



(19)
Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) **DE 699 20 255 T2** 2005.09.15

(12) **Übersetzung der europäischen Patentschrift**

(97) **EP 1 129 537 B1**

(51) Int Cl.7: **H04L 1/00**

(21) Deutsches Aktenzeichen: **699 20 255.8**

(86) PCT-Aktenzeichen: **PCT/EP99/09030**

(96) Europäisches Aktenzeichen: **99 957 323.1**

(87) PCT-Veröffentlichungs-Nr.: **WO 00/30287**

(86) PCT-Anmeldetag: **12.11.1999**

(87) Veröffentlichungstag
der PCT-Anmeldung: **25.05.2000**

(97) Erstveröffentlichung durch das EPA: **05.09.2001**

(97) Veröffentlichungstag
der Patenterteilung beim EPA: **15.09.2004**

(47) Veröffentlichungstag im Patentblatt: **15.09.2005**

(30) Unionspriorität:
9824904 **13.11.1998** **GB**

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

(73) Patentinhaber:
Motorola Ltd., Basingstoke, Hampshire, GB

(72) Erfinder:
**PEARCE, John, David, Basingstoke, Hampshire
RG24 8WE, GB; GIBBS, Alastair, Jon,
Southampton, Hampshire SO30 2XF, GB; EVANS,
Christopher, Alun, Bristol, Avon BS76 5SZ, GB**

(74) Vertreter:
**SCHUMACHER & WILLSAU,
Patentanwaltssozietät, 80335 München**

(54) Bezeichnung: **VERARBEITUNG VON EMPFANGENEN DATEN IN EINEM VERTEILTEN SPRACHERKENNUNGS-
SYSTEM**

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

Gebiet der Erfindung

[0001] Die vorliegende Erfindung betrifft ein Verfahren zum Verarbeiten empfangener Daten in einem verteilten Spracherkennungsprozess. Die vorliegende Erfindung betrifft auch eine Vorrichtung zum Verarbeiten empfangener Daten in einem verteilten Spracherkennungsprozess. Die vorliegende Erfindung ist geeignet, ohne darauf beschränkt zu sein, für das Verarbeiten empfangener Daten, die sich auf Spracherkennungsparameter beziehen, wenn sie über eine Funkkommunikationsverbindung übertragen werden.

Hintergrund der Erfindung

[0002] Die Spracherkennung ist ein Prozess zum automatischen Erkennen von Klängen, Teilen von Wörtern, Wörtern oder Satzteilen von Sprache. Ein derartiger Prozess kann als Schnittstelle zwischen Mensch und Maschine verwendet werden, zusätzlich oder anstelle der üblicherweise verwendeten Mittel wie etwa Schalter, Tastaturen, der Maus usw. Ein Spracherkennungsprozess kann auch verwendet werden, um automatisch Informationen aus einer gesprochenen Kommunikation oder Nachricht zu erlangen.

[0003] Es wurden verschiedene Verfahren entwickelt und werden noch verbessert, um eine automatische Spracherkennung zur Verfügung zu stellen. Einige Verfahren basieren auf einem erweiterten Wissen mit entsprechenden heuristischen Strategien, andere verwenden statistische Modelle.

[0004] In typischen Spracherkennungsprozessen wird die zu verarbeitende Sprache mehrere Male im Verlauf eines Abtastzeitschlitzes abgetastet, beispielsweise 50 oder 100 Mal pro Sekunde. Die abgetasteten Werte werden unter der Verwendung von Algorithmen verarbeitet, um Spracherkennungsparameter zur Verfügung zu stellen. Beispielsweise besteht eine Art von Spracherkennungsparameter aus einem Koeffizienten, der als Mel-Cepstral-Koeffizient bekannt ist. Derartige Spracherkennungsparameter werden in der Form von Vektoren angeordnet, auch als Felder bekannt, was als eine mit einem bestimmten Ordnungsgrad angeordnete Parametergruppe oder -menge aufgefasst werden kann. Der Abtastprozess wird für weitere Abtastzeitschlitz wiederholt. Ein typisches Format besteht darin, dass für jeden Abtastzeitschlitz ein Vektor erstellt wird.

[0005] Die obige Parametrisierung und Platzierung in Vektoren stellt das dar, was als Frontend-Verarbeitung eines Spracherkennungsprozesses bezeichnet werden kann. Die oben beschriebenen in Vektoren angeordneten Spracherkennungsparameter werden dann entsprechend der Spracherkennungstechniken analysiert, was als Backend-Verarbeitung des Spracherkennungsprozesses bezeichnet werden kann. In einem Spracherkennungsprozess, in dem der Frontend-Prozess und der Backend-Prozess am gleichen Ort oder im gleichen Bauteil durchgeführt werden, ist die Wahrscheinlichkeit, Fehler in die Spracherkennungsparameter beim Weiterleiten vom Frontend in das Backend einzuführen, minimal.

[0006] In einem als verteilten Spracherkennungsprozess bekannten Prozess wird der Frontend-Teil des Spracherkennungsprozesses vom Backend-Teil entfernt durchgeführt. An einem ersten Ort wird die Sprache abgetastet, parametrisiert und die Spracherkennungsparameter in Vektoren angeordnet. Die Spracherkennungsparameter werden quantisiert und dann an einen zweiten Ort, beispielsweise über eine Kommunikationsverbindung eines eingerichteten Kommunikationssystems, übertragen. Oft kann der erste Ort ein Fernendgerät und der zweite Ort eine zentrale Verarbeitungsstation sein. Die empfangenen Spracherkennungsparameter werden dann entsprechend der Spracherkennungstechniken am zweiten Ort analysiert. Die quantisierten Spracherkennungsparameter und ihre Anordnung in Vektoren stellen Daten dar, die von der ersten Station gesendet und von der zweiten Station empfangen werden. Um die Übertragung dieser Daten zu erleichtern, werden die Daten typischerweise in einer Framestruktur angeordnet, die eine Mehrzahl von Datenframes umfasst, wobei jedem ein entsprechender Headerframe vorausgeht, der entsprechende gemeinsame Headerinformationen umfasst. Der Headerframe kann auch dergestalt sein, dass ein Headerframe zusätzlich Headerinformationen umfasst, die nur spezifisch für diesen Headerframe oder für den bestimmten, diesem Headerframe entsprechenden, Datenframe, ist.

[0007] Viele Arten von Kommunikationsverbindungen in vielen Arten von Kommunikationssystemen können zur Verwendung in einem verteilten Spracherkennungsprozess in Betracht gezogen werden. Ein Beispiel ist ein konventionelles drahtgebundenes Kommunikationssystem, beispielsweise ein öffentliches Telefonnetz. Ein anderes Beispiel ist ein Funkkommunikationssystem, wie beispielsweise TETRA. Ein anderes Beispiel ist ein zelluläres Funkkommunikationssystem. Ein Beispiel eines geeigneten zellularen Kommunikationssystems ist ein globales System für mobile Kommunikationssysteme (GSM), ein anderes Beispiel sind Systeme wie etwa

das universale Mobiltelekommunikationssystem (UMTS), das sich momentan in der Standardisierung befindet.

[0008] Um jegliche Verwirrung zu vermeiden, wird betont, dass die oben beschriebenen Datenframes nicht mit den Übertragungsframes verwechselt werden sollen, die danach in der Übertragung der Daten über die Kommunikationsverbindung des Kommunikationssystems verwendet werden, in dem die Daten von einem ersten Ort an einen zweiten Ort übertragen werden, beispielsweise die Zeitmultiplex-Zeitframes ("TDMA = time division multiple access") eines zellularen GSM-Funkkommunikationssystems.

[0009] Die Verwendung einer beliebigen Kommunikationsverbindung in einem beliebigen Kommunikationssystem bringt die Möglichkeit mit sich, dass Fehler in die Daten und auch in die Headerinformationen eingeführt werden, die vom ersten Ort an den zweiten Ort über die Kommunikationsverbindung übertragen werden.

[0010] Aufgrund der spezialisierten Spracherkennungstechniken, denen die Sprachparameter unterworfen werden, ist es wünschenswert, Mittel zum Verarbeiten der empfangenen Daten zur Verfügung zu stellen, die einen Grad an Widerstandsfähigkeit gegenüber Fehlern, die in die Headerinformationen eingeführt wurden, in einer Weise anbieten, die insbesondere zu den Charakteristiken des verteilten Spracherkennungsprozesses passt.

[0011] Des weiteren ist es bekannt, in Kommunikationssystemen Fehlerdetektionstechniken zur Verfügung zu stellen, so dass das Vorhandensein eines Fehlers in einem gegebenen Abschnitt von übermittelter Information detektierbar ist. Eine wohlbekanntetechnik ist das zyklische Redundanzcodieren ("cyclic redundancy coding"). Es ist ebenso bekannt, automatische Fehlerkorrekturtechniken in Kommunikationssystemen zur Verfügung zu stellen, so dass ein Fehler in einem gegebenen Abschnitt von übertragener Information korrigiert wird. Eine wohlbekanntetechnik ist die Golay-Fehlerkorrektur. Es ist ebenso bekannt, Fehlerdetektion und Fehlerkorrektur in Kombination einzusetzen.

[0012] Wenn eine automatische Fehlerkorrektur angewendet wird, besteht das Risiko, dass die korrigierte Form des gesamten Informationsabschnittes, der korrigiert wird, weitere Diskrepanzen abweichend vom ursprünglichen Fehlerbestandteil enthält, da derartige verfahren dazu neigen, eine Näherung an eine beste, als insgesamt richtig angesehene Lösung zu beteiligen. Dies ist bei Vorwärtsfehlerkorrekturtechniken der Fall, die das Codieren unter der Verwendung eines Codierschemas auf Blockbasis einsetzen. Ein derartiges Beispiel ist Golay-Codieren, das es beispielsweise ermöglicht, 12 Informationsbits in 24 Bits zu senden, während bis zu 3 Fehler korrigiert werden dürfen. Die Korrekturtechnik bedingt die Korrektur eines ganzen Informationsabschnittes beispielsweise eines gesamten Headerframes in einer Verbundweise. Wenn jedoch mehr als 3 Fehler in den 24 Bits vorkommen, dann korrigiert die Korrekturtechnik den gesamten Header in eine falsch korrigierte Version. Es ist wünschenswert, Mittel zum Verarbeiten empfangener Daten zur Verfügung zu stellen, die die Probleme erleichtern, die mit der Verbundkorrektur eines gesamten Headerframes in eine falsch korrigierte Version in einem verteilten Spracherkennungsprozess verbunden sind.

[0013] Des weiteren sind Techniken zur automatischen Fehlerkorrektur, die evtl. keine Sekundärprobleme beim Anwenden auf andere Informationsformen verursachen, nicht notwendigerweise ohne Probleme, wenn sie auf Fehler in den oben beschriebenen Headerframes in einem verteilten Spracherkennungsprozess angewendet werden, teilweise aufgrund der Art und Weise, wie die Daten in den entsprechenden Datenframes unter der Verwendung der entsprechenden Headerframeinformation verarbeitet werden. Deshalb ist es wünschenswert, Mittel zum Verarbeiten empfangener Daten in einem verteilten Spracherkennungsprozess zur Verfügung zu stellen, die Sekundärprobleme erleichtern.

[0014] Im Stand der Technik ist ein Verfahren zum Verarbeiten empfangener Daten in einem verteilten Spracherkennungsprozess in der internationalen Patentanmeldung WO 95 177 46 A offenbart, aber ein Betrieb wie in der jetzt zu beschreibenden Erfindung ist nicht im Stand der Technik offenbart oder nahegelegt.

Zusammenfassung der Erfindung

[0015] Vorliegende Erfindung betrifft einige oder alle der obigen Aspekte.

[0016] Gemäß einem ersten Aspekt der vorliegenden Erfindung wird ein Verfahren zum Verarbeiten empfangener Daten in einem verteilten Spracherkennungsprozess zur Verfügung gestellt, wie in Anspruch 1 beansprucht.

[0017] Entsprechend einem anderen Aspekt der Erfindung wird eine Vorrichtung zum Verarbeiten empfangener

ner Daten in einem verteilten Spracherkennungsprozess zur Verfügung gestellt, wie in Anspruch 7 beansprucht.

[0018] Weitere Aspekte der Erfindung sind in den abhängigen Ansprüchen beansprucht.

[0019] Die vorliegende Erfindung ist darauf gerichtet, Mittel zum Verarbeiten empfangener Daten zur Verfügung zu stellen, die insbesondere geeignet sind für die Beschaffenheit des verteilten Spracherkennungsprozesses, die Form, in der Daten darin empfangen werden, wenn sie von einem ersten Ort an einen zweiten Ort übertragen werden und die Art und Weise, in der derartige Daten nach dem Empfang an einem zweiten Ort in einem verteilten Spracherkennungsprozess verarbeitet werden.

[0020] Insbesondere wird in dem Verfahren der vorliegenden Erfindung die Möglichkeit, Latenzzeiten in einem Spracherkennungsprozess zu ermöglichen, ausgenutzt. Speziell wird der Faktor ausgenutzt, dass in einem verteilten Spracherkennungsprozess eine Latenzzeit in Richtung des Beginns einer Nachricht oft insbesondere akzeptierbar ist, wenn sie mit einer niedrigen Latenzzeit am Ende einer Nachricht kombiniert wird.

[0021] Zusätzliche spezielle Vorteile werden aus der folgenden Beschreibung und den Figuren ersichtlich.

Kurze Beschreibung der Zeichnungen

[0022] [Fig. 1](#) ist eine schematische Darstellung eines in Vektoren angeordneten Spracherkennungsparameters, der Abtastzeitframes einer Ausführungsform der vorliegenden Erfindung entspricht.

[0023] [Fig. 2](#) ist eine schematische Darstellung eines Headerframes und in einem Datenframe angeordneten Bit-Stromframes einer Ausführungsform der vorliegenden Erfindung.

[0024] [Fig. 3](#) ist eine schematische Darstellung von Datenframes und Headerframes einer Ausführungsform der vorliegenden Erfindung.

[0025] [Fig. 4](#) ist ein Prozessablaufdiagramm einer Ausführungsform der vorliegenden Erfindung.

Beschreibung einer bevorzugten Ausführungsform der Erfindung

[0026] In den unten beschriebenen beispielhaften Ausführungsformen werden die Spracherkennungsparameter in Vektoren arrangiert, die Abtastzeitframes, wie schematisch in [Fig. 1](#) dargestellt, entsprechen.

[0027] Ein Abschnitt eines zu verarbeitenden Sprachsignals **101** ist in [Fig. 1](#) abgebildet. Das Sprachsignal **100** ist in stark vereinfachter Form dargestellt, da es in der Praxis aus einer wesentlich komplizierteren Sequenz von Abtastwerten besteht. Die Abtastzeitframes, von denen in [Fig. 1](#) ein erster Abtastzeitframe **121**, ein zweiter Abtastzeitframe **122**, ein dritter Abtastzeitframe **123** und ein vierter Abtastzeitframe **124** gezeigt sind, sind über das Sprachsignal gelegt, wie in [Fig. 1](#) abgebildet. In der unten beschriebenen Ausführungsform gibt es 100 Abtastzeitframes pro Sekunde. Das Sprachsignal wird wiederholt im Verlauf jedes Abtastzeitframes abgetastet.

[0028] In den unten beschriebenen Ausführungsformen ist der Spracherkennungsprozess einer, in dem insgesamt vierzehn Spracherkennungsparameter eingesetzt werden. Die ersten zwölf davon sind die ersten zwölf statischen Mel-Cepstral-Koeffizienten, d. h.

$$c(m) = [c_1(m), C_2(m), \dots, c_{12}(m)]^T,$$

wobei m die Abtastzeitframennummer bezeichnet. Der dreizehnte eingesetzte Spracherkennungsparameter ist der – nullte Cepstralkoeffizient, d. h. $c_0(m)$. Der vierzehnte eingesetzte Spracherkennungsparameter ist ein logarithmischer Energieterm, d. h. $\log [E(m)]$. Die Details dieser Koeffizienten und ihre Verwendung in Spracherkennungsprozessen ist im Stand der Technik wohlbekannt und bedarf hier keiner weiteren Beschreibung. Des Weiteren sei es vermerkt, dass die Erfindung in anderen Kombinationen von Cepstralkoeffizienten, die Spracherkennungsparameter bilden, durchgeführt werden kann, gleichermaßen mit einer anderen Auswahl oder anderen Schemata von Spracherkennungsparametern als die Cepstralkoeffizienten.

[0029] Die vierzehn Parameter werden für jeden Abtastzeitframe in einen entsprechenden Vektor angeordnet oder formatiert, der auch als Feld bekannt ist, wie in [Fig. 1](#) dargestellt. Der Vektor **131** entspricht dem Abtast-

zeitframe **121**, der Vektor **132** entspricht dem Abtastzeitframe **122**, der Vektor **133** entspricht dem Abtastzeitframe **123** und der Vektor **134** entspricht dem Abtastzeitframe **124**. Ein derartiger Vektor kann allgemein dargestellt werden durch

$$y(m) = \begin{bmatrix} c(m) \\ c_0(m) \\ \log[E(m)] \end{bmatrix}$$

[0030] Die Spracherkennungsparameter werden vor der Übertragung von einem ersten Ort an einen zweiten Ort verarbeitet. In der unten beschriebenen Ausführungsform wird dies wie folgt durchgeführt. Die Parameter des Vektors **131** werden quantisiert. Dies ist durch das direkte Quantisieren des Vektors mit einem geteilten Vektorquantisierer ("split vector quantizer") realisiert. Die Koeffizienten werden in Paare gruppiert und jedes Paar wird unter Verwendung eines für dieses entsprechende Paar vorbestimmten Vektorquantisierungscodetabells (VQ) quantisiert. Die sich ergebende Menge von Indexwerten wird dann verwendet, um den Sprachframe darzustellen. Die Koeffizientenpaarungen von Frontend-Parametern stellen sich wie in Tabelle 1 abgebildet dar, zusammen mit der für jedes Paar verwendeten Codetabellgröße.

Tabelle 1

Eigenschaftspaarungen der geteilten Vektorquantisierung

Codebuch	Größe	Matrixgewichtung $W^{i,j+1}$	Element 1	Element 2
$Q^{0,1}$	64	I	C_1	C_2
$Q^{2,3}$	64	I	C_3	C_4
$Q^{4,5}$	64	I	C_5	C_6
$Q^{6,7}$	64	I	C_7	C_8
$Q^{8,9}$	64	I	C_9	C_{10}
$Q^{10,11}$	64	I	C_{11}	C_{12}
$Q^{12,13}$	256	Nicht-Identität	C_0	$\log[E]$

[0031] Man findet den nächsten VQ-Schwerpunkt unter Verwendung eines gewichteten euklidischen Abstandes, um den Index zu bestimmen,

$$d_j^{i,i+1} = \begin{bmatrix} y_i(m) \\ y_{i+1}(m) \end{bmatrix} - q_j^{i,j+1}$$

$$idx^{i,i+1}(m) = \underset{0 \leq j \leq (N^{i,i+1} - 1)}{\operatorname{argmin}} \left\{ (d_j^{i,i+1})^T W^{i,i+1} (d_j^{i,i+1}) \right\}, i = 0, 2, 4, \dots, 12$$

wobei $q_j^{i,i+1}$ den j-ten Codevektor im Codebuch $Q^{i,i+1}$ bezeichnet, $N^{i,i+1}$ die Größe des Codebuchs ist, $W^{i,i+1}$ die (möglicherweise Gleichheit) gewichtete Matrix, die für das Codebuch $Q^{i,i+1}$ anzuwendende Matrix ist und $idx^{i,i+1}(m)$ den Codebuchindex bezeichnet, der zur Darstellung des Vektors $[y_i(m), y_{i+1}(m)]^T$ gewählt wurde.

[0032] Die erstellten Indizes werden dann in der Form von 44 Bits dargestellt. Diese **44** Bits werden in den ersten 44 Schlitzen eines Bitstromframes **150** platziert, wie es vom Referenzzeichen **141** der [Fig. 1](#) dargestellt wird. Die entsprechenden für den folgenden Vektor, nämlich den Vektor **132**, erstellten 44 Bits werden in den nächsten 44 Schlitzen des Bitstromframes **150** platziert, wie vom Referenzzeichen **142** der [Fig. 1](#) dargestellt. Die verbleibenden Bits des Bitstromframes **150** bestehen aus 4 Bits zyklischem Redundanzcode, wie vom Re-

ferenzzeichen **146** der [Fig. 1](#) dargestellt, wobei der Wert der Bits bestimmt wird, so dass eine Fehlerdetektion in bekannter Art und Weise für die Gesamtheit der vorausgehenden 88 Bits des Bitstromframes **150** zur Verfügung gestellt wird. In ähnlicher Weise werden die vom Vektor **133** zur Verfügung gestellten 44 Bits in den ersten 44 Schlitzen eines zweiten Bitstromframes **155**, wie vom Referenzzeichen **143** der [Fig. 1](#) dargestellt, platziert. Ebenso werden die für den folgenden Vektor, nämlich den Vektor **134**, erstellten entsprechenden 44 Bits in den nächsten 44 Schlitzen des Bitstromframes **155**, wie vom Referenzzeichen **144** der [Fig. 1](#) dargestellt, platziert. Die verbleibenden Bits des Bitstromframes **155** bestehen aus 4 Bits zyklischem Redundanzcode, wie vom Referenzzeichen **148** der [Fig. 1](#) dargestellt. Diese Anordnung wird die folgenden Vektoren wiederholt. Das oben beschriebene Format des Bitstromframes, in dem Bitdaten zweier Vektoren in einem einzigen kombinierten Bitstromframe angeordnet werden, ist lediglich beispielhaft. Beispielsweise könnten die Daten eines jeden Vektors stattdessen in einem einzigen Bitstromframe angeordnet werden, der seine eigenen Fehlerdetektionsbits enthält. In ähnlicher Weise ist die Anzahl von Schlitzen pro Bitstromframe lediglich beispielhaft.

[0033] Die enthaltenen und angeordneten Daten in den oben beschriebenen Bitstromframes wird weiterhin wie in [Fig. 2](#) abgebildet angeordnet. Die Bitstromframes sind in Datenframes angeordnet, wobei jeder Datenframe einen oder mehreren Bitstromframes enthält. In der vorliegenden Ausführungsform enthält jeder Datenframe 12 Bitstromframes. Im vorliegenden Beispiel ist deshalb der Datenframe **230** in [Fig. 2](#) abgebildet und besteht aus den Bitstromframes **150** und **155** plus 10 weiteren Bitstromframes, mit **211-220** bezeichnet. Dem Datenframe **230** geht sein eigener entsprechender Headerframe **240** voraus, wie in [Fig. 2](#) abgebildet. Die entsprechenden Datenframes, dem jeder sein entsprechender Headerframe vorausgeht, sind fortlaufend wie in [Fig. 3](#) abgebildet angeordnet, wobei der Headerframe **240** seinem entsprechenden Datenframe **230** vorausgeht, der Headerframe **310** seinem entsprechenden Datenframe **320** vorausgeht und der Headerframe **330** seinem entsprechenden Datenframe **340** vorausgeht. In der Praxis folgen viele derartige Headerframes mit entsprechenden Datenframes. In der vorliegenden Ausführungsform besteht jeder Headerframe aus 24 Schlitzen. Das oben beschriebene Format der Datenframes und Headerframes ist lediglich beispielhaft. Ähnlich sind die Anzahl von Bitstromframes pro Datenframe und die Anzahl von Schlitzen in einem Headerframe lediglich beispielhaft.

[0034] Jeder Headerframe umfasst gemeinsame Headerinformationen. Dies sind Informationen, die in jedem Header gleich sind. Diese Informationen werden verwendet, wenn die Daten, wie oben beschrieben, verarbeitet werden, wobei das Beispiel gegeben wurde, dass die verwendete Information diejenige war, welche Abtastrate anwendbar ist. Im vorliegenden Beispiel sind die möglichen Abtastratenwerte 8 kHz und 16 kHz. Eine andere Möglichkeit sind Informationen, ob ein Standard- oder ein rauschfestes Protokoll oder Spur anwendbar ist.

[0035] Jeder Headerframe kann optional weiterhin framespezifische Informationen umfassen, dies sind Informationen, die sich zu einem gewissen Grad für unterschiedliche Köpfe unterscheidet. Ein Beispiel für framespezifische Informationen enthält einen Nachrichtenende-Indikator, dies ist ein Signal, das anzeigt, dass die Nachricht endet, entsprechend einem ersten Nachrichtenende-Erkennungsverfahren. In den vorliegenden Beispielen besteht das erste Nachrichtenende-Erkennungsverfahren aus dem Suchen nach aus lauter Nullen bestehenden Bitstromframes innerhalb des letzten Datenframes, was von dem Nachrichtenende-Indikator innerhalb des Headers angezeigt wird, die vom Sender eingefügt werden, um den Datenframe bis zur richtigen Länge aufzufüllen. Der Nachrichtenende-Indikator besteht aus einer einzigen Bitmenge innerhalb des framespezifischen Datenabschnitts des Headers. Ein anderes Beispiel framespezifischer Information ist eines, das eigentlich einen Parallelkanal zur Sprache darstellt, beispielsweise ein Signal eines Tastendrucks.

[0036] Ein optionaler Aspekt der vorliegenden Erfindung, in dem die Headerinformation unter Verwendung eines Codierschemas auf Blockbasis mit einer Vorwärtsfehlerkorrektur codiert ist, wird in den unten beschriebenen beispielhaften Ausführungsformen eingesetzt. Die spezielle Art einer derartigen in den vorliegenden Beispielen eingesetzten Fehlersicherung ist die Golay-Fehlersicherung, deren allgemeine Eigenschaften und Verwendungsmöglichkeiten im Stand der Technik wohlbekannt sind. Im vorliegenden Fall sind die speziellen Werte usw. wie folgt eingesetzt. Die Information für den Headerframe wird in einem [24, 12, 28] erweiterten systematischen Golay-Codewort dargestellt. Dieser Code unterstützt 12 Bits an Daten und hat eine Fehlerkorrekturmöglichkeit für bis zu 3 Bitfehler. Dieser Prozess stellt eine Kapazität bis insgesamt 24 Bits zur Verfügung, bestehend aus 12 für die Headerinformation verfügbarer Bits plus 12 Paritätsbits aus der Fehlersicherung. Wenn die zu verwendende Headerinformation derart ist, dass sie weniger als ihre insgesamt verfügbaren 12 Bits im Schema benötigt, dann können derartige Bits dauerhaft auf Null gesetzt werden und effektiv als verfügbar zur Benutzung bei zukünftigen Änderungen bestimmt werden, d. h. sogenannte Erweiterungsbits. Das Anordnen der 24 Bits in Felder wird in dem Fachmann bekannter Art und Weise durchgeführt. Im vorliegenden

Beispiel ist die Anordnung wie in unten stehender Tabelle 2 abgebildet und die Definition dieser Felder ist in unten stehender Tabelle 3 abgebildet.

Tabelle 2

Bit	8	7	6	5	4	3	2	1	Oktett
	EXP5	EXP4	EXP3	EXP2	EXP1	feType	endflag	sam rate	1
	P4	P3	P2	P1	EXP9	EXP8	EXP7	EXP6	2
	P12	P11	P10	P9	P8	P7	P6	P5	3

Tabelle 3

Feld	Bedeutung	Code	Indikator
sam rate	Abtastrate	0	8 kHz
		1	16 kHz
endFlag	Letzter Mehrfachframe	0	Standard
		1	Letzter Frame
feType	Frontend-Spezifikation	0	Standard (Spur 1)
		1	Rauschfest (Spur 2)
EXP1-EXP9	Erweiterungsbits (TBD)	0	(Null-Auffüllung)
P1-P12	Golay-Code-Paritätsbits	(siehe unten)	

[0037] Das verwendete Golay-Erzeugungspolynom ist

$$g_1(X) = 1 + x^2 + x^4 + x^5 + x^6 + x^{10} + x^{11}$$

[0038] Der Standard- [23, 12] Golay-Code ist mit der Hinzufügung eines geraden Gesamtparitätskontrollbits auf 24 Bits erweitert. Die Paritätsbits des Codeworts werden unter der Verwendung der Berechnung

$$\begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \\ P_6 \\ P_7 \\ P_8 \\ P_9 \\ P_{10} \\ P_{11} \\ P_{12} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}^T \cdot \begin{bmatrix} \text{sampRate} \\ \text{endFlag} \\ \text{feType} \\ \text{EXP 1} \\ \text{EXP 2} \\ \text{EXP 3} \\ \text{EXP 4} \\ \text{EXP 5} \\ \text{EXP 6} \\ \text{EXP 7} \\ \text{EXP 8} \\ \text{EXP 9} \end{bmatrix}$$

erstellt, wobei T die Matrixtransponierung bezeichnet.

[0039] Um jegliche Verwirrung zu vermeiden, wird betont, dass die oben beschriebenen Headerframes, Bitstromframes und Datenframes nicht mit den Übertragungsframes verwechselt werden sollten, die dann bei der Übertragung der Bitstromdaten über die Kommunikationsverbindung des Kommunikationssystems verwendet werden, in dem die Daten von einem ersten Ort an einen zweiten Ort übertragen werden, beispielsweise die Zeitmultiplex-Zeitframes ("TDMA = time division multiple access") eines zellularen GSM-Funkkommunikationssystems, welches das in den hierin beschriebenen Ausführungsformen verwendete Kommunikationssystem ist. Im vorliegenden Beispiel besteht der erste Ort aus einer Fernbenutzerstation und der zweite, d. h. Empfangsort, besteht aus einer zentralen Verarbeitungsstation, die beispielsweise bei einer Basisstation des zellularen Kommunikationssystems lokalisiert sein kann. Folglich werden in den hier beschriebenen Ausführungsformen die Spracherkennungsparameter vom ersten Ort an den zweiten Ort über eine Funkkommunikationsverbindung übertragen. Es ist jedoch erwünscht, dass die Eigenschaften des ersten Ortes und des zweiten Ortes von der Art des in Erwägung ziehenden Kommunikationssystems und der Anordnung des Prozesses zur verteilten Spracherkennung darin abhängen.

[0040] Die Headerframes und Datenframes werden aus ihrem Übertragungsformat am zweiten Ort wiederhergestellt, nachdem sie dorthin übertragen wurden. Danach werden die in den Headerframes und Datenframes enthaltenen Daten unter Verwendung der entsprechenden Headerframeinformationen verarbeitet. Eine Art, die entsprechenden Headerframeinformationen zu verwenden, besteht darin, dass die darin enthaltenen Informationen, welche Abtastrate eingesetzt wurde, verwendet werden. Das Verarbeiten der Daten enthält das Erlangen der Spracherkennungsparameter aus den Bitstromframes sowie das Ausführen der Spracherkennung selbst. Alternativ kann das Verarbeiten das Durchführen von Prozeduren enthalten, die von geeigneten Arten der framespezifischen Headerinformationen abgeleitet wurden. In einem oben beschriebenen Beispiel stellt die framespezifische Headerinformation eigentlich einen der Sprache parallelen Kanal dar, beispielsweise ein Signal für einen Tastendruck, und folglich kann die Verarbeitung das geeignete Antworten in wohlbekannter Art und Weise auf ein derartiges Signal enthalten.

[0041] Das Erlangen der Spracherkennungsparameter aus den Bitstromframes, wie im obigen Absatz erwähnt, wird wie folgt durchgeführt. Die Spracherkennungsparameter werden aus den Bitstromframes durch Ausführen einer umgekehrten Version der oben beschriebenen Vektorquantisierungsprozedur erlangt. Speziell werden aus dem Bitstrom Indizes extrahiert, und unter Verwendung dieser Indizes werden Vektoren in der Form

$$\begin{bmatrix} \hat{y}_i(m) \\ \hat{y}_{i+1}(m) \end{bmatrix} = q_{idx^{i,i+1}(m)}^{i,i+1} \quad i = 0, 2, 4, \dots, 12$$

wiederhergestellt. Ebenso muss, da in den unten stehenden beispielhaften Ausführungsformen die Headerinformation mit einer Vorwärtsfehlerkorrektur unter Verwendung eines Codierschemas auf Blockbasis codiert ist, eine derartige Fehlerkorrektur decodiert werden, nachdem die Daten am zweiten Ort empfangen wurden. Im

vorliegenden Golay-Fehlersicherungsfall kann ein derartiges Codieren auf eine von mehreren verschiedenen, dem Fachmann wohlbekannten Weisen durchgeführt werden.

[0042] Folglich ist oben ein verteilter Spracherkennungsprozess beschrieben, in dem Daten in einer Framestruktur angeordnet werden, die eine Mehrzahl von Datenframes umfasst, denen entsprechende Headerframes vorangehen, die Headerinformationen umfassen, die allgemeine Headerinformationen enthalten, wobei besagte Daten an einem zweiten Ort empfangen werden, nachdem sie von einem ersten Ort übertragen wurden, und besagte Daten werden unter der Verwendung entsprechender Headerframeinformationen verarbeitet, und in denen die Headerinformation unter Verwendung eines Codierschemas auf Blockbasis mit einer Vorwärtsfehlerkorrektur codiert ist.

[0043] Das Verfahren zum Verarbeiten empfangener Daten in einem derartigen Spracherkennungsprozess gemäß einer ersten Ausführungsform ist im Prozessflussdiagramm **400** der [Fig. 4](#) abgebildet. Bezugnehmend auf [Fig. 4](#) zeigt der Funktionskasten **410** den Schritt des Vergleichens einer empfangenen Form besagter gemeinsamer Headerinformationen von jedem einer Mehrzahl von Headerframes. In der vorliegenden Ausführungsform besteht dies aus dem Vergleichen der empfangenen Form gemeinsamer Headerinformationen vom Headerframe **240**, Headerframe **310**, Headerframe **330** und den folgenden Headerframes. Die zu vergleichenden gemeinsamen Headerframeinformationen sind in der vorliegenden Ausführungsform der angezeigte Wert der Abtastrate und ob die Spur eine Standard- oder eine rauschfeste Spur ist.

[0044] Der nächste Schritt besteht im Klassifizieren, wenn eine vorherbestimmte Anzahl besagter Headerframes gefunden wurde, die die gleiche empfangene Form der besagten gemeinsamen Headerinformation besitzt, der gleichen empfangenen Form als eine Referenz, wie im Funktionskasten **420** der [Fig. 4](#) abgebildet. In der vorliegenden Ausführungsform ist die vorherbestimmte Anzahl 30, aber diese Zahl wird man im Allgemeinen gemäß den Anforderungen des speziellen in der Erwägung stehenden Systems wählen. Durch das Ausführen dieses Verfahrensschritts wird über die Werte in den gemeinsamen Headerinformationen, die verwendet werden sollen, Sicherheit erlangt. Die Verzögerung, die durch das Ausführen dieser Prozedur eingeführt wird, insbesondere wenn die vorherbestimmte Anzahl auf ein wesentlich höheres Niveau eingestellt ist, wird in der vorliegenden Erfindung angepasst, da sie von den Eigenschaften eines Systems zur verteilten Spracherkennung Gebrauch macht, in dem Latenzzeiten am Beginn einer Nachricht relativ gut akzeptierbar sind.

[0045] Der nächste Schritt besteht darin, einen oder mehrere empfangene Datenframes zu verarbeiten, die entsprechenden Headerframes entsprechen, deren empfangene Form der gemeinsamen Headerinformation sich von besagter Referenz unterscheidet, indem die Referenzform anstatt der empfangenen unterschiedlichen Form verwendet wird, wie im Funktionskasten **430** der [Fig. 4](#) abgebildet. Folglich werden falsch empfangene oder nicht vertrauenswürdige Versionen der gemeinsamen Headerinformation ausgeschlossen und stattdessen die relevanten Datenframes bevorzugt gemäß den richtigen gemeinsamen Headerinformationen verarbeitet.

[0046] Eine Version der vorliegenden Ausführungsform enthält die Option, mit der Headerinformationen weiterhin framespezifische Headerinformationen umfassen, insbesondere die framespezifischen Headerinformationen, die eigentlich einen Kanal parallel zur Sprache darstellen, nämlich ein Signal eines Tastendrucks. Ein normales Verarbeiten enthält das geeignete Antworten auf dieses Signal in einer wohlbekannten Art und Weise. In dieser Version der vorliegenden Ausführungsform jedoch sind die framespezifischen Headerinformationen eines oder mehrerer Headerframes, deren empfangene Form der gemeinsame Headerinformationen sich von der besagten Referenz unterscheidet, vom Gebrauch ausgeschlossen, d. h. die Antwort auf das Tastendrucksignal ist blockiert. Indem dies durchgeführt wird, wird eine nicht vertrauenswürdige Instruktion vermieden, was auf dem Aspekt basiert, dass wenn die gemeinsame Headerinformation dieses Headerframes falsch ist, dann damit eine erhöhte Wahrscheinlichkeit einhergeht, dass die framespezifische Information innerhalb des gleichen Headerframes ebenso falsch ist.

[0047] Eine andere Version der vorliegenden Ausführungsform enthält weiterhin die Option, dass die Headerinformation weiterhin framespezifische Headerinformationen umfasst, insbesondere enthält es ein Feld, das dazu benutzt wird, einen Nachrichtenende-Indikator eines ersten Nachrichtenende-Erkennungsverfahrens anzuzeigen, wenn es angebracht ist. Das erste Nachrichtenende-Erkennungsverfahren ist, wie es vorhergehend weiter oben beschrieben wurde. Unter Bezugnahme auf das weiter oben beschriebene beispielhafte Headerfeldlayout und unter Bezugnahme auf Tabellen 2 und 3 sei es vermerkt, dass das Nachrichtenende-Indikatorfeld einen oder mehrere der zukünftigen Erweiterungsplätze, die in diesen Tabellen verfügbar sind, verwendet. Eine normale Verarbeitung enthält das Antworten auf diesen Nachrichtenende-Indikator, indem eine Nachrichtenende-Prozedur ausgeführt wird. In dieser Version der vorliegenden Ausführungsform jedoch wird die frame-

spezifische Headerinformation eines oder mehrerer Headerframes, deren empfangene Form der gemeinsamen Headerinformation sich von der besagten Referenz unterscheidet, und zwar der Nachrichtenende-Indikator des ersten Nachrichtenende-Erkennungsverfahrens, in einer anderen Weise verwendet als die framespezifischen Headerinformationen derjenigen Headerframes, deren empfangene Form der gemeinsamen Headerinformation die gleiche ist wie die besagte Referenz. Insbesondere wird als Antwort darauf, welchen Nennwert der Nachrichtenende-Indikator aufweist, anstatt tatsächlich die Nachrichtenende-Prozedur auszuführen, stattdessen eine zweite Nachrichtenende-Prozedur angesteuert. Somit wird zum einen eine möglicherweise falsche Ausführung der Nachrichtenende-Prozedur aufgrund des Vermeidens der Notwendigkeit vermieden, dem Indikator zu antworten, wenn der Indikator sich evtl. bereits in einem Fehlerzustand befindet, nachdem er in einen Headerframe enthalten ist, der bereits falsche gemeinsame Informationen besitzt, zum anderen wurde eine alternative Nachrichtenende-Prozedur im Sinne einer Sicherung aktiviert, um sicherzustellen, dass das Nachrichtenende tatsächlich detektiert wird, wenn die Nachrichtenende-Anzeige tatsächlich richtig war, obwohl sie in einem teilweise falschen Headerframe enthalten war. Man sollte sich bewusst sein, dass die oben beschriebene Nachrichtenende-Thematik lediglich ein Beispiel der framespezifischen Informationstypen repräsentiert, die gemäß der Erfindung und gemäß den Diskrepanzen in den gemeinsamen Headerinformationen unterschiedlich behandelt werden können. Allerdings ist dieser Aspekt der Erfindung im Gegensatz dazu auf beliebige framespezifische Informationsarten anwendbar, die von einer vorsichtigen Behandlung profitieren, wenn es Verdachtsmomente gibt, dass die Information eine höhere Wahrscheinlichkeit eines Fehlers in Headerframes aufweist, die bereits das Vorhandensein eines Fehlers in den gemeinsamen Informationen, die Teil desselben sind, gezeigt haben.

[0048] Alle oben beschriebenen Ausführungsformen können die optionale Eigenschaft aufweisen, dass die besagte Headerinformation mit einer Vorwärtsfehlerkorrektur codiert ist. unter Verwendung eines Codierschemas auf Blockbasis. Die Details, wie ein derartiges Schema ausgeführt wird, sind bereits oben dargelegt worden. Ist diese Eigenschaft enthalten, werden die Vorteile der vorliegenden Erfindung insbesondere aufgrund der Verbundeigenschaften derartiger Fehlerkorrekturarten verstärkt. Nachdem der gesamte Headerframe in einer Verbundweise korrigiert wird, wenn sich die gemeinsame Headerinformation von dem unterscheidet, was empfangen werden sollte, wird höchstwahrscheinlich der Rest der Headerinformation im Wesentlichen ebenfalls fehlerhaft sein. Im vorliegenden Beispiel tritt dies insbesondere wahrscheinlich dann ein, wenn mehr als drei Bits in einem Headerframe fehlerhaft sind.

[0049] In den bisherigen Ausführungsformen werden die Datenframes, die empfangen wurden, bevor die Referenz bestimmt ist, einfach von der späteren Verarbeitungsphase ausgeschlossen. Im Sinne eines Ausgleichs bietet dies die Vorteile eines reduzierten Verarbeitens. In alternativen Ausführungsformen jedoch werden die Datenframes, die vor der Bestimmung der besagten Referenz empfangen wurden, gepuffert, bevor sie nach der Bestimmung der besagten Referenz verarbeitet werden. Im Sinne eines Ausgleichs bieten diese alternativen Ausführungsformen eine bessere Qualität in dem Sinne, dass die Daten nicht verloren gehen, obwohl zusätzliches Verarbeiten einschließlich zusätzlicher Pufferung benötigt wird. Diese alternative Ausführungsformen stellen darüber hinaus einen zusätzlich vorteilhaften Gebrauch der Latenzzeitcharakteristiken eines verteilten Spracherkennungsprozesses zur Verfügung, insbesondere den Aspekt, dass die Latenzzeit am Beginn der Nachricht relativ gut toleriert wird.

[0050] Im Fall der oben beschriebenen Ausführungsform werden die beschriebenen Datenverarbeitungsschritte von einem programmierbaren Digitalsignalverarbeitungsbauelement, wie etwa einem aus der DSP 56xxx-Bauelementefamilie (Warenzeichen) von Motorola durchgeführt. Alternativ kann eine anwendungsspezifische integrierte Schaltung ("ASIC = application specific integrated circuit") verwendet werden. Es gibt weitere Möglichkeiten. Beispielsweise kann eine Schnittstelleneinheit verwendet werden, die zwischen einem Funkempfänger und einem Computersystem vermittelt, das einen Teil eines Backend-Spracherkennungsprozessors bildet.

Patentansprüche

1. Verfahren zum Verarbeiten empfangener Daten in einem verteilten Spracherkennungsprozess, wobei der verteilte Spracherkennungsprozess einer ist, bei dem Daten in einer Framestruktur angeordnet sind, die eine Mehrzahl von Datenframes (**230, 320, 340**) umfasst, denen jeweils ein entsprechender Headerframe (**240, 310, 330**) vorangeht, der Headerinformation einschließlich gemeinsamer Headerinformation umfasst, wobei die von einem ersten Ort gesendeten Daten an einem zweiten Ort empfangen werden und die Daten unter Verwendung entsprechender Headerframeinformation verarbeitet werden; wobei das Verfahren **dadurch gekennzeichnet** ist, dass es die Schritte umfasst: Vergleichen (**410**) einer empfangenen Form der gemeinsamen Headerinformation von jedem einer Mehrzahl

von Headerframes (**240, 310, 330**);

Klassifizieren (**420**), wenn festgestellt wird, dass eine vorherbestimmte Anzahl von Headerframes (**240, 310, 330**) die gleiche empfangene Form der gemeinsamen Headerinformation aufweist, der gleichen empfangenen Form als eine Referenz; und

Verarbeiten (**430**) eines oder mehrerer empfangener Datenframes (**230, 320, 340**), die den entsprechenden Headerframes (**240, 310, 330**) entsprechen, deren empfangene Form der gemeinsamen Headerinformation sich von der Referenz unterscheidet, indem die Referenzform anstatt der empfangenen unterschiedlichen Form verwendet wird.

2. Verfahren nach Anspruch 1, wobei die Headerinformation weiterhin framespezifische Headerinformation umfasst und wobei die framespezifische Headerinformation des einen oder der mehreren Headerframes (**240, 310, 330**), deren empfangene Form der gemeinsamen Headerinformationen sich von der Referenz unterscheidet, vom Gebrauch ausgeschlossen wird.

3. Verfahren nach Anspruch 1, wobei die Headerinformation weiterhin framespezifische Headerinformation umfasst und wobei die framespezifische Headerinformation des einen oder der mehreren Headerframes (**240, 310, 330**), deren empfangene Form der gemeinsamen Headerinformation sich von der Referenz unterscheidet, auf eine andere Weise verwendet wird, verglichen mit der framespezifischen Headerinformation derjenigen Headerframes, deren empfangene Form der gemeinsamen Headerinformation die gleiche ist wie die Referenz.

4. Verfahren nach Anspruch 3, wobei die framespezifische Information, die auf eine andere Weise verwendet wird, ein Nachrichtenende-Indikator eines ersten Nachrichtenende-Erkennungsverfahrens ist und als Antwort darauf ein zweites Nachrichtenende-Erkennungsverfahren ausgelöst wird.

5. Verfahren nach einem der vorhergehenden Ansprüche, wobei die Headerinformation vorwärtsfehlerkorrekturcodiert ist, unter Verwendung eines Codierschemas auf Blockbasis.

6. Verfahren nach einem der vorhergehenden Ansprüche, wobei Datenframes, die vor dem Bestimmen der Referenz empfangen werden, gepuffert werden, bevor sie verarbeitet werden, nachdem die Referenz bestimmt wurde.

7. Vorrichtung zum Verarbeiten empfangener Daten in einem verteilten Spracherkennungsprozess, wobei der verteilte Spracherkennungsprozess einer ist, bei dem Daten in einer Framestruktur angeordnet sind, die eine Mehrzahl von Datenframes (**230, 320, 340**) umfasst, denen jeweils ein entsprechender Headerframe (**240, 310, 330**) vorangeht, der Headerinformation einschließlich gemeinsamer Headerinformation umfasst, wobei die von einem ersten Ort gesendeten Daten an einem zweiten Ort empfangen werden und die Daten unter Verwendung entsprechender Headerframeinformation verarbeitet werden;

wobei die Vorrichtung dadurch gekennzeichnet ist, dass sie umfasst

Mittel zum Vergleichen einer empfangenen Form der gemeinsamen Headerinformation von jedem einer Mehrzahl von Headerframes (**240, 310, 330**);

Mittel zum Klassifizieren, wenn festgestellt wird, dass eine vorherbestimmte Anzahl von Headerframes (**240, 310, 330**) die gleiche empfangene Form der gemeinsamen Headerinformation aufweist, der gleichen empfangenen Form als eine Referenz; und

Mittel zum Verarbeiten eines oder mehrerer empfangener Datenframes (**230, 320, 340**), die den entsprechenden Headerframes (**240, 310, 330**) entsprechen, deren empfangene Form der gemeinsamen Headerinformation sich von der Referenz unterscheidet, indem die Referenzform anstatt der empfangenen unterschiedlichen Form verwendet wird.

8. Vorrichtung nach Anspruch 7, wobei die Headerinformation weiterhin framespezifische Headerinformation umfasst und wobei die framespezifische Headerinformation des einen oder der mehreren Headerframes (**240, 310, 330**), deren empfangene Form der gemeinsamen Headerinformation sich von der Referenz unterscheidet, vom Gebrauch ausgeschlossen ist.

9. Vorrichtung nach Anspruch 7, wobei die Headerinformation weiterhin framespezifische Headerinformation umfasst und wobei die framespezifische Headerinformation des einen oder der mehreren Headerframes (**240, 310, 330**), deren empfangene Form der gemeinsamen Headerinformation sich von der Referenz unterscheidet, auf eine andere Weise verwendet wird, verglichen mit der framespezifischen Headerinformation derjenigen Headerframes, deren empfangene Form der gemeinsamen Headerinformation die gleiche ist wie die Referenz.

10. Vorrichtung nach Anspruch 9, wobei die framespezifische Information, die auf eine andere Weise verwendet wird, ein Nachrichtenende-Indikator eines ersten Nachrichtenende-Erkennungsverfahrens ist und als Antwort darauf ein zweites Nachrichtenende-Erkennungsverfahren ausgelöst wird.

11. Vorrichtung nach einem der Ansprüche 7 bis 10, wobei die Headerinformation vorwärtsfehlerkorrekturcodiert ist, unter Verwendung eines Codierschemas auf Blockbasis.

12. Vorrichtung nach einem der Ansprüche 7 bis 11, wobei die Datenframes, die vor dem Bestimmen der Referenz empfangen werden, gepuffert werden, bevor sie verarbeitet werden, nachdem die Referenz bestimmt wurde.

Es folgen 2 Blatt Zeichnungen

Anhängende Zeichnungen

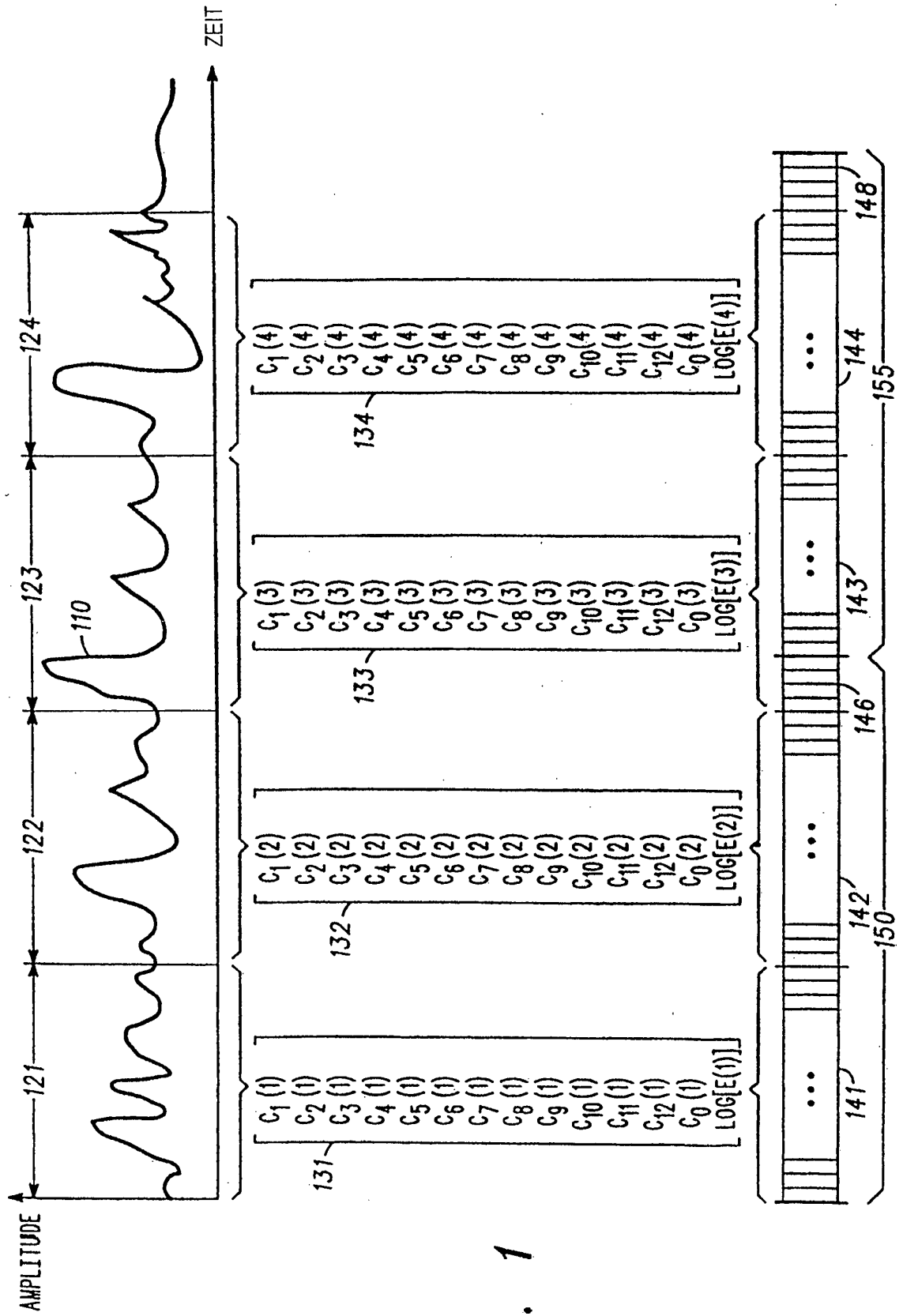


FIG. 1

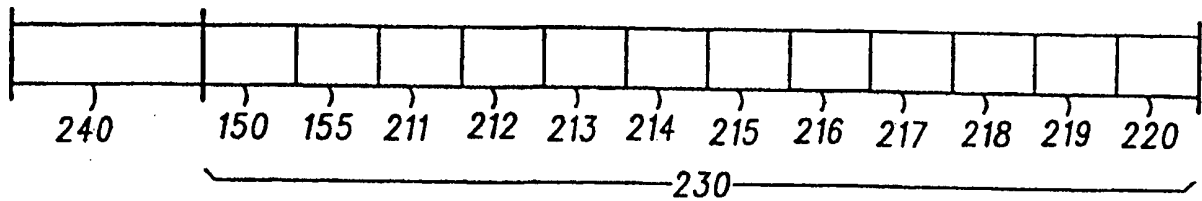


FIG. 2

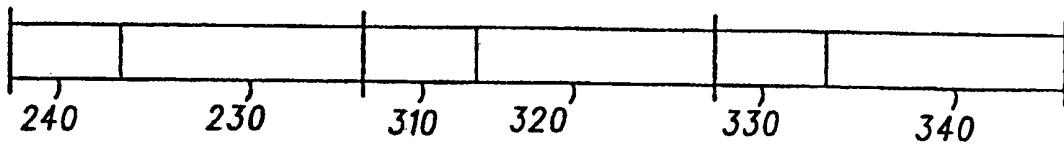
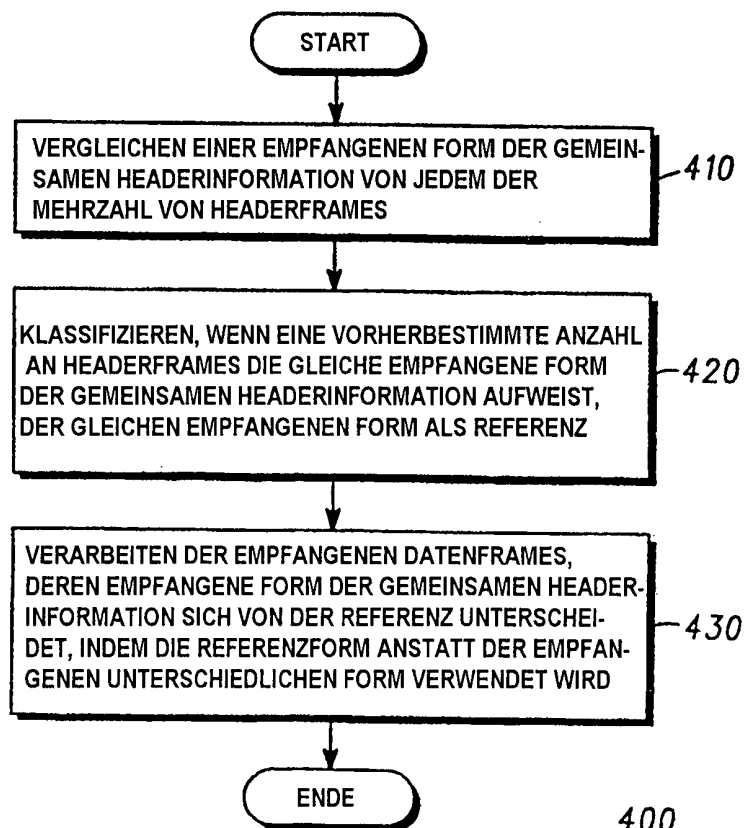


FIG. 3



400
FIG. 4