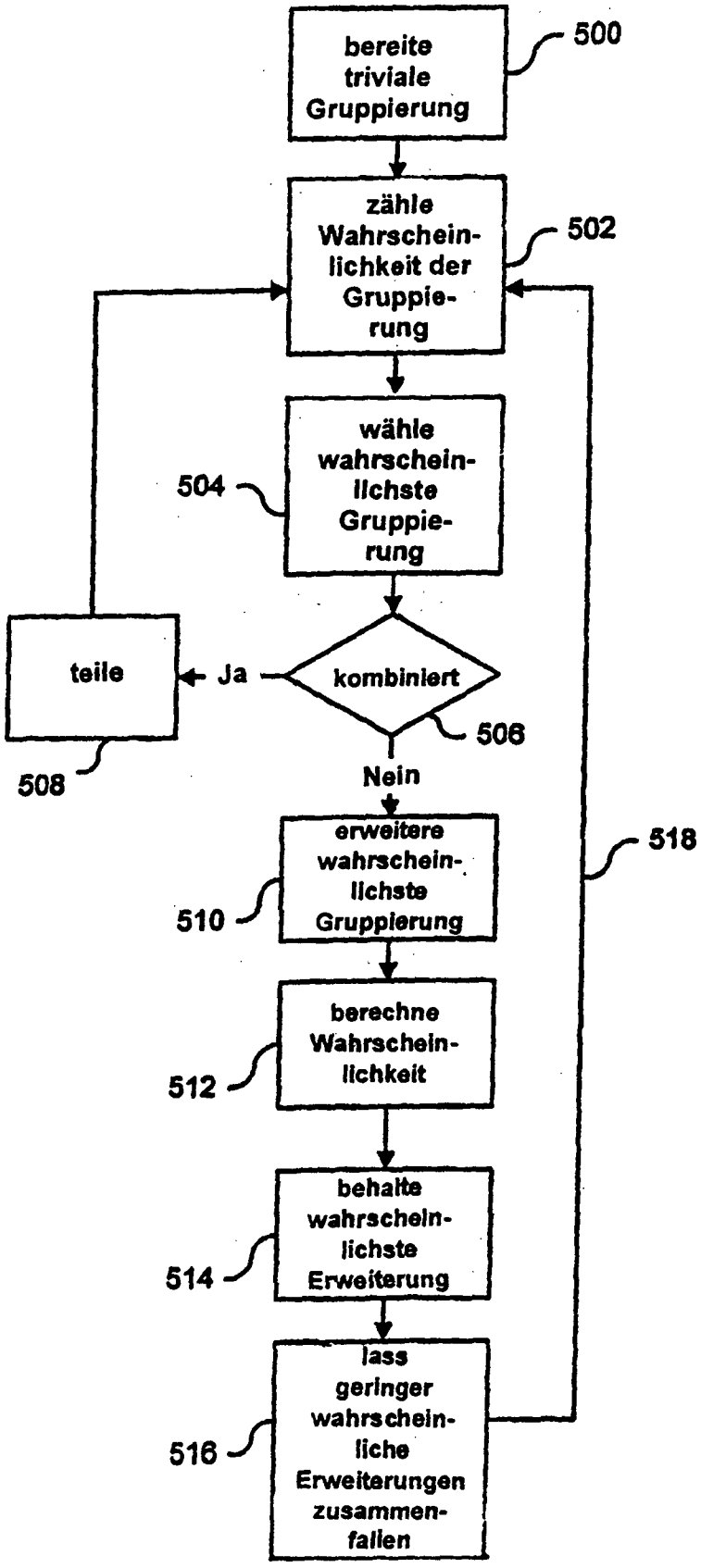
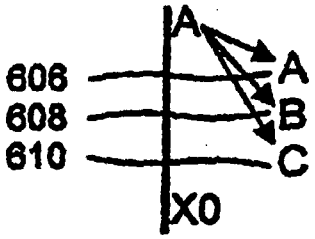

bereite triviale Gruppierung
500
zähle Wahrschein-lichkeit der Gruppie-rung
502
wähle wahrschein-lichste Gruppie-rung
504
kombiniert
Ja
teile
506
508
Nein
erweitere wahrschein-lichste Gruppierung
510
518
berechne Wahrschein-lichkeit
512
behalte wahrschein-lichste Erweiterung
514
lass geringer wahrschein-liche Erweiterungen zusammen-fallen
516

FIG. 6

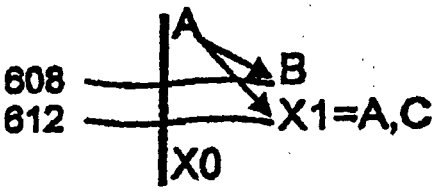

8/15 erweitern
7/15

FIG. 7



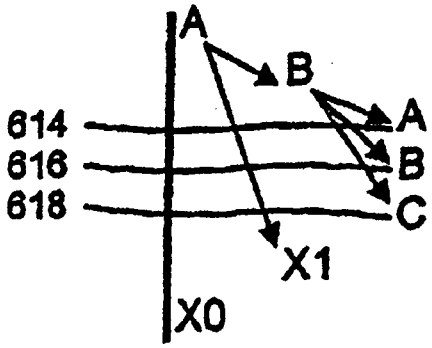

2/9
4/9 beibehalten
0/9

FIG. 8




4/9 erweitern
2/9
3/9

FIG. 9




1/8 beibehalten
1/8
0/8
3/8
3/8

FIG. 10

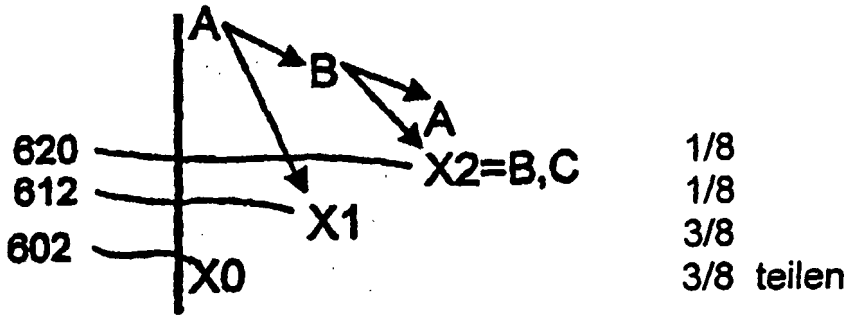



FIG. 11

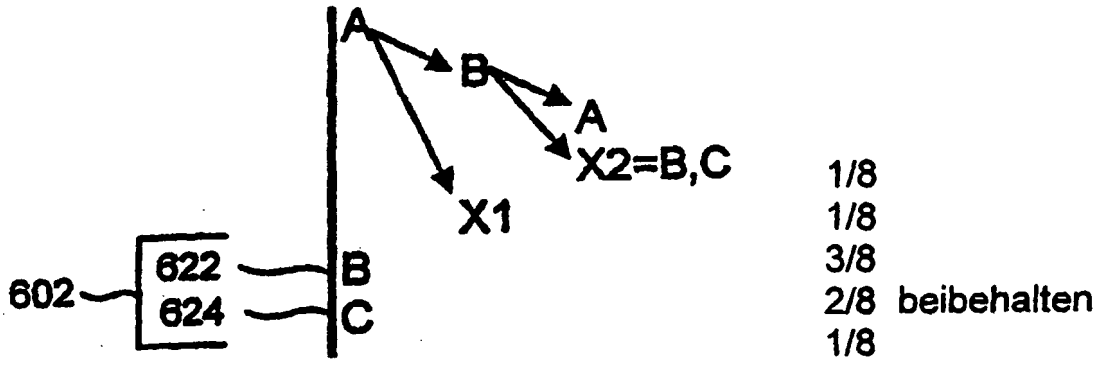



FIG. 12

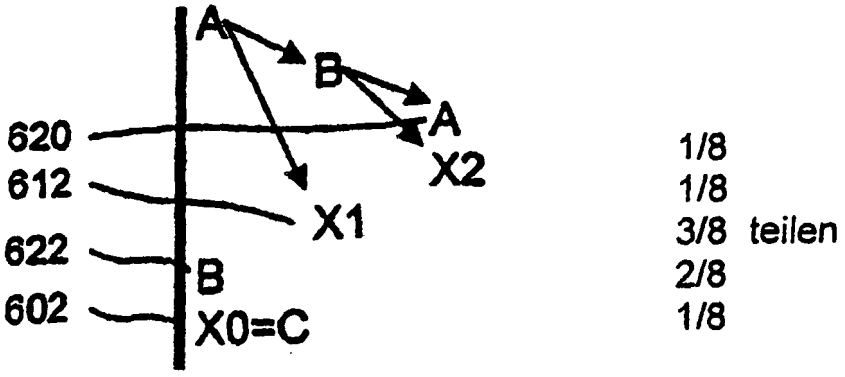


FIG. 13

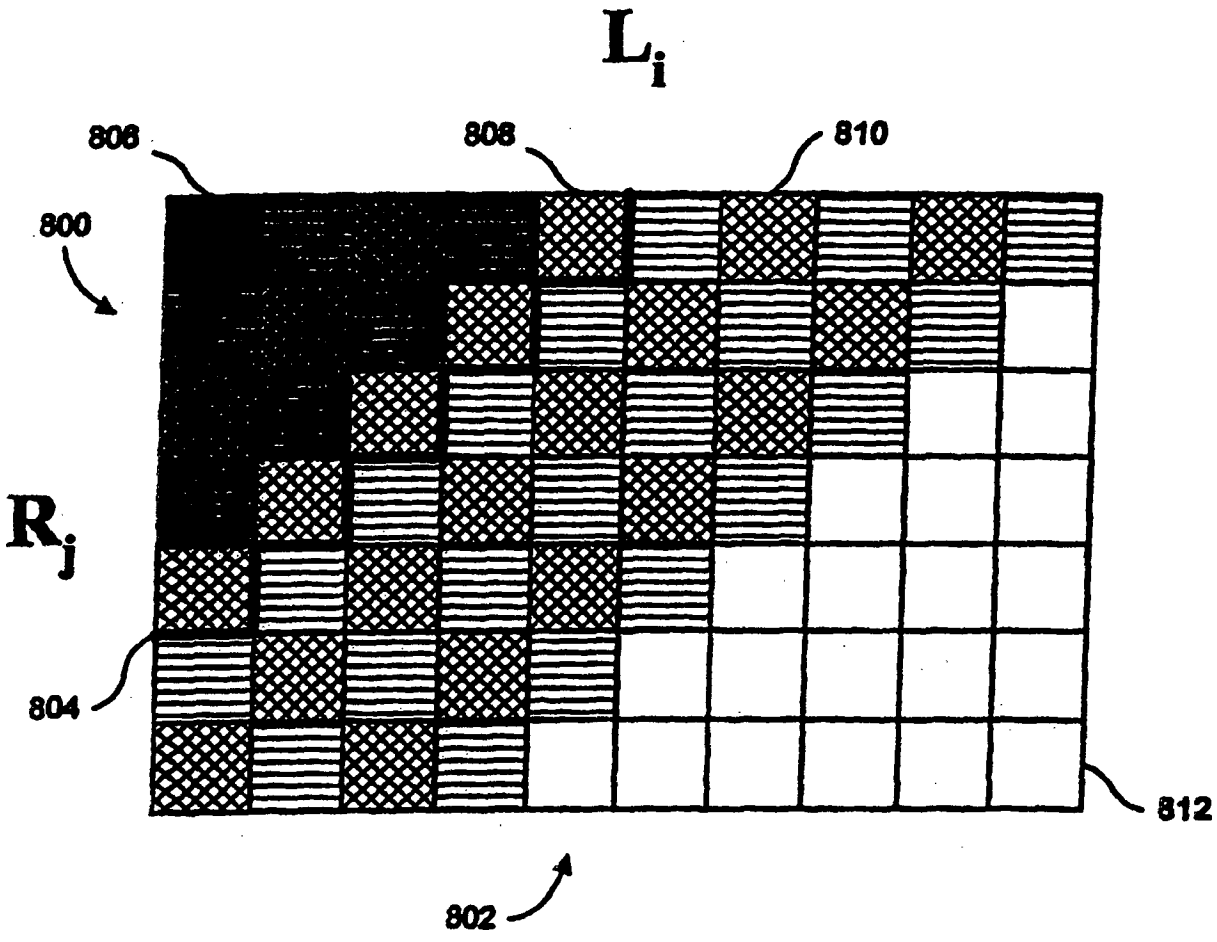
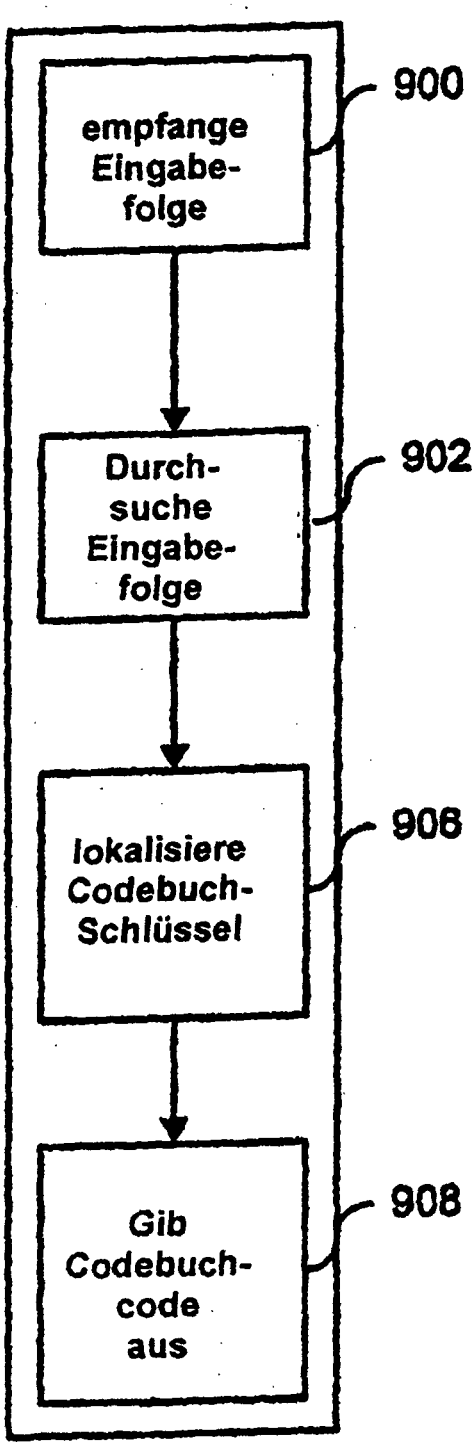


FIG. 14


empfange Eingabe-folge
900
Durch-suche Eingabe-folge
902
lokalisiere Codebuch-Schlüssel
906
Gib Codebuch-code aus
908