



(19)
Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) **DE 699 18 172 T2** 2009.10.01

(12) **Übersetzung der europäischen Patentschrift**

(97) **EP 1 142 129 B1**

(51) Int Cl.⁸: **H03M 7/42** (2006.01)

(21) Deutsches Aktenzeichen: **699 18 172.0**

(86) PCT-Aktenzeichen: **PCT/US99/29108**

(96) Europäisches Aktenzeichen: **99 966 062.4**

(87) PCT-Veröffentlichungs-Nr.: **WO 2000/036752**

(86) PCT-Anmeldetag: **07.12.1999**

(87) Veröffentlichungstag
der PCT-Anmeldung: **22.06.2000**

(97) Erstveröffentlichung durch das EPA: **10.10.2001**

(97) Veröffentlichungstag
der Patenterteilung beim EPA: **16.06.2004**

(47) Veröffentlichungstag im Patentblatt: **01.10.2009**

(30) Unionspriorität:
211294 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

(72) Erfinder:
**CHEN, Wei-ge, Issaquah, US; LEE, Ming-Chieh,
Bellevue, US**

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

(54) Bezeichnung: **ENTROPIEKODIERUNG VON VARIABLEN ZU VARIABLEN LÄNGE**

Anmerkung: Innerhalb von neun Monaten nach der Bekanntmachung des Hinweises auf die Erteilung des europäischen Patents kann jedermann beim Europäischen Patentamt gegen das erteilte europäische Patent Einspruch einlegen. Der Einspruch ist schriftlich einzureichen und zu begründen. Er gilt erst als eingelegt, wenn die Einspruchsgebühr entrichtet worden ist (Art. 99 (1) Europäisches Patentübereinkommen).

Die Übersetzung ist gemäß Artikel II § 3 Abs. 1 IntPatÜG 1991 vom Patentinhaber eingereicht worden. Sie wurde vom Deutschen Patent- und Markenamt inhaltlich nicht geprüft.

Beschreibung

[0001] Die Erfindung betrifft Datenkompression im Allgemeinen und eine Form von Entropiecodierung im Besonderen.

[0002] In einer typischen Audio-Codierungsumgebung werden Daten als eine lange Folge von Symbolen dargestellt, die in einen Coder eingegeben werden. Die eingegebenen Daten werden durch einen Coder codiert, über einen Kommunikationskanal übertragen (oder einfach gespeichert) und durch einen Decoder decodiert. Während des Codierens wird die Eingabe vorverarbeitet, abgetastet, umgewandelt, komprimiert oder anderweitig in eine Form zur Übertragung oder Speicherung verarbeitet. Nach der Übertragung oder Speicherung versucht der Decoder, die ursprüngliche Eingabe zu rekonstruieren.

[0003] Audio-Codierv Verfahren können gewöhnlich in zwei Klassen eingeteilt werden, nämlich Zeitbereichsverfahren und Frequenzbereichsverfahren. Zeitbereichsverfahren, z. B. ADPCM, LPC, arbeiten direkt im Zeitbereich, während Frequenzbereichsverfahren die Audiosignale in den Frequenzbereich transformieren, wo die eigentliche Kompression stattfindet. Die Frequenzbereichs-Codecs können weiter in entweder Unterband- oder Transformations-Coder getrennt werden, obwohl die Unterscheidung zwischen den beiden nicht immer klar ist. Die Verarbeitung eines Audiosignals im Frequenzbereich wird sowohl durch klassische Signalverarbeitungstheorien als auch menschliche Wahrnehmungsmodelle (z. B. Psychoakustik) begründet. Das Innenohr, speziell die Basilarmembran, verhält sich wie ein Spektralanalysator und transformiert das Audiosignal in Spektraldaten, bevor die weitere neurale Verarbeitung vonstatten geht.

[0004] Die Frequenzbereichs-Audiocodecs machen oft Gebrauch von vielen Arten von Gehörmaskierung, die mit dem menschlichen Hörsystem einhergehen, um das ursprüngliche Signal zu modifizieren und sehr viele Einzelheiten/Redundanzen zu beseitigen. Da die menschlichen Ohren nicht imstande sind, diese Modifikationen wahrzunehmen, wird eine effiziente Kompression erreicht. Maskierung wird gewöhnlich in Verbindung mit Quantisierung durchgeführt, sodass Quantisierungsrauschen bequem "verschleiert" werden kann. Bei modernen Audio-Codierv Verfahren werden quantisierte Spektraldaten gewöhnlich durch Anwenden von Entropiecodierung, z. B. Huffman-Codierung, weiter komprimiert.

[0005] Kompression ist erforderlich, weil es eine grundsätzliche Begrenzung des Kommunikationsmodells ist, dass Übertragungskanäle in der Regel eine begrenzte Kapazität oder Bandbreite besitzen. Folglich ist es häufig erforderlich, den Informationsgehalt der Eingangsdaten zu verringern, um ihnen zu erlau-

ben, zuverlässig, wenn überhaupt, über den Kommunikationskanal übertragen zu werden. Im Laufe der Zeit ist sehr viel Mühe bei der Entwicklung verlustloser und verlustbehafteter Kompressionsverfahren aufgewandt worden, um die Größe von Daten zum Übertragen oder Speichern zu reduzieren. Ein beliebtes verlustloses Verfahren ist Huffman-Codierung, die eine besondere Form von Entropiecodierung ist.

[0006] Entropiecodierung weist Codewörter verschiedenen Eingangsssequenzen zu und speichert alle Eingangssequenzen in einem Codebuch. Die Komplexität von Entropiecodierung hängt von der Zahl m von möglichen Werten ab, die eine Eingangssequenz X annehmen kann. Für ein kleines m gibt es wenig mögliche Eingangskombinationen, und das Codebuch für die Nachrichten kann sehr klein sein (z. B. sind nur ein paar Bits nötig, um alle möglichen Eingangssequenzen eindeutig darzustellen). Für digitale Anwendungen ist das Codealphabet höchstwahrscheinlich eine Serie von binären Ziffern (0, 1), und Codewortlängen werden in Bits gemessen.

[0007] Wenn bekannt ist, dass eine Eingabe aus Symbolen mit gleicher Wahrscheinlichkeit des Vorkommens besteht, ist es eine optimale Codierung, gleich lange Codewörter zu verwenden. Aber es ist nicht typisch, dass ein Eingangsstrom gleiche Wahrscheinlichkeit des Empfangens irgendeiner bestimmten Nachricht aufweist. In der Praxis sind gewisse Nachrichten wahrscheinlicher als andere, und Entropiecoder machen davon Gebrauch, um die mittlere Länge von Codewörtern unter erwarteten Eingaben zu minimieren. Herkömmlich werden jedoch Eingangssequenzen fester Länge Codes variabler Länge zugewiesen (oder umgekehrt, Sequenzen variabler Länge werden Codes fester Länge zugewiesen).

[0008] Internationale Veröffentlichung WO 98/40969 beschreibt ein System zum Komprimieren einer ASCII- oder ähnlich codierten Textdatei. Pipelining von bestimmten Datenkompressionsalgorithmen wird in Bailey und Mukkamala, "Pipelining Data Compression Algorithms", The Computer Journal, Vol. 33 Nr. 4 (August 1990) beschrieben. Ein adaptiver Algorithmus für verlustlose Kompression von digitalem Audio wird in Shamoon und Heegard, "A Rapidly Adaptive Lossless Compression Algorithm für High Fidelity Audio Coding", Proc. IEEE Data Compression Conf., 430-39 (1994) beschrieben. Ein Vergleich der H.261- und MPEG1-Videokompressionsstandards wird in von Roden, "H.261 und MPEG1 - A Comparison", Proc. IEEE 15. Ann. Int'l Conf. On Computers and Comm., 65-71 (1996) beschrieben.

[0009] Die Erfindung wird durch die Gegenstände der unabhängigen Ansprüche definiert.

[0010] Bevorzugte Ausführungen werden in den ab-

hängigen Ansprüchen definiert.

[0011] Die Erfindung betrifft die Verwendung eines variabel zu variabel Entropiecoders, um einen beliebigen Eingabestrom zu codieren. Ein variabel zu variabel Entropiecoder codiert Eingabesequenzen variabler Länge mit Codes variabler Länge. Um die Codebuchgröße zu begrenzen, können Entropietyp-Codes nur wahrscheinlichen Eingaben zugewiesen werden, und alternative Codes werden benutzt, um weniger wahrscheinliche Sequenzen zu identifizieren.

[0012] Um das Absuchen des Codebuches zu optimieren, kann es in Abschnitten organisiert werden, die getrennt abgesucht werden. Eine Anordnung ist z. B., alle gespeicherten Eingabesequenzen in dem Codebuch entsprechend dem ersten Symbol in der Eingangssequenz zu gruppieren. Eine Hash-Codierfunktion, Sammlung von Zeigern oder ein anderes Verfahren können benutzt werden, um sofort zu einem gegebenen Abschnitt des Codebuches zu springen. Jeder Abschnitt kann des Weiteren entsprechend der mit dem Eintrag verbundenen Wahrscheinlichkeit sortiert werden. Zum Beispiel kann jeder Abschnitt so sortiert sein, dass höchstwahrscheinliche Eingaben in dem Abschnitt zuerst stehen, um so die Wahrscheinlichkeit zu erhöhen, dass eine Übereinstimmung schnell gefunden wird.

[0013] Das Abgleichen von Codebucheinträgen hängt von der internen Darstellung des Buches ab. In einer Baumstruktur können z. B. Knoten jedes Zeichen der Eingabe darstellen, sodass das Erreichen eines Blattes das Ende und die Identifikation einer bestimmten Gruppierung von Eingabesymbolen bedeutet. In einer Tabellenstruktur kann ein Musterabgleichalgorithmus auf jeden Tabelleneintrag in dem geeigneten Abschnitt angewandt werden. Abhängig von der Implementierung der Tabelle und der Abgleichalgorithmen kann das Suchen durch die Erkenntnis erleichtert werden, dass nur so viele Eingabesymbole wie die längste Gruppierung in dem Codebuchabschnitt in Betracht gezogen werden müssen. Nach Finden einer Codebuchübereinstimmung kann der entsprechende Entropietyp-Code ausgegeben und das Suchen mit dem nächsten Symbol, das der abgeglichenen Eingabe folgt, wiederholt werden.

[0014] Die veranschaulichten Ausführungen konzentrieren sich auf das Codieren von Audiodaten, und von dem Eingangsstrom wird erwartet, dass er irgendein Datenstrom ist, z. B. Zahlen, Zeichen oder Binärdaten, die Audio codieren. Zur Einfachheit wird der Eingangsstrom hierin als eine Serie von Symbolen bezeichnet, wo sich jedes "Symbol" auf die geeignete Maßeinheit für die einzelne Eingabe bezieht. Der Eingangsstrom kann von einem lokalen Speicher, von Intranetzen, dem Internet oder von Streaming-Daten herrühren (z. B. Microsoft's "NET-SHOW"™ Client/Server-Streaming-Architek-

tur).

[0015] [Fig. 1](#) ist ein Blockschaltbild eines Computersystems, das benutzt werden kann, variabel zu variabel Entropiecodierung zu implementieren.

[0016] [Fig. 2](#) zeigt ein Grundkommunikationsmodell zum Übertragen von Streaming- und Nicht-Streaming-Daten.

[0017] [Fig. 3](#) ist ein Flussdiagramm, das das Erzeugen eines Codebuches mit variabel langen Einträgen für variabel lange Symbolgruppierungen zeigt.

[0018] [Fig. 4–Fig. 10](#) veranschaulichen das Erzeugen eines Codebuches nach [Fig. 3](#) für ein Alphabet (A, B, C).

[0019] [Fig. 11](#) zeigt das Codieren von Audiodaten.

[0020] [Fig. 12](#) veranschaulicht einen Entropiecoder.

[0021] Die Erfindung ist in einem audiovisuellen Codec (Kompressor/Dekompressor) implementiert worden. Dies ist nur ein Beispiel wie die Erfindung implementiert werden kann. Die Erfindung ist gedacht, immer dort verwendet zu werden, wo Entropiecodierung benutzt werden kann, und ist auf das Komprimieren jeder Art von Daten anwendbar. Kurz beschrieben, optimale Entropiecodierung erfordert übermäßige Ressourcen, und die veranschaulichten Ausführungen liefern eine fast optimale Codierungslösung, die weit weniger Ressourcen erfordert.

Exemplarische Betriebsumgebung

[0022] [Fig. 1](#) und die folgende Erörterung sind gedacht, eine kurze, allgemeine Beschreibung einer geeigneten Rechenumgebung zu geben, in der die Erfindung implementiert werden kann. Während die Erfindung in dem allgemeinen Kontext von computer-ausführbaren Anweisungen eines Computerprogramms, das auf einem Personal Computer läuft, beschrieben wird, werden die Fachleute in der Technik anerkennen, dass die Erfindung auch in Kombination mit anderen Programmmodulen implementiert werden kann. Programmmodule umfassen im Allgemeinen Routinen, Programme, Komponenten, Datenstrukturen usw., die bestimmte Aufgaben durchführen oder bestimmte abstrakte Datentypen implementieren. Weiterhin werden die Fachleute in der Technik anerkennen, dass die Erfindung mit anderen Computersystemkonfigurationen praktiziert werden kann, die handgehaltene Geräte, Multiprozessorsysteme, Mikroprozessor-basierte oder programmierbare Verbraucherelektronik, Minicomputer, Mainframe-Computer und dergleichen einschließen. Die veranschaulichte Ausführung der Erfindung kann auch in verteilten Rechenumgebungen praktiziert werden, wo Tasks durch entfernte Verarbeitungseinrichtungen,

die durch ein Kommunikationsnetz verbunden sind, durchgeführt werden. Aber einige Ausführungen der Erfindung können auf allein stehenden Computern praktiziert werden. In einer verteilten Rechenumgebung können sich Programmmodule sowohl in lokalen als auch in entfernten Speichereinrichtungen befinden.

[0023] Mit Veweis auf [Fig. 1](#) umfasst ein exemplarisches System zur Implementierung der Erfindung einen Computer **20**, einschließlich einer Verarbeitungseinheit **21**, eines Systemspeichers **22** und eines Systembusses **23**, der verschiedene Systemkomponenten, einschließlich des Systemspeichers, mit der Verarbeitungseinheit **21** verbindet. Die Verarbeitungseinheit **21** kann jeder von verschiedenen käuflichen Prozessoren sein, einschließlich Intel x86, Pentium und kompatiblen Mikroprozessoren von Intel und anderen, des Alpha-Prozessors von Digital und des PowerPC von IBM und Motorola. Zweifach-Mikroprozessoren und andere Multiprozessorarchitekturen können auch als die Verarbeitungseinheit **21** verwendet werden.

[0024] Der Systembus kann jeder von verschiedenen Typen von Busstrukturen sein, einschließlich eines Speicherbusses oder Speicherkontrollers, eines Peripheriebusses und eines Lokalbusses, der eine Vielfalt von herkömmlichen Busarchitekturen wie PCI, AGP, VESA, Microchannel, ISA und EISA, um nur ein paar zu nennen, verwendet. Der Systemspeicher enthält einen Nurlesespeicher (ROM) **24** und einen Direktzugriffsspeicher (RAM) **25**. Im ROM **24** ist ein Basis-Eingabe/Ausgabesystem (BIOS) **26** gespeichert, das die Basisroutinen enthält, die helfen, Information zwischen Elementen in dem Personal Computer **20** zu übertragen, z. B. während des Hochlaufens.

[0025] Der Personal Computer **20** enthält weiter ein Festplattenlaufwerk **27**, ein Magnetplattenlaufwerk **28**, z. B. zum Lesen und Beschreiben einer wechselbaren Platte **29**, und ein optisches Plattenlaufwerk **30**, z. B. zum Lesen einer CD-ROM Platte **31**, oder zum Lesen oder Beschreiben von anderen optischen Medien. Das Festplattenlaufwerk **27**, das Magnetplattenlaufwerk **28** und das optische Plattenlaufwerk **30** sind mit dem Systembus **23** über eine Festplattenschnittstelle **32**, eine Magnetplattenschnittstelle **33** und eine optische Plattenschnittstelle **34** verbunden. Die Laufwerke und ihre zugehörigen computerlesbaren Medien liefern nicht flüchtige Speicherung von Daten, Datenstrukturen, computerausführbaren Anweisungen usw. für den Computer **20**. Obwohl die Beschreibung von computerlesbaren Medien oben sich auf eine Festplatte, eine wechselbare Magnetplatte und eine CD bezieht, werden die Fachleute erkennen, dass andere Arten von computerlesbaren Medien, z. B. Magnetkassetten, Flash-Speicherkarten, digitale Videoplatten, Bernoulli-Kassetten und

Magnetkassetten, Flash-Speicherkarten, digitale Videoplatten, Bernoulli-Kassetten und dergleichen, ebenfalls in der exemplarischen Betriebsumgebung verwendet werden können.

[0026] Eine Anzahl von Programmmodulen kann in den Laufwerken und dem RAM **25** gespeichert werden, einschließlich eines Betriebssystems **35**, einem oder mehr Anwendungsprogrammen (z. B. Internet-Browser-Software) **36**, anderen Programmmodulen **37** und Programmdateien **38**.

[0027] Ein Benutzer kann Befehle und Information über eine Tastatur **40** und eine Zeigegerät, z. B. eine Maus **42**, in den Computer **20** eingeben. Andere Eingabeeinrichtungen (nicht gezeigt) können ein Mikrofon, einen Joystick, ein Gamepad, eine Satellitenschüssel einen Scanner oder dergleichen einschließen. Diese und andere Eingabeeinrichtungen werden oft mit der Verarbeitungseinheit **21** durch eine Serialport-Schnittstelle **46** verbunden, die mit dem Systembus verbunden ist, können aber durch andere Schnittstellen, z. B. einen Parallelport, einen Gameport oder einen Universal-Serienbus (USB) verbunden werden. Ein Monitor **47** oder eine andere Art von Anzeigeeinrichtung sind ebenfalls mit dem Systembus **23** über eine Schnittstelle, z. B. einen Videoadapter **48**, verbunden. Zusätzlich zu dem Monitor umfassen Personal Computer typischerweise andere periphere Ausgabeeinrichtungen (nicht gezeigt), z. B. Lautsprecher und Drucker.

[0028] Der Computer **20** soll unter Verwendung von logischen Verbindungen zu einem oder mehr entfernten Computern, z. B. ein entfernter Computer **49**, in einer vernetzten Umgebung arbeiten. Obwohl in [Fig. 1](#) nur eine Speichereinrichtung **50** gezeigt wurde, kann der entfernte Computer **49** ein Web-Server, Router, eine Peer-Einrichtung oder ein anderer gemeinsamer Netzknoten sein und typischerweise viele oder alle bezüglich des Computers **20** beschriebenen Elemente enthalten. Der Computer **20** kann den entfernten Computer **49** über eine durch eine Gateway **55** (z. B. ein Router, eine überlassene Leitung oder andere Netzwerkverbindung) hergestellte Internet-Verbindung, eine Modem- **54** Verbindung oder durch ein lokales bürinternes Netzwerk (LAN) **51** oder ein Weitbereichs-Netzwerk (WAN) **52** kontaktieren. Es ist einzusehen, dass die gezeigten Netzwerkverbindungen exemplarisch sind und andere Mittel zum Herstellen einer Kommunikationsverbindung zwischen Computern benutzt werden können.

[0029] Entsprechend den Gepflogenheiten von in der Technik der Computerprogrammierung erfahrenen Personen wird die vorliegende Erfindung, sofern nicht anders angegeben, unten mit Verweis auf Akte und symbolische Darstellungen von Operationen, die durch den Computer **20** durchgeführt werden, be-

schrieben. Solche Akte und Operationen werden manchmal als computerausgeführt bezeichnet. Man wird einsehen, dass die Akte und symbolisch dargestellten Operationen die Manipulation durch die Verarbeitungseinheit **21** von elektrischen Signalen, die Datenbits darstellen, was eine resultierende Transformation oder Reduktion der elektrischen Signaldarstellung bewirkt, und die Unterhaltung von Datenbits in Speicherstellen des Speichersystems (einschließlich des Systemspeichers **22**, der Festplatte **27**, den Floppydisks **29** und der CD-ROM **31**), um dadurch den Betrieb des Computersystems zu rekonfigurieren oder anderweitig zu ändern, sowie andere Verarbeitung von Signalen umfassen. Die Speicherstellen, wo Datenbits unterhalten werden, sind physikalische Stellen, die entsprechend den Datenbits bestimmte elektrische, magnetische oder optische Eigenschaften aufweisen.

[0030] [Fig. 2](#) zeigt ein Grundkommunikationsmodell. In einem Grundkommunikationsmodell gibt es eine Datenquelle oder Sender **200**, einen Kommunikationskanal **204** und einen Datenempfänger **208**. Die Quelle kann jemand sein, der auf einem Telefon über Telefonleitungen mit einer anderen Person spricht. Oder die Quelle kann eine durch drahtlose Verfahren an Fernsehgeräte oder Radioempfänger gesendete Fernseh- oder Radioübertragung sein. Oder die Quelle kann eine digitale Codierung von Audiodaten sein, die über eine verdrahtete oder drahtlose Strecke (z. B. ein LAN oder das Internet) an einen entsprechenden Decoder für die Information gesendet werden.

[0031] Um das Senden und Empfangen der Daten zu ermöglichen, wird ein Coder **202** benutzt, um die Datenquelle auf das Senden über den Kommunikationskanal **204** vorzubereiten. Der Coder ist zuständig für das Umwandeln der Quelldaten in ein für den Kanal **204** geeignetes Format. Zum Beispiel wird im Kontext eines gewöhnlichen Telefonanrufs jemandens Stimme durch den Telefonhörer von Stimmklängen in Analogimpulse umgewandelt, die als analoge Daten an lokale Telefonempfangsanlagen gesendet werden. Dieses Analogsignal wird dann in digitale Form umgesetzt, mit zahlreichen anderen ähnlich codierten Unterhaltungen gemultiplext und über eine gemeinsame Leitung an den Empfänger übertragen. In [Fig. 2](#) entspricht der Kanal **204** zum großen Teil einem gemeinsamen Pfad, den sich vielfache Sender und Empfänger teilen. Für Netzwerkanwendungen ist der Kanal **204** gewöhnlich ein Intranet oder das Internet. Am empfangenden Ende **208** wird ein Decoder **206** benötigt, um den Codierprozess umzukehren, um so dem Empfänger sensible Daten zu präsentieren.

[0032] Dieses einfache Modell berücksichtigt jedoch nicht die wirklichen Forderungen von Anwendungsprogrammen. Ein Client (z. B. ein Anwen-

dungsprogramm) möchte empfangene Daten in Echtzeit verarbeiten, anzeigen oder abspielen, sobald sie über eine Netzwerkstrecke rückgewonnen sind. Zu diesem Zweck wird ein Streaming-Liefersystem benötigt, d. h., ein adaptives Datenübertragungssystem, das eine Anwendungsstufen-Bandbreitenreservierung für einen Datenstrom erlaubt. Streaming-Umgebungen stehen im Gegensatz zu herkömmlichen Netzwerk-Programmen, z. B. bestimmten Versionen von Internet-Browsern, die den Inhalt von Web-Seiten auf einer nicht bevorrechtigten Basis herunterladen und es erlauben, die Dateninhaltslieferung über die Netzwerkverbindung **204** für bestimmte Rückgewinnungsbedürfnisse (z. B. eine Verbindung mit langsamer Einwahl) zu instrumentieren (und zu optimieren).

[0033] Ein exemplarisches Streaming-Format (SF) ist das Microsoft 'Active Streaming Format'. Im Allgemeinen definiert SF die Struktur von komplexen, synchronisierten Objektdatenströmen. Jedes Objekt kann in einen SF-Datenstrom gelegt werden, einschließlich Audio- und Videodatenobjekten, Skripts, ActiveX-Steuerungen und HTML-Dokumenten. SF-Daten können in einer Datei gespeichert oder in Echtzeit unter Verwendung von Audio- und Video-Erfassungshardware erzeugt werden. Eine Anwendungsprogrammierungsschnittstelle (API), die einer Implementierung des SF entspricht, kann eine Anwendung mit Unterstützung zum Liefern und Empfangen von Streaming-Inhalt bereitstellen. Eine solche API ist der Microsoft Audio Compression Manager (ACM), der Funktionen zur Verarbeitung (Komprimieren und Liefern) von Audiodaten bereitstellt. Andere Netzwerk-APIs, die benutzt werden können, um das SF zu unterstützen, schließen die Microsoft Win32 Internet Extensions (WinInet), Win-Sock und TCP/IP APIs ein. (Zu weiterer Information siehe die 1998 Visual Studio 6.0 MSDN Library). Man beachte, dass beabsichtigt ist, dass verarbeitete Daten zur späteren Rückgewinnung durch einen Client gespeichert werden können, und dass solches Rückgewinnen in einem Nicht-Streaming-Format (z. B. durch ein kleines Abspielgerät) durchgeführt werden kann.

[0034] Um Streaming- oder Nicht-Streaming-Daten zu übertragen, wandeln Netze wie das Internet die Quelldaten in eine für das Netz geeignete Paketform um. Pakete enthalten gewöhnlich Routing-Information sowie die tatsächlichen Daten. SF-Datenströme bestehen vorzugsweise aus Datenpaketen, die über jedes digitale Netz übertragen werden können, indem sie eines zu einer Zeit in das Datenfeld von Netzwerkpaketen eingefügt werden. Jedes SF-Paket kann einen priorisierten Mix aus Daten von verschiedenen Objekten innerhalb des Stromes enthalten, sodass die Bandbreite auf höher priorisierte Objekte konzentriert (oder organisiert, um Durchsatz zu optimieren) werden kann. Diese Daten können in Echtzeit erfasst, im nicht flüchtigen Speicher gespeichert,

von bestehenden Audio- oder Videoformaten umgewandelt, durch Kombinieren von Audio mit Bildern und Skripts erzeugt oder über das Netzwerk an ein Client Programm oder einen Betrachter geliefert werden. Der Client-Empfänger **208** der Streaming-Daten kann eine herkömmliche "Helfer"-Anwendung (für Kompatibilität mit der alten Web-Veröffentlichungslösung) oder eine modernere in einer Webseite eingebettete Webseitensteuerung (z. B. ein ActiveX-Objekt) sein.

[0035] SF Datenströme werden von dem herkömmlichen Netzwerkinhalt in Echtzeit unterschieden, wenn ein Client sie empfängt. Das Abspielen des geströmten Inhalts wird leider anfällig für Übertragungsverzögerungen. Wenn Daten nicht zuverlässig eintreffen, oder wenn die Übertragungsgeschwindigkeit unter ein annehmbares Minimum fällt, kann das Abspielen des Inhalts nicht mit einer annehmbaren Rate weitergehen. Glatte strömendes Abspielen bei einem Client verlangt, dass die Übertragung eine Bandbreite kleiner als die verfügbare Bandbreite des Client benötigt (z. B. die Geschwindigkeit der Strecke **204** minus Netzarbeits-Overhead). Eine Wählverbindung zu dem Internet stellt typisch eine Bandbreite von 28–34 kpps bereit. Audiovisuelle Quelldaten (die bandbreitenintensiv sind) müssen folglich erheblich komprimiert werden, um ihre Übertragung über Verbindungen mit niedriger Bitrate zu erlauben. Der Grad an Kompression beeinflusst notwendigerweise die Qualität des wiedergegebenen Signals. Ein Server stellt vorzugsweise mehrfache für verschiedene Netzwerk-Geschwindigkeiten optimierte Quellen bereit oder benutzt ein adaptives Rückkopplungssystem, um eine Echtzeit-Analyse des tatsächlichen Durchsatzes des Client durchzuführen.

[0036] Sobald SF-Datenpakete codiert **202** und in Netzwerkpakete gelegt und über das Netzwerk **204** gesendet sind, sorgt die Routing-Technologie des Netzwerks für das Liefern der Netzwerkpakete an den Empfänger **208**. Von dem Rundfunksender **200** werden vorzugsweise eine Vielfalt von Netzwerk- und Anwendungsprotokollen, z. B. UDP, TCP, RTP, IP, Multicast, IPX und HRRP, unterstützt.

[0037] Wie oben erörtert ist die Bandbreite begrenzt, und der Coder **202** muss gewöhnlich Daten vor dem Übertragen komprimieren. Ein besonders wirkungsvolles Verfahren zur Codierung von Quelldaten-Frequenzkoeffizienten, um zuverlässige Übertragung über einen Kommunikationskanal sicherzustellen, ist Entropiecodierung. Entropiecodierung zieht Nutzen aus Datenkohärenz und ist wirkungsvoll, wenn Symbole eine ungleichmäßige Wahrscheinlichkeitsverteilung aufweisen.

[0038] [Fig. 3](#) ist ein Flussdiagramm, das ein bevorzugtes Verfahren zum Erzeugen des Codebuches eines Entropiecoders zeigt. Im Gegensatz zu Verfah-

ren des Standes der Technik veranschaulicht [Fig. 3](#) insbesondere, wie ein Codebuch mit variabel langen Codezuweisungen für variabel lange Symbolgruppierungen zu erzeugen ist. Wie oben erörtert benötigen Verfahren des Standes der Technik entweder Codes fester Länge oder feste Blöcke von Eingabe. Bevorzugte Implementierungen überwinden die Ressourcen-Anforderungen von Großdimensions-Vektorcodierung und die Nichtanwendbarkeit von Codierung auf Wörter gleicher Längen, indem ein Entropie-basierter variabel zu variabel Code bereitgestellt wird, wo Codewörter variabler Länge benutzt werden, um X-Sequenzen variabler Länge zu codieren.

[0039] Es sei angenommen, dass y_i jede Quellensymbolgruppe $\{x_j\}$, für $1 \leq j \leq N_i$ mit der Wahrscheinlichkeit P_i des Vorkommens innerhalb des Eingangsstromes ([Fig. 2](#) Kanal **204**) darstellt, und dass jeder Gruppe ein entsprechendes Codewort mit L_i Bits zugewiesen wird. Es wird davon ausgegangen, dass x_i aus einer festen Alphabet vorbestimmter Größe genommen wird. Die Aufgabe ist, die Gleichung

$$L = \sum_i (L_i \cdot P_i) / N_i$$

zu minimieren.

[0040] Anstatt eine allgemeine Lösung für das Problem zu finden, wird das Problem in zwei verschiedene Aufgaben geteilt. Die erste Aufgabe ist das Identifizieren einer (sub-optimalen) Gruppierung eines Satzes von Eingangssymbolen $\{x_i\}$ durch eine unten beschriebene Weise. Die zweite Aufgabe ist das Zuweisen eines Entropiecodes für die gruppierten Symbole $\{y_i\}$. Man beachte, dass bekannt ist, dass, wenn die Quelle nicht kohärent ist (d. h. der Eingang unabhängig oder ohne Gedächtnis ist), jede Gruppierung, die die gleiche Konfiguration von $\{N_j\}$ hat, die gleiche Codiereffizienz erzielen kann. Bei dieser Situation wird die erste Aufgabe belanglos.

[0041] Um die erste Aufgabe durchzuführen, wird eine anfängliche triviale Symbolgruppierung **300**, z. B. $\{y_i\} = \{x_i\}$ vorbereitet. Diese anfängliche Konfiguration nimmt an, dass ein exemplarischer Eingangstrom benutzt wird, um das Erzeugen des Codebuches zu üben. Es versteht sich, dass ein Computer mit Software-Konstruktionen, z. B. Datenstrukturen, programmiert werden kann, um den Empfang jedes Symbols von einer Eingabe zu verfolgen. Solche Datenstrukturen können als eine Binärtyp-Baumstruktur, Hash-Tabelle oder eine Kombination der zwei implementiert werden. Andere gleichwertige Strukturen können ebenfalls verwendet werden.

[0042] Nach Bestimmen der trivialen Gruppierung wird die Wahrscheinlichkeit des Vorkommens jedes y_i in **302** berechnet. Eine solche Wahrscheinlichkeit wird in Bezug auf irgendeine exemplarische Eingabe

bestimmt, die benutzt wird, um die Codebucherzeugung zu üben. Sowie weitere Symbole zu der Symboldatenstruktur hinzugefügt werden, werden die Wahrscheinlichkeiten dynamisch justiert.

[0043] Als Nächstes wird die meist wahrscheinliche Gruppierung y_i (bezeichnet mit ymp) in **304** identifiziert. Wenn in **306** das Symbol mit der höchsten Wahrscheinlichkeit eine Gruppierung von früheren Symbolen mit niedrigerer Wahrscheinlichkeit ist, wird in **308** die Gruppierung in ihre Bestandteilsymbole zerlegt, und die Verarbeitung wird von Schritt **302** neu begonnen. (Obwohl Symbole kombiniert werden können, bewahrt die Gruppe die Erinnerung von allen Symbolen darin, sodass Symbole extrahiert werden können.)

[0044] Wenn das Symbol keine Gruppierung ist, geht die Verarbeitung bei Schritt **310** weiter, wo die meist wahrscheinliche Gruppierung dann vorläufig mit einzelnen Symbolerweiterungen x_i 's erweitert **310** wird. Vorzugsweise wird ymp mit jedem Symbol aus dem verwendeten X-Alphabet erweitert. Jedoch kann ein Prädiktor verwendet werden, um nur einen Erweiterungssatz zu erzeugen, der nur wahrscheinliche Erweiterungen enthält, wenn das Alphabet sehr groß ist, und bekannt ist, dass viele Erweiterungen unwahrscheinlich sind. Zum Beispiel kann ein Prädiktor auf semantischer oder zusammenhanglicher Bedeutung basiert sein, sodass sehr unwahrscheinliche Erweiterungen von vornherein ignoriert werden können.

[0045] Die Wahrscheinlichkeit für jede vorläufige Erweiterung von ymp wird dann berechnet **312**, und nur die meist wahrscheinliche Erweiterung wird bewahrt **314**. Der Rest der weniger wahrscheinlichen Erweiterungen werden als eine kombinierte Gruppierung zusammenfallen gelassen **316** und im Codebuch mit einem speziellen Symbol gespeichert, um eine kombinierte Gruppierung anzuzeigen. Dieses Wild-Card-Symbol stellt jede beliebige Symbolgruppierung mit ymp als ein Präfix dar, aber mit einer Erweiterung (Suffix), die sich von der meist wahrscheinlichen Erweiterung unterscheidet. Das heißt, wenn $ymp + xmp$ die meist wahrscheinliche Wurzel und Erweiterung ist, werden die anderen weniger wahrscheinlichen Erweiterungen als $ymp^*, * <> xmp$ dargestellt. (Man beachte, dass diese Erörterung zur Klarheit von serieller Verarbeitung von Einsymbol-Erweiterungen ausgeht; jedoch wird parallele Verarbeitung von Mehrsymbol-Erweiterungen erwogen.)

[0046] Eine in der Technik erfahrene Person wird verstehen, dass das Anwenden von Einsymbol-Erweiterungen und das Bewahren nur der meist wahrscheinlichen Gruppierung zur Klarheit der Erörterung auferlegte Einschränkungen sind. Weiter ist zu verstehen, dass, obwohl sich die Erörterung auf serielle Verarbeitung konzentriert, die Codebuch-Konstruktion parallel durchgeführt werden kann.

[0047] Doe Codebuch-Konstruktion wird vollendet, indem die Schritte **302–326** wiederholt **318** werden, bis alle Erweiterungen gemacht sind oder die Zahl der Codebucheinträge eine vorbestimmte Grenze erreicht. Das heißt, das Wiederholen der Berechnung von Wahrscheinlichkeiten für jedes laufende y_i **302**, wo der Codebuchsatz $\{Y\}$ nun $ymp + xmp$ enthält, und jeweils Wählen **304** und Gruppieren von meist und geringst wahrscheinlichen Erweiterungen. Die Wirkung des wiederholten Anwendens der obigen Operationen ist, automatisch Symbolgruppierungen mit hoher Korrelation zu sammeln, sodass Zwischengruppen-Korrelation minimiert wird. Dies minimiert den Zähler von L , während gleichzeitig die Länge des meist wahrscheinlichen y_i maximiert wird, sodass der Nenner von L maximiert wird.

[0048] [Fig. 4–Fig. 10](#) veranschaulichen ein Codebuch gemäß [Fig. 3](#) für ein Alphabet (A, B, C). Für diese Erörterung wird das Codebuch in Bezug auf einen exemplarischen Eingangsstrom "A A A B B A A C A B A B B A B" definiert. Wie oben erörtert können ein oder mehr exemplarische Eingänge benutzt werden, um ein Codebuch zu erzeugen, das dann von Codern und Decodern verwendet wird, um beliebige Eingänge zu verarbeiten. Zur Klarheit, das Buch wird als eine Baumstruktur präsentiert, obwohl es tatsächlich als eine lineare Tabelle, Hash-Tabelle, Datenbank usw. implementiert werden kann. Wie gezeigt ist der Baum von links nach rechts orientiert, wo die linke Spalte (z. B. "A" und "X0") die oberste Reihe einer Baumtyp-Struktur darstellt, und nacheinander gezahnte Reihen die "Kinder" des Knotens der vorangehenden Reihe darstellen (z. B. in einem Oben-Unten-Baum für [Fig. 5](#) ist "A" ein Elternknoten der ersten Reihe für einen Mittelkindknoten der zweiten Reihe "B").

[0049] Bei der Herstellung des Codebuches ist die allgemeine Regel, den meist wahrscheinlichen Blattknoten auszusuchen, ihn zu erweitern, Wahrscheinlichkeiten neu zu berechnen, um den meist wahrscheinlichen Blattknoten zu bestimmen, und dann die übrigen Geschwisterknoten in einen einzigen X_n -Knoten ($n = 0 \dots N$, verfolgen jedes Mal, wenn Knoten kombiniert wurden) zu verdichten. Wenn sich ergibt, dass der meist wahrscheinliche Knoten ein Gruppenknoten ist, werden die Gruppe geteilt, Wahrscheinlichkeiten neu berechnet und der meist wahrscheinliche Mitglieds-knoten bewahrt (d. h., die restlichen Gruppenmitglieder werden neu gruppiert). Die Verarbeitung wiederholt sich, bis ein Stopp-Zustand erreicht ist, z. B. ein Codebuch mit vorbestimmter Größe.

[0050] [Fig. 4](#) zeigt eine anfängliche Gruppierung für den Eingangsstrom "A A A B B A A C A B A B B A B". Ein anfängliches Zerlegen des Eingangs zeigt Wahrscheinlichkeiten des Vorkommens von $A = 8/15$, $B = 6/15$ und $C = 1/15$. Diese anfängliche triviale Grup-

pierung kann auf der Basis verschiedener Kriterien erzeugt werden, wobei das einfachste ist, einen Knoten erster Stufe für jedes Zeichen in dem Alphabet zu haben. Wenn jedoch das eingegebene Alphabet groß ist, kann die triviale Gruppe auf eine Untermenge von Symbolen mit höchster Wahrscheinlichkeit begrenzt werden, wo die restlichen Symbole in eine X-Gruppierung kombiniert werden. [Fig. 4](#) veranschaulicht diese Verfahren durch Beginnen mit nur zwei Anfangsgruppen, Gruppe A **400** mit Wahrscheinlichkeit 8/15 und Gruppe X0 **402** mit Wahrscheinlichkeit 7/15, wo X0 alle verbleibenden Symbole niedriger Wahrscheinlichkeit in dem Alphabet, z. B. B und C, darstellt.

[0051] Nach Herstellen einer anfänglichen trivialen Gruppierung wird der Blattknoten mit höchster Wahrscheinlichkeit zur Erweiterung ausgewählt (siehe auch [Fig. 3](#) Erörterung bezüglich Verarbeitungsfolge). Wie in [Fig. 5](#) gezeigt wird daher Gruppe A **400** vorläufig durch jedes Zeichen im Alphabet erweitert (oder man kann die Erweiterung auf eine Untermenge davon begrenzen, wie für das Erzeugen der Anfangsgruppierung beschrieben). Wahrscheinlichkeiten werden dann in Bezug auf den Eingangsstrom "A A A B B A A C A B A B B A B" neu berechnet, um Werte für die vorläufigen Erweiterungen A **406**, B **408** und C **410** zu bestimmen. Das Ergebnis sind neun Teilungsgruppen, wo "A A" 2/9 erscheint, "A B" 4/9 erscheint und "A C" 0/9 erscheint. Die meist wahrscheinliche Erweiterung "A B" wird daher beibehalten, und die anderen Erweiterungen werden in X1 = A, C zusammenfallen gelassen. Man beachte, dass, obwohl diese Erörterung wiederholt alle Wahrscheinlichkeiten neu berechnet, es ein effizienterer Weg ist, Wahrscheinlichkeiten und Verbindungen für jeden Knoten innerhalb des Knotens zu bewahren, und nur Information, wenn nötig, zu berechnen.

[0052] [Fig. 6](#) zeigt das Zusammenfallen in X1 für [Fig. 5](#). Die Verarbeitung wiederholt sich mit dem Identifizieren des Knotens mit höchster Wahrscheinlichkeit, z. B. Knoten B **408** mit Wahrscheinlichkeit 4/9.

[0053] Wie in [Fig. 7](#) gezeigt wird dieser Knoten **408** vorläufig mit Symbolen A **414**, B **416**, C **418** erweitert, und, wie oben erörtert, wird die vorläufige Gruppierung mit höchster Wahrscheinlichkeit bewahrt. Nach Neuberechnen der Wahrscheinlichkeiten sind das Ergebnis acht Teilungsgruppen, in denen die Symbolfolge "A B A" **414** einmal erscheint, "A B B" **416** einmal erscheint und "A B C" **418** überhaupt nicht erscheint. Da die vorläufigen Erweiterungen A **414** und B **416** die gleiche Wahrscheinlichkeit des Vorkommens haben, muss eine Regel definiert werden, um zu wählen, welches Symbol zu bewahren ist. Für diese Erörterung wird, wann immer es gleiche Wahrscheinlichkeiten gibt, der Knoten der höchsten Reihe (z. B. der Kindknoten ganz links in einem Oben-Un-

ten-Baum) bewahrt. Desgleichen wird, wenn es einen Konflikt zwischen Baumreihen gibt, der Knoten der Reihe ganz links (z. B. der Knoten nächst der Wurzel eines Oben-Unten-Baumes) bewahrt.

[0054] Man beachte, dass das oben beschriebene Zerlegen des exemplarischen Eingangs für die nachlaufenden zwei Symbole "A B" des Eingangs nicht einsteht. Wie in [Fig. 7](#) gezeigt, gibt es kein Blatt, das "A B" entspricht, da diese Konfiguration in "A B A", "A B B" und "A B C" erweitert wurde. Zum Ausgleich können Codebucheinträge erzeugt werden, um für ein solches Ende von Eingangsssequenzen einzustehen, oder dem Eingang, der keinen Eintrag hat, kann mit einem Sonderzeichen entkommen werden, das in den codierten Ausgangsstrom eingefügt wird. Zum Beispiel kann ein Sondersymbol benutzt werden, um das Ende der Eingabe anzuzeigen, um dadurch zu verstehen zu geben, wie nachlaufende Zeichen beim Decodieren zu handhaben sind.

[0055] Deshalb wird, wie in [Fig. 8](#) gezeigt, der Knoten A **414** bewahrt, und die Knoten B **416** und C **418** werden in einen Knoten X2 = B, C **420** mit kombinierter Wahrscheinlichkeit von 1/8 + 0/8 kombiniert. Der nächste Schritt ist nun, den Knoten mit der gegenwärtig höchsten Wahrscheinlichkeit in Bezug auf den Eingangsstrom zu erweitern. Wie gezeigt haben Knoten X1 = A, C **412** und X0 = B, C **402** die gleiche Wahrscheinlichkeit des Vorkommens (3/8). Wie oben erörtert wird der höchste Knoten in dem Baum (X0 **402**) erweitert. (Obwohl es nur erforderlich ist, konsistent zu sein, ist es vorzuziehen, Knoten höherer Stufe zu erweitern, da dies die Effizienz der Codierung durch Erhöhen der Zahl von langen Codewörtern erhöhen kann.)

[0056] X0 **402** ist jedoch ein kombinierter Knoten, sodass er zerlegt anstatt erweitert werden muss. [Fig. 9](#) zeigt das Ergebnis des Zerlegens von Knoten X = in seine Bestandssymbole B **422** und C **424**. Neuberechnen der Wahrscheinlichkeiten zeigt, dass die Symbolfolge "A B C" 1/8 erscheint, "A B X2" 1/8 erscheint, "A X1" 3/8 erscheint, "B" **422** 2/8 erscheint und "C" 1/8 erscheint. Da dies eine geteilte Operation ist, wird der geteilte Knoten mit höchster Wahrscheinlichkeit, z. B. B **422**, bewahrt, und die übrigen Knoten werden wieder in X0 = C **424** rekombiniert.

[0057] [Fig. 10](#) zeigt das Ergebnis des Bewahrens des Hochwahrscheinlichkeits-Knotens B **422**. Man beachte, dass die Gruppierung X0 jetzt nur ein einziges Symbol "C" darstellt. Nach Revidieren der Wahrscheinlichkeiten muss der Knoten mit höchster Wahrscheinlichkeit identifiziert und geteilt oder erweitert werden. Wie gezeigt erscheint die Symbolfolge "A B A" 1/8, "A B X2" erscheint 1/8, "A X1" erscheint 3/8, "B" erscheint 2/8 und "X0" erscheint 1/8. Deshalb muss Knoten X1 **412**, als ein kombinierter Knoten, geteilt werden.

[0058] Das Teilen geht wie oben erörtert weiter, und die Verarbeitung des Codebuches wiederholt sich, wie in [Fig. 3](#) gezeigt, wobei Knoten höchster Wahrscheinlichkeit erweitert oder geteilt werden, bis ein Stopp-Zustand erreicht wird (z. B. das Codebuch erreicht eine Maximalgröße). Sobald das Codebuch einen Stopp-Zustand erreicht hat, steht es zur Codierung von Daten zum Senden über einen Kommunikationskanal zur Verfügung. Man beachte, dass für die Konfiguration in [Fig. 10](#) die mittlere Zahl von Bits pro Eingangssymbol, gebrochene Bits unter "idealer" Skalar-Huffman-Codierung angenommen, etwa 0.8 Bits/Symbol beträgt (variiert abhängig davon, wie der nachlaufende Eingang "A B" behandelt wird). Dies stellt eine wesentliche Einsparung (etwa 10%) gegenüber früheren verlustlosen Kompressionsverfahren dar.

[0059] [Fig. 11](#) zeigt ein Übertragungsmodell zum Übertragen von Audiodaten über einen Kanal **460**. Es wird angenommen, dass der Kanal **460** eine begrenzte Bandbreite hat, und deshalb einige Kompression von Quelldaten **450** erforderlich ist, bevor die Daten zuverlässig gesendet werden können. Man beachte, dass, obwohl diese Erörterung auf das Übertragen von Audiodaten gerichtet ist, die Erfindung auch für das Übertragen anderer Daten gilt, z. B. audiovisuelle Information mit eingebetteten Audiodaten (z. B. gemultiplext in einem MPEG-Datenstrom) oder andere Datenquellen mit kompressiblen Datenmustern (z. B. kohärente Daten).

[0060] Wie gezeigt werden Quelldaten **450** in einen Zeit/Frequenz-Transformationscoder **452**, z. B. eine Filterbank oder diskrete Kosinustransformation, eingegeben. Der Transformationscoder **452** ist bestimmt, einen kontinuierlichen oder abgetasteten Zeitbereichseingang, z. B. eine Audiodatenquelle, in mehrfache Frequenzbänder von vorbestimmter (obwohl vielleicht unterschiedlicher) Bandbreite umzuwandeln. Diese Bänder können dann bezüglich eines menschlichen Hörwahrnehmungsmodells **454** (z. B. ein psychoakustisches Modell) analysiert werden, um Komponenten des Signals zu bestimmen, die ohne hörbare Auswirkung sicher reduziert werden können. Zum Beispiel ist bekannt, dass bestimmte Frequenzen unhörbar sind, wenn bestimmte andere Töne oder Frequenzen in dem Eingangssignal vorhanden sind (gleichzeitige Maskierung). Folglich können solche unhörbaren Signale sicher aus dem Eingangssignal entfernt werden. Der Gebrauch von menschlichen Hörmodellen ist wohl bekannt, z. B. die MPEG 1, 2 und 4 Standards. (Man beachte, dass solche Modelle mit einer Quantisierungsoperation **456** verbunden werden können).

[0061] Nach Durchführen der Zeit/Frequenz-Transformation werden Frequenzkoeffizienten innerhalb jedes Bereiches quantisiert **456**, um jeden Koeffizienten (Amplitudenpegel) in einen aus einem endlichen

Satz von möglichen Werten genommenen Wert umzuwandeln, wobei jeder Wert eine Größe hat, der auf den zum Darstellen des Frequenzbereiches zugeteilten Bits basiert. Der Quantisierer kann ein herkömmlicher gleichförmiger oder ungleichförmiger Quantisierer, z. B. ein 'Midriser'- oder 'Midreader'-Quantisierer, mit (oder ohne) Speicher sein. Das grundsätzliche Quantisierungsziel ist das Identifizieren einer optimalen Bitzuteilung zum Darstellen der Eingangssignaldaten, d. h. den Gebrauch von verfügbaren Codierbits zu verteilen, um das Codieren der (akustisch) signifikanten Teile der Quelldaten sicherzustellen. Verschiedene Quantisierungsverfahren, z. B. Quantisierungsschrittgröße-Vorhersage, um eine gewünschte Bitrate (konstante Bitrate angenommen) zu erfüllen, können verwendet werden. Nachdem die Quelle **450** quantisiert **456** ist, werden die entstandenen Daten entsprechend dem Codebuch von [Fig. 3](#) entropiecodiert **458**.

[0062] [Fig. 12](#) zeigt ein Verfahren zur Implementierung des Entropiecoders **458** von [Fig. 11](#) durch Anwenden des Codebuches von [Fig. 3](#) auf die quantisierten Daten. Das Codebuch für variabel zu variabel Codierung kann benutzt werden, um andere Typen von Daten zu codieren. Wie gezeigt werden die quantisierten Daten als Eingabe in den Entropiecoder **458** von [Fig. 11](#) empfangen **480**. Es versteht sich, dass die Eingabe in irgendeiner Form von diskreten Signalen oder Paketdaten ist, und dass zur Einfachheit der Erörterung alle Eingaben als eine lange Serie von diskreten Symbolen angenommen werden. Die empfangene Eingabe **480** wird abgetastet **482**, um einen entsprechenden Codebuchschlüssel in dem Codebuch von [Fig. 3](#) aufzufinden. Ein solches Abtasten entspricht einem Daten-Nachsehen, und abhängig von der zur Implementierung des Codebuches benutzten Datenstruktur wird das genaue Verfahren des Nachsehens variieren.

[0063] Es stehen verschiedene Verfahren zur Speicherung und Bearbeitung des Codebuches zur Verfügung. Eine Struktur für ein Codebuch ist transversal und Speicherung eines N-rangigen (binär, tertiär usw.) Baumes, wo Symbolgruppierungen eine Querung der Baumstruktur bestimmen. Der Pfad eines Blattknotens des Baumes stellt das Ende einer erkannten Symbolsequenz dar, wobei ein Entropiecode mit der Sequenz verbunden ist. (Man beachte, dass das Codebuch als eine Tabelle implementiert werden kann, wo ein Tabelleneintrag die ganze Eingabesequenz enthält, z. B. der Pfad zu dem Knoten.) Knoten können in Software als eine Struktur, Klassendefinition oder andere Struktur codiert werden, die die Speicherung eines oder mehr mit dem Knoten verbundener Symbole und Verbinden eines entsprechenden Entropiecodes erlaubt.

[0064] Alternativ kann das Codebuch als eine Tabelle strukturiert sein, wobei jede Folge von Eingabe-

symbolen nach der Wahrscheinlichkeit des Vorkommens sortiert ist und eine hoch wahrscheinliche Eingabe an der Spitze der Tabelle steht. Für große Tabellen kann die Tabelle nach dem ersten Symbol sortiert sein, d. h. alle Symbolserien beginnend mit "A" sind in einer Gruppe zusammengefasst, gefolgt von Serien beginnend mit "B" usw. Bei dieser Anordnung sind alle Einträge in der Gruppierung nach der Wahrscheinlichkeit ihres Vorkommens sortiert. Die Position des Beginns jedes Abschnitts wird markiert/verfolgt, sodass eine Hash-Funktion (z. B. ein Nachsehen basierend auf dem ersten Symbol) benutzt werden kann, um den richtigen Teil der Codebuchtafel ausfindig zu machen. Bei dieser Nachseh-Tabellenlösung zum Speichern des Codebuches, wird, sobald das erste Symbol gefunden ist, der entsprechende Tabellenabschnitt ausgiebig durchsucht, bis ein übereinstimmender Eintrag ausfindig gemacht ist. Der mit dem übereinstimmenden Eintrag verbundene Code **484** wird dann als der codierte Ersatz ausgegeben **486**.

[0065] Mit [Fig. 11](#) fortfahrend wird, sobald der Ausgang **486** bekannt ist, dieser Ausgang über den Kommunikationskanal **460** übertragen. Das empfangende Ende **470** implementiert einen Umkehr-Codierprozess, d. h. eine Serie von Schritten, um die Codierung der Quelldaten **450** rückgängig zu machen. Das heißt, die codierten Daten **486** werden als Eingabe für einen Entropiedecoder **462** empfangen, der ein umgekehrtes Codebuch-Nachsehen durchführt, um den codierten Ausgang **486** wieder in die ursprüngliche Eingangssymbolserie **480** ([Fig. 12](#)) umzuwandeln. Die rückgewonnenen Eingangsdaten **480** werden dann durch einen Dequantisierer **464** und einen Zeit/Frequenz-Transformationsdecoder **466** verarbeitet, um die ursprünglichen Codieroperation umzukehren, was rekonstruierte Daten **468** ergibt, die den ursprünglichen Quelldaten **450** gleichen. Man sollte beachten, dass die rekonstruierten Daten **468** sich den ursprünglichen Quelldaten nur annähern, wenn, wie hierin angenommen, ein verlustbehaftetes System eingesetzt wird.

[0066] Nachdem die Prinzipien der Erfindung mit Bezug auf eine veranschaulichte Ausführung beschrieben und veranschaulicht wurden, wird man anerkennen, dass die veranschaulichte Ausführung in Aufbau und Detail modifiziert werden kann, ohne von diesen Prinzipien abzuweichen.

Patentansprüche

1. Verfahren zur Codierung einer Folge von digitalen Audiodatensymbolen (**450**) zum Übertragen über einen Kommunikationskanal (**460**), wobei das Verfahren das Identifizieren einer ersten Gruppierung variabler Größe von Audiodatensymbolen in der Folge von digitalen Audiodatensymbolen umfasst, wobei das Verfahren gekennzeichnet ist durch:

Codieren (**486**) der ersten Gruppierung variabler Größe von Symbolen mit einer Codieranordnung, die als Ausgang ein Entropie-Codewort erzeugt, das der ersten Gruppierung von Symbolen entspricht, wobei das Codieren ein vorher konstruiertes Codebuch verwendet, das Gruppierungen variabler Größe von Audiodatensymbolen mit entsprechenden Entropie-Codewörtern zur variabel-zu-variabel-Kompression verbindet, und

wiederholt Identifizieren und Codieren (**486**) von nachfolgenden Gruppierungen variabler Größe von Symbolen, sodass wenigstens zwei identifizierte Gruppierungen von Symbolen in der Folge von digitalen Audiodatensymbolen unterschiedliche Größen aufweisen.

2. Verfahren nach Anspruch 1, wobei das Codebuch entsprechend einem ersten Symbol jeder Gruppierung in dem Codebuch in Abschnitte gegliedert ist, wobei jeder Abschnitt weiter nach der Wahrscheinlichkeit des Vorkommens jedes Eintrags innerhalb jedes Abschnitts sortiert ist, und wobei das Identifizieren einer Gruppierung variabler Größe umfasst: Identifizieren eines Abschnitts durch ein erstes Symbol der Gruppierung variabler Größe, und Abgleichen der Gruppierung variabler Größe gegen jeden Abschnittseintrag, bis ein Gegenstück gefunden ist.

3. Verfahren nach Anspruch 2, wobei das Gegenstück eine Größe hat, wobei eine solche Größe eine neue Position in der Folge von digitalen Audiodatensymbolen zum Identifizieren einer zweiten Gruppierung variabler Größe von Symbolen angibt.

4. Verfahren nach Anspruch 3, wobei die erste und die zweite Gruppierung variabler Größe unterschiedliche Größen aufweisen.

5. Verfahren nach Anspruch 1, wobei das Codebuch als eine Tabelle mit Abschnitten aufgebaut ist, wobei solche Abschnitte durch ein erstes Symbol jeder Gruppierung variabler Größe von Symbolen definiert sind, und wobei Tabelleneinträge innerhalb eines Abschnitts entsprechend der Wahrscheinlichkeit des Vorkommens jeder Gruppierung variabler Größe von Symbolen in dem Abschnitt sortiert sind.

6. Verfahren nach einem der Ansprüche 1 bis 5, wobei ein Eingangskanal mit einem Plattenspeicher in Verbindung steht, wobei das Verfahren des Weiteren den Schritt des Lesens der Folge von digitalen Datensymbolen aus dem Plattenspeicher umfasst, um die Identifikation und Codierung der Gruppierungen variabler Größe zu gestatten.

7. Verfahren nach einem der Ansprüche 1 bis 5, das weiter umfasst: Empfangen der Folge von digitalen Audiodatensymbolen von einem Eingangskanal, und

nach dem Codieren (**486**) einer Gruppierung variabler Größe von Symbolen, Übertragen des der Gruppierung von Symbolen entsprechenden Entropie-Codewortes über den Kommunikationskanal (**460**).

8. Verfahren nach Anspruch 7, wobei die Folge von digitalen Audiodatensymbolen in Echtzeit empfangen wird, und das Übertragen in Echtzeit durchgeführt wird.

9. Verfahren nach Anspruch 7, das weiter das Empfangen einer Verbindungsanforderung von einem Client über eine Client-Netzwerkverbindung umfasst, wobei das Übertragen über die Client-Netzwerkverbindung durchgeführt wird.

10. Verfahren nach Anspruch 9, wobei die Folge von digitalen Audiodatensymbolen empfangen und in einem nicht flüchtigen Speicher gespeichert wird, und das Übertragen bis zum Empfang der Verbindungsanforderung von dem Client aufgeschoben wird.

11. Verfahren zur Decodierung eines komprimierten Audiodatenstromes von einem Kommunikationskanal (**460**), das umfasst:
Empfangen eines Entropie-Codewortes;
Nachsehen des Entropie-Codewortes in einem vorher konstruierten Codebuch, das eine Entsprechung zwischen Entropie-Codewörtern und Serien variabler Größe von Symbolen zur variabel-zu-variabel-Dekompression enthält, und
Ausgeben einer dem Codewort in dem Codebuch entsprechenden Serie variabler Größe von Audiodatensymbolen.

12. Verfahren nach Anspruch 11, wobei der Schritt des Nachsehens des Entropie-Codewortes "Hashing" des Entropie-Codewortes einschließt, um einen Index zu einem Eintrag in einer Hash-Tabelle zu erlangen.

13. Verfahren nach Anspruch 12, wobei das Codebuch entsprechend einem ersten Symbol jeder in dem Codebuch gespeicherten Serie variabler Größe von Symbolen in Abschnitte gegliedert ist.

14. Computerlesbares Medium, in dem computerführbare Anweisungen gespeichert sind, die einen Computer veranlassen, das Verfahren nach einem der Ansprüche 1 bis 13 durchzuführen.

15. System zum Übertragen eines komprimierten audiovisuellen Datenstromes von einem Server-Netzwerkservice an einen Client-Netzwerkservice über eine Netzwerkverbindung (**460**), wobei ein solcher komprimierter audiovisueller Datenstrom durch Ersetzen einer Folge variabler Größe von audiovisuellen Dateneingangssymbolen durch einen Ausgabe-Entropiecode gebildet wird, wobei das Sys-

tem umfasst

einen Eingangspuffer zur Speicherung einer zu komprimierenden und an den Client zu übertragenden Serie von unkomprimierten audiovisuellen Datensymbolen;

einen Ausgangsspeicher zur Speicherung eines Entropie-Codewortes, das eine komprimierte Version der Serie von unkomprimierten Symbolen in dem Eingangspuffer darstellt,

wobei das System gekennzeichnet ist durch:

einen Codebuchspeicher zur Speicherung eines vorher vorhandenen Codebuches, das ein Entropie-Codewort für eine Serie variabler Größe von Symbolen enthält, wobei das Codebuch Serien variabler Größe von Symbolen mit entsprechenden Entropie-Codewörtern zur variabel-zu-variabel-Kompression verbindet;

eine Suchanordnung zum Nachsehen eines Entropie-Codewortes für eine bestimmte Serie von Symbolen in dem Codebuch, und

eine Codieranordnung mit einem Eingangskanal, der mit dem Eingangspuffer in Verbindung steht, und einem Ausgang, der mit dem Ausgangsspeicher in Verbindung steht,

wobei die Codieranordnung die Suchanordnung anwendet, um das Entropie-Codewort für die Serie von unkomprimierten Symbolen zur Speicherung in dem Ausgangsspeicher nachzusehen.

16. System nach Anspruch 15, das weiter eine Übertragungseinrichtung zum Übertragen des Inhalts des Ausgangsspeichers an den Client über die Netzwerkverbindung (**460**) umfasst.

17. System nach Anspruch 16, wobei ein Streaming-Netzbetriebsprotokoll benutzt wird, um über die Netzwerkverbindung (**460**) zu kommunizieren.

18. System zur Decodierung von komprimierten audiovisuellen Daten, die von einer Netzwerkverbindung (**460**) empfangen werden, wobei das System umfasst:

einen Eingangsspeicher zur Speicherung eines Codewortes;

einen Ausgangspuffer zur Speicherung einer dem Codewort in dem Eingangsspeicher entsprechenden Serie von unkomprimierten audiovisuellen Datensymbolen,

wobei das System gekennzeichnet ist durch:

einen Codebuchspeicher zur Speicherung eines vorher vorhandenen Codebuches, das eine Serie variabler Größe von Symbolen für ein Entropie-Codewort enthält, wobei das Codebuch Entropie-Codewörter mit entsprechenden Serien variabler Größe von Symbolen zur variabel-zu-variabel-Dekompression verbindet;

eine Suchanordnung zum Nachsehen eines Entropie-Codewortes für eine bestimmte Serie von Symbolen in dem Codebuch, und

eine Decodieranordnung mit einem Eingangskanal,

der mit dem Eingangsspeicher in Verbindung steht, und einem Ausgang, der mit dem Ausgangspuffer in Verbindung steht, wobei die Decodieranordnung die Suchanordnung anwendet, um die dem Entropie-Codewort entsprechende Serie von umkomprimierten Symbolen nachzusehen und eine solche Serie in dem Ausgangspuffer zu speichern.

19. System nach Anspruch 18, wobei ein Streaming-Netzbetriebsprotokoll benutzt wird, um über die Netzwerkverbindung (**460**) zu kommunizieren.

20. System nach Anspruch 18, in dem die Decodieranordnung durch ein Anwendungsprogramm implementiert ist, das auch das "Hyper-Text Markup Language"-Protokoll implementiert.

Es folgen 7 Blatt Zeichnungen

FIG. 1

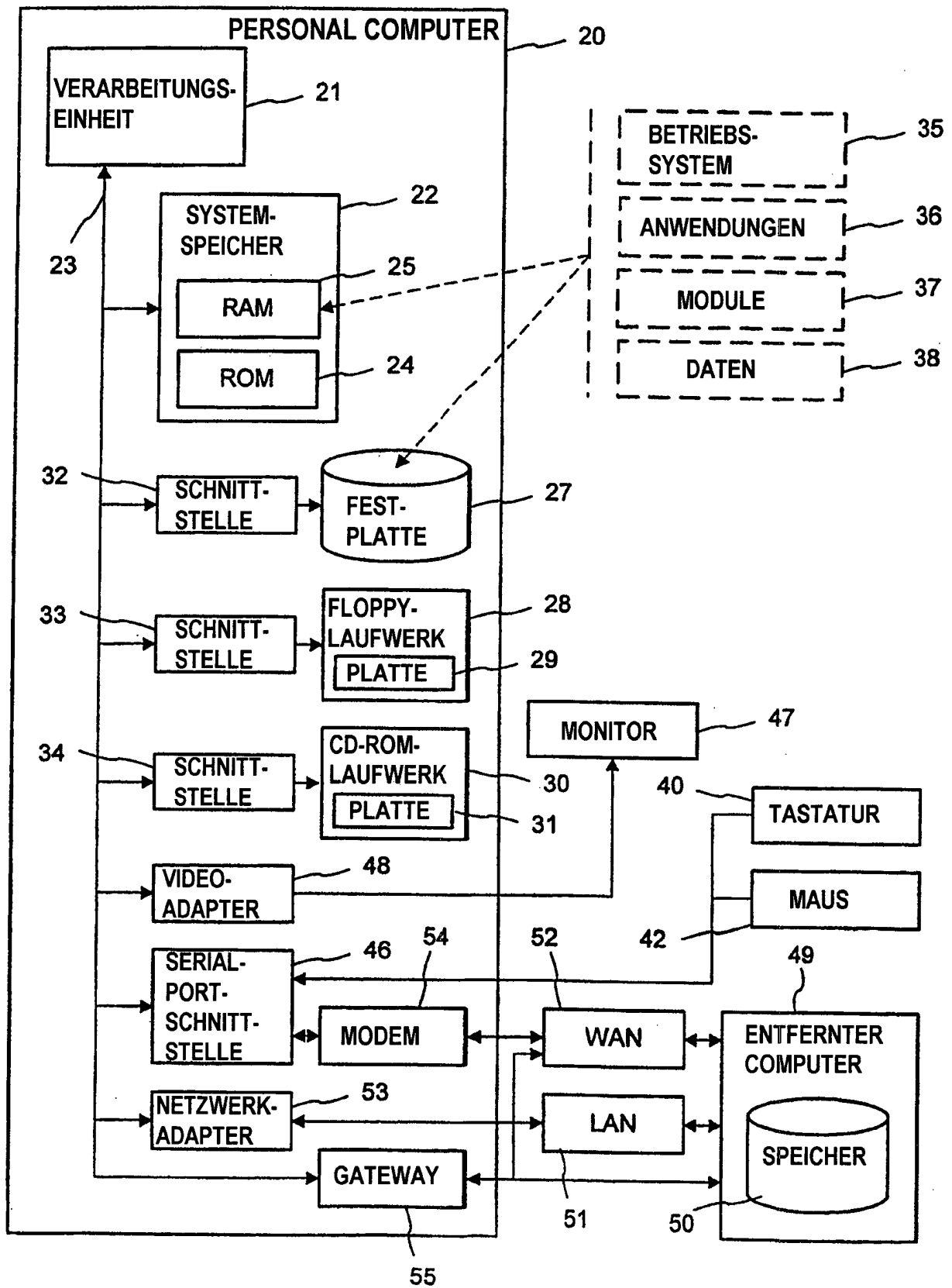


FIG. 2

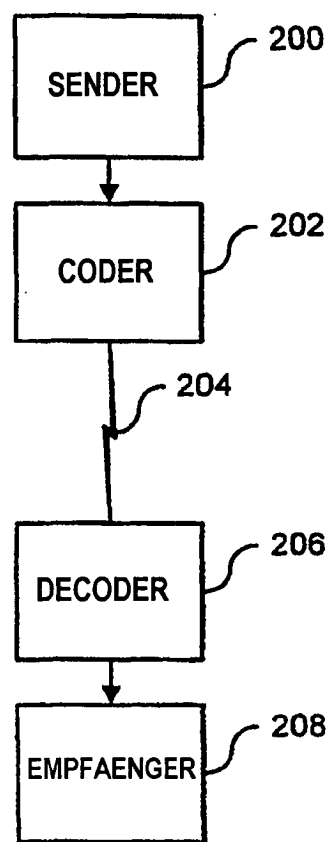


FIG. 3

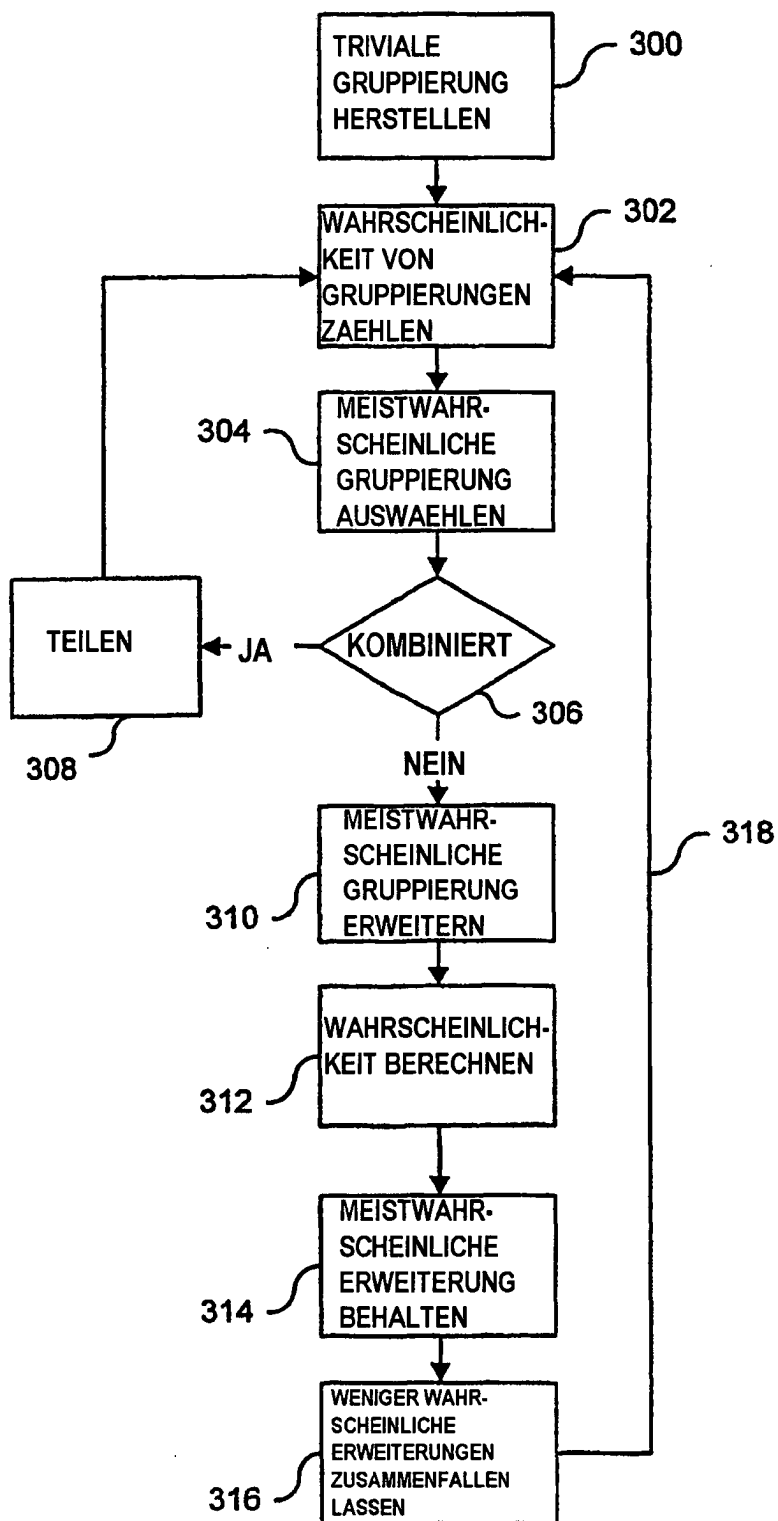
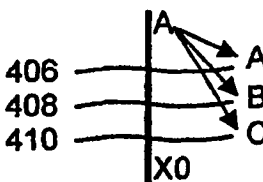


FIG. 4



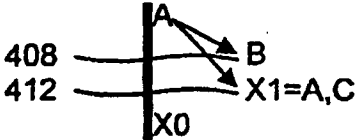
8/15 ERWEITERN
7/15

FIG. 5



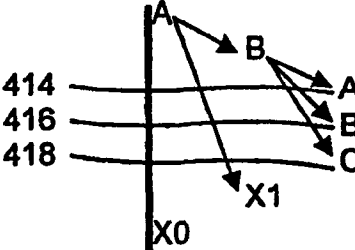
2/9
4/9 BEHALTEN
0/9

FIG. 6



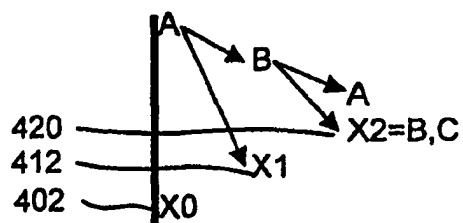
4/9 ERWEITERN
2/9
3/9

FIG. 7



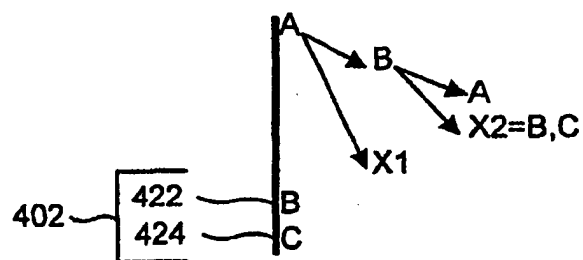
1/8 BEHALTEN
1/8
0/8
2/8
3/8

FIG. 8



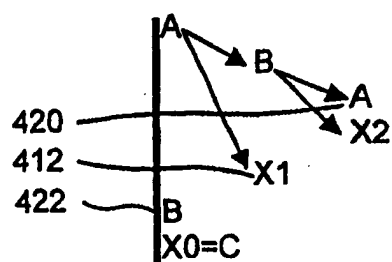
1/8
1/8
3/8
3/8 TEILEN

FIG. 9



1/8
1/8
3/8
2/8 BEHALTEN
1/8

FIG. 10



1/8
1/8
3/8 TEILEN
2/8
1/8

FIG. 11

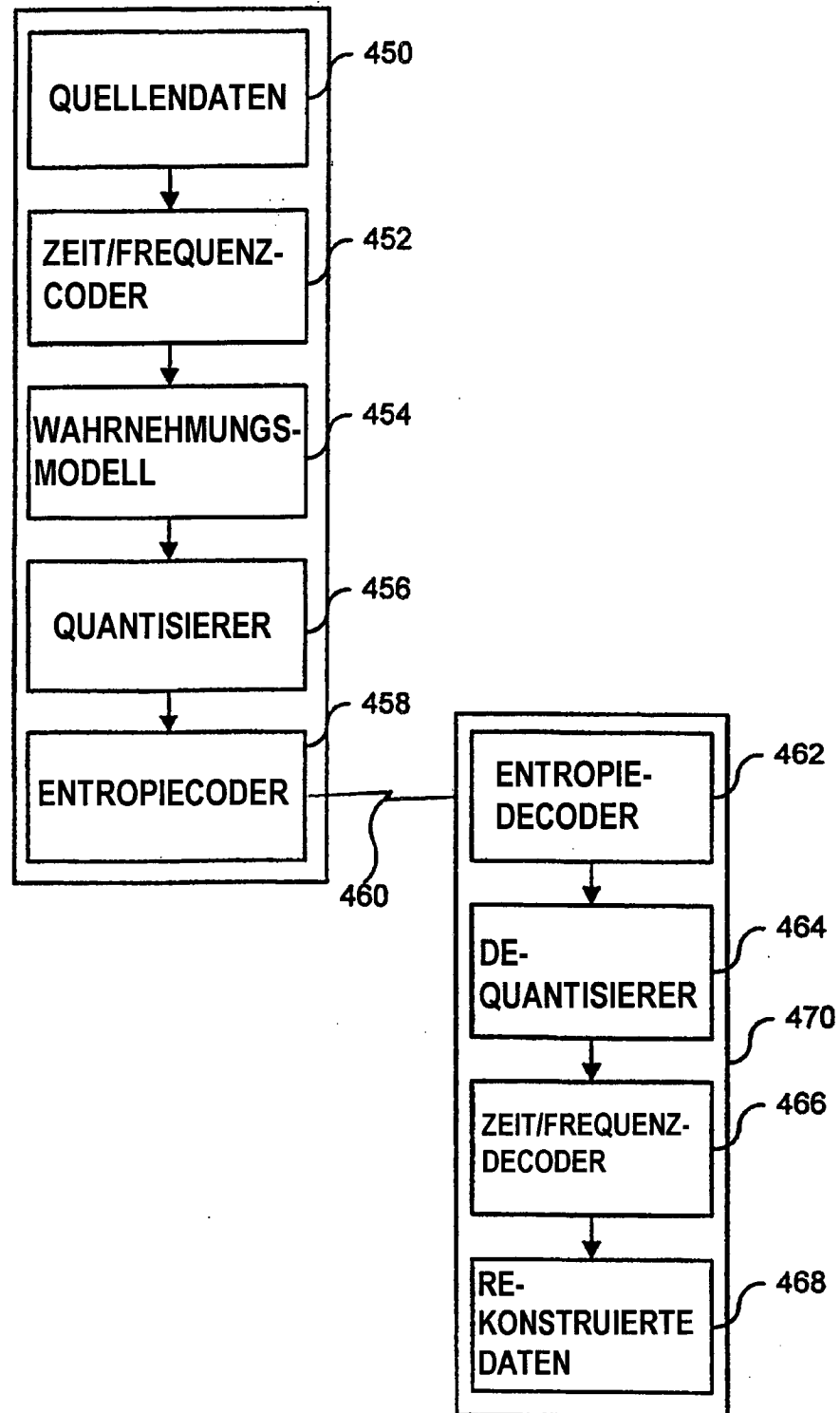


FIG. 12

