



(19)
Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) **DE 697 35 679 T2** 2006.11.23

(12)

Übersetzung der europäischen Patentschrift

(97) **EP 1 100 273 B1**

(21) Deutsches Aktenzeichen: **697 35 679.5**

(96) Europäisches Aktenzeichen: **01 101 680.5**

(96) Europäischer Anmeldetag: **06.11.1997**

(97) Erstveröffentlichung durch das EPA: **16.05.2001**

(97) Veröffentlichungstag

der Patenterteilung beim EPA: **12.04.2006**

(47) Veröffentlichungstag im Patentblatt: **23.11.2006**

(51) Int Cl.⁸: **H04N 7/50** (2006.01)

H03M 7/42 (2006.01)

H04N 7/30 (2006.01)

(30) Unionspriorität:

29351296 **06.11.1996** **JP**

6264997 **17.03.1997** **JP**

(73) Patentinhaber:

**Matsushita Electric Industrial Co., Ltd., Kadoma,
Osaka, JP**

(74) Vertreter:

Eisenführ, Speiser & Partner, 28195 Bremen

(84) Benannte Vertragsstaaten:

DE, FR, GB, NL

(72) Erfinder:

**Boon, Chong Seng, Moriguchi-shi, Osaka
570-0014, JP**

(54) Bezeichnung: **Verfahren zur Bilddecodierung**

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 zur Bilddecodierung.

Stand der Technik

[0002] Komprimierte Codierung ist zum effizienten Speichern oder Übertragen eines digitalen Bilds unbedingt erforderlich. Ein Verfahren, das zur komprimierten Codierung eines digitalen Bilds verwendet wird, ist die diskrete Kosinus-Transformation (DCT), die eine herausragende Technologie bei typischen Standards wie JPEG (Joint Photographic Experts Group) und MPEG (Moving Picture Experts Group) ist. Darüber hinaus kommen auch Wellenformcodierungsverfahren wie Codieren mit unterteilten Bändern, Wavelet-Transformations- und Fraktalcodierung als Codierungsverfahren zur Anwendung. Um redundante Signale zwischen Bildern zu entfernen, wird hauptsächlich eine Intrabild-Codierung aufgrund einer räumlichen Korrelation in einem Stehbild durchgeführt; außerdem wird auch eine Interbild-Codierung durchgeführt, bei der eine Interbild-Voraussage durch Bewegungskompensation aufgrund einer zeitlichen Korrelation zwischen den einzelnen Gruppen von benachbarten Stehbildern erfolgt und ein erhaltenes Differenzsignal der Wellenformcodierung unterzogen wird. Mit diesen beiden Verfahren können codierte Daten mit einer hohen Komprimierungsrate erhalten werden.

[0003] Bei der herkömmlichen Bildcodierung beispielsweise nach dem MPEG-Standard, wird nach dem DCT-Verfahren o. Ä. eine Quantisierung durchgeführt, an die sich eine Codierung mit variabler Wortlänge anschließt. Insbesondere wird ein Verfahren wie DCT, Codieren mit unterteilten Bändern oder Wavelet an digitalen Bilddaten durchgeführt und die mit dem vorgenannten Verfahren erhaltenen Transformationskoeffizienten werden unter Verwendung einer vorgeschriebenen Quantisierungsskala quantisiert, um quantisierte Transformationskoeffizienten zu erzeugen, an denen dann die Codierung mit variabler Wortlänge vorzunehmen ist.

[0004] Die durch Quantisierung erhaltenen quantisierten Transformationskoeffizienten werden in einem zweidimensionalen Array angeordnet und jeder Koeffizient bezeichnet Null oder ungleich Null. Beim Codieren mit variabler Wortlänge werden die Koeffizienten (Null und ungleich Null), die in dem zweidimensionalen Array in einer vorgegebenen Abtast-Reihenfolge angeordnet sind, zickzack-gescannt, um die Koeffizienten in eine Dimension umzugruppieren, und anschließend werden Ereignisse aus dem eindimensionalen Koeffizienten-Array erzeugt. Jedes Ereignis besteht aus einem Run, der

eine Anzahl vorhergehender Null-Koeffizienten bezeichnet, und einem Level, der den Wert eines Koeffizienten ungleich Null angibt. Dann wird die Codierung durch Bezugnahme auf eine vorher erstellte Tabelle von Codes mit variabler Wortlänge durchgeführt, in der ein eindeutiger Code mit variabler Wortlänge jedem aus einem Run und einem Level bestehenden Ereignis zugeordnet wird. Anstelle dieses Verfahrens kann auch ein anderes Verfahren verwendet werden, beispielsweise das Verfahren der Zuordnung eines weiteren Codes, der angibt, ob ein Koeffizient ein letzter Koeffizient ist, oder ein Verfahren zur Codierung mit Verweis auf eine Tabelle, wo Codes mit variabler Wortlänge Ereignissen zugeordnet werden, die jeweils aus einer Gruppe (Run, Level, Last) bestehen, die durch Addieren eines Last, der angibt, ob ein Koeffizient ein letzter Koeffizient ist, zu einem Run und einem Level erhalten wird.

[0005] [Fig. 18](#) ist ein Ablaufdiagramm, das den Ablauf eines herkömmlichen Codierungsverfahrens für ein aus (Last, Run, Level) bestehendes Ereignis darstellt. Nachstehend wird das herkömmliche Codierungsverfahren unter Bezugnahme auf das Ablaufdiagramm von [Fig. 18](#) beschrieben.

[0006] Nach der Initiierung des Prozesses im Schritt **1801** wird ein j-tes Ereignis als Eingabe-Ereignis, das ein zu verarbeitendes Objekt ist, eingegeben. Im Schritt **1803** wird ein Vergleich zwischen diesem Eingabe-Ereignis und einem in einer Tabelle von Codes mit variabler Wortlänge enthaltenen Bezugsereignis angestellt.

[0007] Die [Fig. 19](#) bis [Fig. 22](#) zeigen ein Beispiel für die Tabelle von Codes mit variabler Wortlänge, die für das herkömmliche Verfahren des Codierens mit variabler Wortlänge verwendet wird. Wie dargestellt, entsprechen in der Tabelle die jeweils aus (Last, Run, Level) bestehenden Ereignisse einzelnen Codes. In den Figuren bezeichnet „s“ in einer „VLC-CODE“-Zelle Null, wenn der Level positiv ist, während es 1 bezeichnet, wenn der Level negativ ist. Wenn Last 0 ist, gibt es an, dass der betreffende Koeffizient nicht der letzte Koeffizient ist.

[0008] Im Schritt **1803** von [Fig. 18](#) wird durch Verweis auf diese Tabelle von Codes mit variabler Wortlänge ein Vergleich zwischen dem Eingabe-Ereignis und jedem Bezugsereignis durchgeführt, das in der aus (Last, Run, Level) bestehenden Tabelle enthalten ist. Im Schritt **1804** wird aufgrund dieses Vergleichs entschieden, ob es ein Bezugsereignis gibt, dessen (Last, Run, Level) identisch mit dem des Eingabe-Ereignisses ist. Wenn im Schritt **1804** entschieden wird, dass es ein solches Bezugsereignis gibt, wird Schritt **1805** ausgeführt, während dann, wenn kein solches Bezugsereignis gefunden wird, Schritt **1806** ausgeführt wird. Wenn jetzt Schritt **1805** ausgeführt wird, wird ein Code mit variabler Wortlänge, der

dem Bezugsereignis entspricht, ausgegeben. Dann schließt sich Schritt **1807** an, der später beschrieben wird.

[0009] Wenn jedoch beim Ausführen von Schritt **1806**, das heißt, in dem Fall, dass der (Last, Run, Level), der mit dem des Ereignisses als zu verarbeitendes Objekt identisch ist, beim Nachschlagen in der Tabelle von Codes mit variabler Wortlänge nicht gefunden werden kann, wird im Schritt **1806** eine Codierung mit fester Wortlänge durchgeführt. Die **Fig. 23** sind Diagramme, die die Codierung mit fester Wortlänge begrifflich darstellen. Wie in den Figuren gezeigt, werden die in **Fig. 23(a)** gezeigten Codes Runs zugeordnet, und die in **Fig. 23(b)** gezeigten Codes werden Levels zugeordnet, und diese Codes sind die Ergebnisse der Codierung. Bei diesem Verfahren der Codierung mit fester Wortlänge wird ein Steuercode zum Anzeigen, dass ein Codierungsergebnis ein Code mit fester Wortlänge ist, so addiert, dass er vor dem Ergebnis der Codierung übertragen wird, um den durch Codierung mit fester Wortlänge erhaltenen Code von dem durch Codierung mit variabler Wortlänge erhaltenen Code zu unterscheiden. Dieser Steuercode wird als „Escape-Sequenz (ESC)“ bezeichnet, und hier wird der in **Fig. 22** angegebene Code „0000011“ als Steuercode verwendet. Insbesondere werden bei der Durchführung der Codierung mit fester Wortlänge codierte Daten erhalten, die einen 7-Bit-ESC-Code und den Code umfassen, der nach dem ESC-Code zugeordnet wird.

[0010] Nach der Ausführung von Schritt **1805** oder **1806** wird im nachfolgenden Schritt **1807** entschieden, ob ein Eingabe-Ereignis, das als letztes Eingabe-Ereignis verarbeitet wird, ein letztes Eingabe-Ereignis ist oder nicht. Wenn entschieden wird, dass das Ereignis das letzte Ereignis ist, wird die Codierung im Schritt **1808** beendet. Wenn jedoch entschieden wird, dass das Ereignis nicht das letzte Ereignis ist, wird im Schritt **1809** j um 1 erhöht, und es erfolgt ein Rücksprung zum Schritt **1802**, mit dem ein nächstes Eingabe-Ereignis in der gleichen Weise verarbeitet wird. So wird das Verfahren bis zum letzten Koeffizienten ungleich Null fortgesetzt.

[0011] Wie vorstehend beschrieben, werden bei der herkömmlichen Bildcodierung Ereignisse für quantisierte Transformationskoeffizienten erzeugt und es wird ein Vergleich zwischen dem Ereignis und einem in einer Tabelle von Codes mit variabler Wortlänge enthaltenen Bezugsereignis durchgeführt. Wenn das passende Ereignis gefunden wird, wird ein aus der Tabelle stammender Code mit variabler Wortlänge verwendet, während in dem Fall, dass kein passendes Ereignis gefunden wird, das Ergebnis der Codierung mit fester Wortlänge, zu dem ein Steuercode (ESC-Code) addiert wird, verwendet wird.

[0012] In der Regel wird eine Tabelle von Codes mit

variabler Wortlänge erstellt, indem die statistische Verteilung unter Verwendung von Zahlenbildern untersucht wird und den am häufigsten vorkommenden Ereignissen kürzere Codes zugeordnet werden, wodurch die Gesamtkomprimierungsrate verbessert wird. Wenn, wie vorstehend beschrieben, die Codierung nicht unter Verwendung einer Tabelle von Codes mit variabler Wortlänge durchgeführt wird, wird eine Codierung mit fester Wortlänge durchgeführt und dadurch die Komprimierungsrate verschlechtert. Es ist daher zweckmäßig, die Festlegung so vorzunehmen, dass es kaum zur Codierung mit fester Wortlänge kommt. Wenn bei der in **Fig. 18** gezeigten herkömmlichen Bildcodierung im Schritt **1804** entschieden wird, dass es kein betreffendes Bezugsereignis gibt, wird im Schritt **1806** sofort die Codierung mit fester Wortlänge durchgeführt, was bedeutet, dass der Anteil der Durchführung der Codierung mit variabler Wortlänge mit einer besseren Codierungsleistung direkt von dem Anteil bestimmt wird, in dem ein betreffendes Ereignis in einer Tabelle von Codes mit variabler Wortlänge vorliegt.

[0013] Bei der Bild-Kompressionscodierung ändert sich jedoch die statistische Verteilung von Koeffizienten mit der Quantisierungsskala, und es ist klar, dass die Verteilung von Koeffizienten mit einer hohen Komprimierungsrate erheblich von der mit einer niedrigen Komprimierungsrate abweicht. Wenn also keine entsprechende Tabelle von Codes mit variabler Wortlänge für Koeffizienten als Objekte verwendet wird, verringert sich der Anteil, in dem das betreffende Ereignis in der Tabelle von Codes mit variabler Wortlänge vorliegt, und der Anteil der Codierung mit fester Wortlänge erhöht sich, was eine niedrigere Codierungsleistung zur Folge hat.

[0014] Im Allgemeinen nimmt bei einer Erhöhung der Quantisierungsskala die Komprimierungsrate von quantisierten Transformationskoeffizienten zu, während das von diesen quantisierten Transformationskoeffizienten abgeleitete Ereignis meist einen niedrigeren Level-Wert hat. Die in den **Fig. 19** bis **Fig. 22** gezeigte Tabelle von Codes mit variabler Wortlänge enthält zahlreiche Ereignisse mit niedrigen Level-Werten, denen kurze Codes zugeordnet werden, und sie ist zur Verwendung in dem Fall geeignet, dass quantisierte Transformationskoeffizienten mit einer relativ hohen Komprimierungsrate als Verarbeitungsobjekte verwendet werden. Wenn also die Quantisierungsskala klein ist und die Komprimierungsrate der quantisierten Transformationskoeffizienten niedrig ist, haben viele der abgeleiteten Ereignisse große Level-Werte. So erhöht sich der Anteil, in dem ein betreffendes Ereignis nicht in der Tabelle von Codes mit variabler Wortlänge vorliegt, und den in der Tabelle vorliegenden Ereignissen werden oft relativ lange Codes zugeordnet, was zu einer erhöhten Bitzahl des Codierungsergebnisses führt. Mit anderen Worten, die Verwendung einer Tabelle von Codes

mit variabler Wortlänge, die für eine niedrige Komprimierungsrate bei der Codierung mit hoher Komprimierungsrate ausgelegt ist, hat eine Abnahme der Codierungsleistung zur Folge.

[0015] Wenn, wie vorstehend beschrieben, keine entsprechend erstellte Tabelle von Codes mit variabler Wortlänge für die Koeffizienten, die die Verarbeitungsobjekte der Codierung mit variabler Wortlänge sind, verwendet wird, nimmt der Umfang des Codes zu und es wird keine Verbesserung der Komprimierungsrate erzielt (was ein Problem der herkömmlichen Bildcodierung ist), da keine Codierung mit variabler Wortlänge durchgeführt werden kann und die Codierung mit fester Wortlänge häufiger durchgeführt wird und die Anzahl der Bits auch dann zunimmt, wenn die Codierung mit variabler Wortlänge durchgeführt wird.

[0016] Ein bekanntes Bilddecodierungsverfahren zum Gewinnen eines Codes mit variabler Wortlänge ist im Dokument EP-A-0.595.242 dargelegt, das einen vielseitigen Escape-Run-Level-Codierer zur Verwendung in einer digitalen Videosignal-Verarbeitungsvorrichtung beschreibt, die Run-Level-Daten und eine Escape-Präfix-Sequenz tragende Escape-Sequenz und Escape-Sequenzlängendaten erzeugt. Eine Auswahllogik (28) stellt den Level der Level-Daten in den Run-Level-Daten fest. Wenn festgestellt wird, dass der Level im mittleren Bereich liegt, wählen drei Selektoren (8, 16, 21) Daten mit Unterlänge, Daten vom Code-Erweiterer 13, der einen sinnlosen 8-Bit-Code einfügt, bzw. Daten vom Code-Verketter 1. Die gewählten Daten werden in einem zweiten Code-Verketter (30) verkettet, um einen maßgeschneiderten Code mit einer Länge von 36 Bit zu erzeugen, wobei die effektive Datenlänge jedoch z. B. 20 Bit ist. Wenn festgestellt wird, dass der Level über oder unter dem mittleren Bereich liegt, wählen die drei Selektoren Daten mit Überlänge, Daten aus der Leitung (14), die die Escape-Sequenz überträgt, bzw. Daten aus einem der beiden Level-Codierer (23, 24). Die gewählten Daten werden im zweiten Code-Verketter (30) verkettet, um einen maßgeschneiderten Code mit einer Länge von 36 Bit zu erzeugen, wobei die effektive Datenlänge jedoch z. B. 28 Bit ist.

Kurze Darstellung der Erfindung

[0017] In Anbetracht der vorstehend beschriebenen Probleme wird die vorliegende Erfindung vorgeschlagen.

[0018] Die vorliegende Erfindung betrifft also ein Bilddecodierungsverfahren zum Gewinnen eines Codes mit variabler Wortlänge aus kompressionscodierten Daten, das in den beigefügten Ansprüchen definiert ist.

Kurze Beschreibung der Zeichnungen

[0019] [Fig. 1](#) ist ein Ablaufdiagramm, das den Ablauf des Verfahrens zur Codierung mit variabler Wortlänge nach einer ersten Ausführungsform, die dem Verständnis der vorliegenden Erfindung dient, darstellt.

[0020] [Fig. 2](#) ist Blockdiagramm, das die Konfiguration einer Bildcodierungsvorrichtung nach dieser Ausführungsform zeigt.

[0021] Die [Fig. 3](#) sind erläuternde Diagramme, die Beispiele für einen Code zeigen, der nach dem Verfahren zur Codierung mit variabler Wortlänge nach dieser Ausführungsform transformiert wird.

[0022] [Fig. 4](#) ist ein Blockdiagramm, das die Konfiguration einer Bilddecodierungsvorrichtung nach dieser Ausführungsform zeigt.

[0023] [Fig. 5](#) ist ein Ablaufdiagramm, das den Ablauf eines Verfahrens zur Decodierung mit variabler Wortlänge nach dieser Ausführungsform darstellt.

[0024] [Fig. 6](#) ist ein Ablaufdiagramm, das den Ablauf eines Verfahrens zur Codierung mit variabler Wortlänge nach einer zweiten Ausführungsform, die dem Verständnis der vorliegenden Erfindung dient, darstellt.

[0025] [Fig. 7](#) ist ein Ablaufdiagramm, das den Ablauf eines Verfahrens zur Decodierung mit variabler Wortlänge nach dieser Ausführungsform darstellt.

[0026] Die [Fig. 8](#) sind Ablaufdiagramme, die den Ablauf eines Verfahrens zur Codierung mit variabler Wortlänge nach einer dritten Ausführungsform der vorliegenden Erfindung darstellen.

[0027] [Fig. 9](#) ist ein Ablaufdiagramm, das den Ablauf eines Verfahrens zur Decodierung mit variabler Wortlänge nach dieser Ausführungsform darstellt.

[0028] [Fig. 10](#) ist ein Ablaufdiagramm, das den Ablauf eines Verfahrens zur Codierung mit variabler Wortlänge nach einer vierten Ausführungsform darstellt.

[0029] [Fig. 11](#) ist ein Ablaufdiagramm, das den Ablauf eines Verfahrens zur Decodierung mit variabler Wortlänge nach dieser Ausführungsform darstellt.

[0030] [Fig. 12](#) ist ein Ablaufdiagramm, das den Ablauf eines Verfahrens zur Codierung mit variabler Wortlänge nach einer fünften Ausführungsform darstellt.

[0031] Die [Fig. 13](#) sind Diagramme, die den Zusammenhang zwischen Quantisierungsskalen und Divi-

soren, die für den Transformationsprozess in dem Verfahren zur Codierung mit variabler Wortlänge nach dieser Ausführungsform zeigen.

[0032] [Fig. 14](#) ist ein Ablaufdiagramm, das den Ablauf eines Verfahrens zur Decodierung mit variabler Wortlänge nach dieser Ausführungsform darstellt.

[0033] [Fig. 15](#) ist ein Ablaufdiagramm, das den Ablauf eines Verfahrens zur Codierung mit variabler Wortlänge nach einer sechsten Ausführungsform darstellt.

[0034] [Fig. 16](#) ist ein Blockdiagramm, das die innere Konfiguration eines Codierers für Codes mit variabler Wortlänge zeigt, der in der Bildcodierungsvorrichtung nach dieser Ausführungsform enthalten ist.

[0035] [Fig. 17](#) ist ein Blockdiagramm, das die innere Konfiguration eines Decodierers für Codes mit variabler Wortlänge zeigt, der in der Bilddecodierungsvorrichtung nach dieser Ausführungsform enthalten ist.

[0036] [Fig. 18](#) ist ein Ablaufdiagramm, das den Ablauf eines herkömmlichen Verfahrens zur Codierung mit variabler Wortlänge darstellt.

[0037] Die [Fig. 19](#) bis [Fig. 22](#) sind Diagramme, die Beispiele für eine Tabelle von Codes mit variabler Wortlänge zeigen, die zur Codierung mit variabler Wortlänge verwendet wird.

[0038] Die [Fig. 23](#) sind Darstellungen zur Erläuterung des Verfahrens zur Codierung mit fester Wortlänge.

Detaillierte Beschreibung der bevorzugten Ausführungsformen

Ausführungsform 1

[0039] Ein Bildcodierungsverfahren gemäß einer ersten Ausführungsform soll die Möglichkeit der Verwendung einer Tabelle von Codes mit variabler Wortlänge durch Ausführung der Transformation an Ereignissen, die zu codierende Objekte sind, erhöhen und dadurch die Codierungsleistung verbessern.

[0040] [Fig. 1](#) ist ein Ablaufdiagramm, das den Ablauf eines Verfahrens zur Codierung mit variabler Wortlänge gemäß der ersten Ausführungsform der vorliegenden Erfindung darstellt. [Fig. 2](#) ist ein Blockdiagramm, das die Konfiguration einer Bildcodierungsvorrichtung zeigt, die die Codierung mit variabler Wortlänge gemäß der ersten Ausführungsform durchführt, und die [Fig. 3](#) sind Diagramme, die Beispiele für das Ergebnis der Codierung mit variabler Wortlänge zeigen.

[0041] Zunächst wird die Bildcodierungsvorrichtung gemäß der ersten Ausführungsform anhand von [Fig. 2](#) beschrieben. Wie in der Figur gezeigt, weist die Bildcodierungsvorrichtung gemäß der ersten Ausführungsform einen ersten Addierer **202**, einen Codierer **203**, einen zweiten Addierer **210**, einen Codierer für Codes mit variabler Wortlänge (VLC) **211**, einen Bildspeicher (FM) **213**, einen Bewegungsextraktor (ME) **214** und einen Bewegungskompensator (MC) **215** auf und empfängt von einem Eingabeterminal **201** digitale Bilddaten, die Gegenstand der Codierung sind, und gibt von einem Ausgabeterminal **206** die codierten Daten aus, die die Ausgangsdaten der Bildcodierungsvorrichtung sind. Der Codierer **203** enthält eine diskrete Kosinus-Transformationseinheit (DCT) **204** und einen Quantisierer (Q) **205**, und der Decodierer **207** enthält einen Umkehrquantisierer (IQ) **208** und eine diskrete Umkehr-Kosinus-Transformationseinheit (IDCT) **209**.

[0042] Nachstehend wird die Funktionsweise der so aufgebauten Bildcodierungsvorrichtung beschrieben. Zunächst wird ein zu codierendes Bild, digitale Bilddaten, in das Eingabeterminal **201** eingegeben. Das zu codierende Bild wird in eine Vielzahl kleiner benachbarter Bereiche unterteilt. Bei der ersten Ausführungsform wird ein zu codierendes Bild in eine Vielzahl von benachbarten 16×16 -Blöcken unterteilt und die Codierung erfolgt für jeden einzelnen Block.

[0043] Ein Objektblock, der ein Block als zu codierendes Objekt ist, wird über eine Leitung **216** in den ersten Addierer **202** sowie über eine Leitung **225** in den Bewegungsextraktor **214** eingegeben. Der Bildspeicher **213** speichert ein bereits wiederhergestelltes Bild, was später beschrieben wird, und der Bewegungsextraktor **214** empfängt das gespeicherte Bild über eine Leitung **224** und führt mit dem Bild als Bezugsbild die Bewegungsextraktion an dem zu codierenden Objektblock aus, um einen Bewegungsvektor zu erhalten. Bei der Bewegungsextraktion werden Bewegungsverschiebungsinformationen, die ein Vorhersagesignal mit minimalem Fehler für einen zu codierenden Objektblock ergeben, durch Block-Anpassung oder ein anderes Verfahren erhalten, und die Bewegungsverschiebungsinformationen werden als Bewegungsvektor verarbeitet. Der Bewegungsextraktor **214** gibt den Bewegungsvektor an den Bewegungskompensator **215** aus. Der Bewegungsvektor wird auch über eine Leitung **228** an den Codierer für Codes mit variabler Wortlänge **211** ausgegeben, wo er in einen Code mit variabler Wortlänge transformiert wird und zum Ergebnis der Codierung als Teil der Nebeninformationen addiert wird.

[0044] Der Bewegungskompensator **215** verwendet ebenfalls das im Bildspeicher **213** gespeicherte Bild als Bezugsbild, erzeugt aus diesem Bezugsbild und dem Bewegungsvektor einen vorhergesagten Block und gibt ihn über eine Leitung **226** an den ersten Ad-

dierer **202** sowie über eine Leitung **227** an den zweiten Addierer **210** aus. Der erste Addierer **202** leitet eine Differenz zwischen dem eingegebenen Objektblock und dem eingegebenen vorhergesagten Block ab, um einen Differenzblock zu erzeugen, und gibt diesen über eine Leitung **217** an den Codierer **203** aus.

[0045] Der Differenzblock wird im Codierer **203** kompressionscodiert. Bei der ersten Ausführungsform wird der Differenzblock in die diskrete Kosinus-Transformationseinheit **204** eingegeben und in Koeffizienten in einem Frequenzbereich transformiert. Die Koeffizienten in dem Frequenzbereich werden über eine Leitung **218** in den Quantisierer **205** eingegeben, der unter Verwendung einer vorgeschriebenen Quantisierungsskala eine Quantisierung durchführt. Wenn der erste Addierer **202** keine Differenz ableitet, das heißt, wenn eine Intra-Bildcodierung erfolgt, wird der Objektblock in den Codierer **203** eingegeben, wo er in der gleichen Weise wie der Differenzblock verarbeitet wird.

[0046] Quantisierte Transformationskoeffizienten und die Quantisierungsskala, die für die Quantisierung verwendet werden, werden vom Quantisierer **205** über eine Leitung **219** an den Codierer für Codes mit variabler Wortlänge **203** ausgegeben, wo die Codierung erfolgt, und das Ergebnis der Codierung wird zusammen mit den Nebeninformationen, die den Bewegungsvektor und die Quantisierungsskala umfassen, an das Ausgabeterminal **206** ausgegeben. Gleichzeitig werden die quantisierten Transformationskoeffizienten an den Decodierer **207** ausgegeben, wo die Erweiterung erfolgt. Bei der ersten Ausführungsform wird im Umkehrquantisierer **208** eine Umkehrquantisierung unter Verwendung der vorgenannten Quantisierungsskala durchgeführt, und das Ergebnis wird über eine Leitung **221** in die diskrete Umkehr-Kosinus-Transformationseinheit **209** eingegeben, wo räumliche Bereichsdaten durch diskrete Umkehr-Kosinus-Transformation abgeleitet werden. Der durch Erweiterung im Decodierer **207** erhaltene Differenzblock wird über die Leitung **221** an den zweiten Addierer **210** ausgegeben, wo ein über die Leitung **227** übertragener vorhergesagter Block addiert wird, um einen wiederhergestellten Block zu erzeugen. Dieser wiederhergestellte Block wird über eine Leitung **222** an den Bildspeicher **213** ausgegeben und dort gespeichert, um als Bezugsbild bei der nachfolgenden Codierung verwendet zu werden.

[0047] Bei der Bildcodierungsvorrichtung der ersten Ausführungsform empfängt der Codierer für Codes mit variabler Wortlänge **211** als Eingangsdaten quantisierte Transformationskoeffizienten und Nebeninformationen, die die Quantisierungsskala, den Bewegungsvektor usw. umfassen, und wandelt sie in Codes mit variabler Wortlänge oder Codes mit fester Wortlänge um. Nachstehend wird unter Bezugnahme

auf das Diagramm von [Fig. 1](#) beschrieben, wie quantisierte Transformationskoeffizienten vom Codierer für Codes mit variabler Wortlänge **211** verarbeitet werden.

[0048] Wie bei der Erläuterung des Standes der Technik beschrieben, werden Transformationskoeffizienten als zweidimensionales Array erhalten, das aus Null-Koeffizienten und Koeffizienten ungleich Null besteht. Bei diesem zweidimensionalen Array wird eine Umgruppierung in einer festgelegten Reihenfolge (im Zickzack) durchgeführt, um ein eindimensionales Array zu erhalten, und es werden Ereignisse erzeugt, die jeweils aus einer Anzahl von fortschreitenden Null-Koeffizienten (Run) und einem Wert der Koeffizienten ungleich Null (Level) bestehen. Dann wird Last addiert, der die Informationen umfasst, die angeben, ob ein durch Level dargestellter Transformationskoeffizient der letzte Koeffizient in einem Block ist oder nicht. Ein so erzeugtes Ereignis, das aus (Last, Run, Level) besteht, wird als Eingabe-Ereignis behandelt, das ein vom Codierer für Codes mit variabler Wortlänge **211** zu verarbeitendes Objekt ist.

[0049] Nach der Initiierung des Prozesses im Schritt **101** wird im Schritt **102** ein j-tes Ereignis erhalten. Im Schritt **103** wird ein Vergleich zwischen dem Eingabe-Ereignis und einem Bezugsereignis durchgeführt, das in einer Tabelle von Codes mit variabler Wortlänge im Codierer für Codes mit variabler Wortlänge **211** vorliegt. Für die Tabelle von Codes mit variabler Wortlänge werden die herkömmlich benutzten Tabellen der [Fig. 19](#) bis [Fig. 22](#) verwendet, und der Vergleich wird mit jedem Bezugsereignis, das in der Tabelle enthalten ist und aus (Last, Run, Level) besteht, durchgeführt.

[0050] Im Schritt **104** wird entschieden, ob ein Bezugsereignis mit einem passenden (Last, Run, Level) im vorstehenden Vergleichsprozess vorhanden ist oder nicht. Wenn im Schritt **104** entschieden wird, dass das Bezugsereignis vorhanden ist, wird Schritt **105** ausgeführt, und wenn es nicht vorhanden ist, wird Schritt **106** ausgeführt. Wenn jetzt Schritt **105** ausgeführt wird, wird ein Code mit variabler Wortlänge, der dem Bezugsereignis entspricht, ausgeführt, und danach wird Schritt **111** ausgeführt.

[0051] Wenn jedoch Schritt **106** ausgeführt wird, wird an dem im Eingabe-Ereignis enthaltenen Level eine Transformation unter Verwendung einer festgelegten Funktion ausgeführt, um ein transformiertes Eingabe-Ereignis zu erzeugen. Im anschließenden Schritt **107** wird wie bei Schritt **103** ein Vergleich zwischen dem transformierten Eingabe-Ereignis und einem Bezugsereignis durchgeführt. Im anschließenden Schritt **108** wird festgestellt, ob ein passendes Bezugsereignis vorhanden ist oder nicht, und wenn ein solches Bezugsereignis vorhanden ist, wird

Schritt **110** ausgeführt, und wenn keines vorhanden ist, wird Schritt **109** ausgeführt. Wenn jetzt Schritt **110** ausgeführt wird, wird das Ergebnis der Codierung ausgegeben, das der Code mit variabler Wortlänge ist, der dem Bezugsereignis entspricht, zu dem ein vorher eindeutig festgelegter zweiter Steuercode addiert wird. Daran schließt sich Schritt **111** an.

[0052] Wenn jedoch Schritt **109** ausgeführt wird, wird das Eingabe-Ereignis dem in den **Fig. 23** gezeigten Codierungsverfahren unterzogen, um einen Code mit fester Wortlänge zu erzeugen. Dann wird ein Codierungsergebnis ausgegeben, das der Code mit fester Wortlänge ist, zu dem ein erster Steuercode, der vom zweiten Steuercode unterscheidbar ist, addiert wird.

[0053] Wenn entweder Schritt **110** oder Schritt **109** ausgeführt wird, wird in einem anschließenden Schritt **111** festgestellt, ob ein verarbeitetes Eingabe-Ereignis ein letztes Eingabe-Ereignis ist oder nicht, und wenn das Ereignis das letzte ist, wird der Codierungsprozess im Schritt **112** beendet. Wenn das Ereignis jedoch nicht das letzte ist, wird im Schritt **113** um 1 erhöht, und es erfolgt ein Rücksprung zum Schritt **102**, wodurch ein nächstes Eingabe-Ereignis auf die gleiche Weise verarbeitet wird. So wird das Verfahren bis zum letzten Koeffizienten ungleich Null fortgeführt.

[0054] Bei der ersten Ausführungsform wird eine Deduktion eines Offset-Werts mit einem festgelegten Wert als festgelegte Funktion zum Erhalten eines transformierten Eingabe-Ereignisses aus dem Eingabe-Ereignis durchgeführt. Insbesondere wird ein transformiertes Eingabe-Ereignis durch Deduktion eines Offsets vom Wert des Levels des Eingabe-Ereignisses abgeleitet. Vorzugsweise wird ein Offset vom Absolutwert des Levels deduziert und dann ein Zeichencode des Levels (positiv oder negativ) angehängt. Der Offset kann ein Einzelwert sein oder einen variablen Wert haben, der Werten des Last und Run entspricht. Bei der ersten Ausführungsform kann der Offset aus der Tabelle in den **Fig. 19** bis **Fig. 22** dadurch erhalten werden, dass der maximale Level-Wert für jede Gruppe (Last, Run) als Offset-Wert verwendet wird: beispielsweise Offset = 12 für (Last, Run) = (0, 0), Offset = 6 für (Last, Run) = (0, 1) und Offset = 4 für (Last, Run) = (0, 2).

[0055] Die in den **Fig. 19** bis **Fig. 22** gezeigte Tabelle von Codes mit variabler Wortlänge ist für Transformationskoeffizienten mit einer relativ hohen Komprimierungsrate geeignet, und viele der in den Ereignissen in der Tabelle enthaltenen Levels haben kleine Werte. Daher wird bei einem Eingabe-Ereignis, dessen passendes Ereignis nicht in der Tabelle enthalten ist, die Deduktion zu seinem Level-Wert durchgeführt, um ein transformiertes Eingabe-Ereignis mit einem kleinen Level-Wert zu erhalten, wodurch der An-

teil der Durchführung der Codierung mit variabler Wortlänge sowie der Anteil der Verwendung kurzer Codes, die den Ereignissen mit kleinen Level-Werten in der Tabelle zugeordnet werden, erhöht werden.

[0056] Als erster Steuercode wird „0000011“ verwendet, was mit dem in **Fig. 23** gezeigten ESC-Code, der herkömmlich verwendet wird, identisch ist. Als zweiter Steuercode wird der zweimal wiederholte ESC-Code verwendet.

[0057] Die **Fig. 3** sind Diagramme, die Beispiele für das Codierungsergebnis zeigen. Hier wird das Codierungsverfahren nach der ersten Ausführungsform mit dem herkömmlichen Codierungsverfahren verglichen. Nehmen wir an, dass ein Ereignis mit (Last, Run, Level) = (0, 0, 13) ein zu codierendes Objekt ist. Die in den **Fig. 19** bis **Fig. 22** gezeigte Tabelle von Codes mit variabler Wortlänge hat diese Gruppe (0, 0, 13) nicht. Wenn daher das in den **Fig. 23** gezeigte herkömmliche Codierungsverfahren wie vorstehend beschrieben ausgeführt wird, wird die in den **Fig. 23** gezeigte Codierung mit fester Wortlänge durchgeführt und der ESC-Code wird hierzu addiert, um das Codierungsergebnis auszugeben. **Fig. 3(a)** zeigt das Ergebnis dieses Verfahrens, und es werden 6 Bit **303** und 8 Bit **304**, die jeweils Levels darstellen, nach 7 Bit **301** als ESC-Code und 1 Bit **302**, das Last darstellt, ausgegeben. Daher werden Levels durch **303**, die einen Null-Run zeigen, und den Code mit fester Wortlänge **304** dargestellt. Somit werden insgesamt 22 Bit benötigt.

[0058] Wenn jedoch im Ergebnis des Verfahrens gemäß der ersten Ausführungsform das betreffende Bezugsereignis in der Tabelle von Codes mit variabler Wortlänge in den Schritten **103** bis **104** des Ablaufdiagramms von **Fig. 1** nicht vorhanden ist, wird im Gegensatz zum Stand der Technik die Codierung mit fester Wortlänge nicht sofort durchgeführt, und stattdessen werden die Transformation und der Vergleich nochmals durchgeführt, wie in den Schritten **106** bis **108** gezeigt. Bei der Transformation erfolgt eine Deduktion des Offsets (= 12) vom Level-Wert 13, um einen Level-Wert 1 zu erhalten, sodass ein transformiertes Eingabe-Ereignis (0, 0, 1) erhalten wird. Da, wie in **Fig. 18** gezeigt, das Ereignis (0, 0, 1) in der Tabelle von Codes mit variabler Wortlänge vorhanden ist, wird nach Schritt **108** Schritt **110** ausgeführt. Somit wird, wie in **Fig. 3(b)** gezeigt, das Ergebnis der Codierung mit einem zweiten Steuercode **806** und einem Code mit variabler Wortlänge **807**, der (0, 0, 1) entspricht, abgeleitet. In diesem Fall sind also insgesamt 17 Bit erforderlich, was bedeutet, dass der Umfang des Codes gegenüber dem herkömmlichen Code mit 22 Bit geringer ist.

[0059] Wenn in dem Ablaufdiagramm von **Fig. 1** beim Vergleichen mit dem transformierten Eingabe-Ereignis ein betreffendes Bezugsereignis der Ta-

belle von Codes mit variabler Wortlänge nicht vorhanden ist, ist das Verfahren im Schritt **109** identisch mit dem herkömmlichen. Daher kann bei der ersten Ausführungsform im ungünstigsten Fall die mit dem Stand der Technik identische Codierung durchgeführt werden. Wenn die Länge des Codes, zu dem der zweite Steuercode addiert wird, größer als die des Codes ist, zu dem der erste Steuercode addiert wird, kann eine niedrigere Anzahl von Bits durch Codierung mit einem ersten Steuercode und einem Code mit fester Wortlänge erhalten werden.

[0060] Nachstehend wird ein Decodierungsverfahren für codierte Daten beschrieben, die durch das Bildcodierungsverfahren nach der ersten Ausführungsform erhalten werden.

[0061] [Fig. 4](#) ist ein Blockdiagramm, das die Konfiguration einer Bilddecodierungsvorrichtung gemäß der ersten Ausführungsform zeigt. Wie in [Fig. 4](#) gezeigt, weist die Bilddecodierungsvorrichtung gemäß der ersten Ausführungsform einen Datenanalysator (DA) **402**, einen Addierer **406**, einen Decodierer für Codes mit variabler Wortlänge (VLD) **408**, einen Decodierer **403**, einen Bildspeicher (FM) **409** und einen Bewegungskompensator (MC) **410** auf, und sie empfängt codierte Daten, die Gegenstand des Decodierungsverfahrens sind, von einem Eingabeterminal **401** und gibt die decodierten Daten, die die Ausgangsdaten der Bilddecodierungsvorrichtung sind, von einem Ausgabeterminal **407** aus. Des Weiteren weist der Decodierer **403** einen Umkehrquantisierer (IQ) **404** und eine diskrete Umkehr-Kosinus-Transformationseinheit (IDCT) **405** auf.

[0062] Nachstehend wird die Funktionsweise der wie vorstehend aufgebauten Bilddecodierungsvorrichtung der ersten Ausführungsform zur Durchführung der Decodierung der codierten Daten, die von der in [Fig. 2](#) gezeigten Bildcodierungsvorrichtung kompressionscodiert worden sind, beschrieben.

[0063] Codierte Daten werden als zu verarbeitendes Objekt in das Eingabeterminal **401** eingegeben und dann von Datenanalysator **402** analysiert, um einen Code mit variabler Wortlänge entsprechend den quantisierten Transformationskoeffizienten und die Codes der Nebeninformationen, die die Quantisierungsskala und den Bewegungsvektor umfassen, abzuleiten. Aus diesen werden die Quantisierungsskala und der Code mit variabler Wortlänge der quantisierten Transformationskoeffizienten über eine Leitung **412** zum Decodierer für Codes mit variabler Wortlänge **408** gesendet und der Bewegungsvektor wird über eine Leitung **418** an den Bewegungskompensator **410** gesendet. Im Decodierer für Codes mit variabler Wortlänge **408** wird der Code mit variabler Wortlänge der quantisierten Transformationskoeffizienten in die Transformationskoeffizienten durch Verweis auf die Tabelle von Codes mit variabler Wortlänge decodiert

und dann werden die Koeffizienten umgruppiert, um komprimierte Differenzblöcke zu erhalten, und die Blöcke werden an den Decodierer **403** ausgegeben.

[0064] Im Decodierer **403** wird der komprimierte Differenzblock erweitert, um einen Differenzblock zu erzeugen. Im Decodierer **403** der ersten Ausführungsform wird im Umkehrquantisierer **404** unter Verwendung einer Quantisierungsskala eine Umkehrquantisierung durchgeführt und die diskrete Umkehr-Kosinus-Transformationseinheit **405** transformiert das Ergebnis, sodass ein Frequenzbereichssignal in ein Raumbereichssignal umgewandelt wird. Dieses Signal wird als Differenzblock an einen Addierer **406** ausgegeben.

[0065] Im Bewegungskompensator **410**, der den Bewegungsvektor als Eingangssignal empfängt, wird anhand des Bewegungsvektors eine Adresse erzeugt, um auf einen Bildspeicher **409** zuzugreifen, und für das im Bildspeicher **409** gespeicherte Bild wird ein vorhergesagter Block erzeugt. Der vorhergesagte Block wird als Ausgangssignal für den Addierer **406** erzeugt, wo das Ausgangssignal und der durch Erweiterung erhaltene Differenzblock addiert werden, um einen wiederhergestellten Block zu erzeugen. Das so wiederhergestellte Bild wird vom Ausgabeterminal **407** als Ausgangssignal der Bilddecodierungsvorrichtung an den Bildspeicher **409** ausgegeben, um dort gespeichert zu werden.

[0066] [Fig. 5](#) ist ein Ablaufdiagramm, das den Ablauf des Verfahrens der Decodierung mit variabler Wortlänge mit dem Decodierer für Codes mit variabler Wortlänge **408** zeigt. Nachstehend wird die Funktionsweise des Decodierers für Koeffizienten mit variabler Wortlänge **408** bei der Decodierung unter Bezugnahme auf das Ablaufdiagramm von [Fig. 5](#) beschrieben.

[0067] Nach der Initiierung des Prozesses im Schritt **501** wird zunächst ein aufgrund von quantisierten Transformationskoeffizienten erhaltener Code mit variabler Wortlänge, der das j-te zu verarbeitende Objekt ist, als Eingabe-Code mit variabler Wortlänge eingegeben. Im Schritt **503** wird festgestellt, ob der Eingabe-Code mit variabler Wortlänge einen Steuercode enthält oder nicht, und wenn er keinen Steuercode enthält, wird Schritt **504** ausgeführt; wenn er den Code enthält, wird Schritt **505** ausgeführt. In dem Fall, dass Schritt **504** ausgeführt wird, das heißt, wenn kein Steuercode enthalten ist, wird durch Verweis auf die Tabelle von Codes mit variabler Wortlänge eine Gruppe von (Last, Run, Level), die dem eingegebenen Code mit variabler Wortlänge entspricht, ausgegeben. Daran schließt sich Schritt **510** an, der später beschrieben wird.

[0068] Wenn jedoch Schritt **505** ausgeführt wird, das heißt, wenn ein Steuercode enthalten ist, wird

festgestellt, ob der Steuercode der zweite Steuercode ist oder nicht. Wenn er der zweite Steuercode ist, wird Schritt **506** ausgeführt, und wenn er nicht der zweite Steuercode ist, wird Schritt **507** ausgeführt. In dem Fall, dass Schritt **506** ausgeführt wird, das heißt, wenn der Steuercode der zweite Steuercode ist, wird eine Gruppe (Last, Run, Level), die dem eingegebenen Code mit variabler Wortlänge entspricht, unter Verwendung der Tabelle von Codes mit variabler Wortlänge erhalten, und sein Level-Wert wird mit einer vorgegebenen Funktion transformiert und das Ergebnis wird ausgegeben. Daran schließt sich Schritt **510** an, der später beschrieben wird. Bei der ersten Ausführungsform wird ein vorgegebener Offset addiert, dessen Wert sich entsprechend einer Gruppe (Last, Run) wie vorstehend beschrieben ändert.

[0069] Nachstehend wird das in [Fig. 3\(b\)](#) gezeigte Decodierungsverfahren für den Code betrachtet. Wenn im Schritt **505** festgestellt wird, dass es sich um den zweiten Steuercode handelt, wird Schritt **506** ausgeführt. Dadurch kann eine Gruppe (Last, Run, Level), die einem Teil des Codes „100“ entspricht, der als **306** in [Fig. 3\(b\)](#) dargestellt ist, durch Verweis auf die Tabelle von Codes mit variabler Wortlänge erhalten werden, und ein Offset = 12 wird zu diesem Level addiert, um die Originalgruppe (Last, Run, Level) = (0, 0, 13) zu regenerieren.

[0070] Wenn Schritt **507** ausgeführt wird, wird auch festgestellt, ob der Steuercode der erste Steuercode ist oder nicht. Wenn er der erste Steuercode ist, wird Schritt **508** ausgeführt, aber wenn er nicht der erste Steuercode ist, ist klar, dass ein anderer Steuercode als der erste und zweite Steuercode addiert worden ist. Daher wird im Schritt **509** das Decodierungsverfahren beendet oder es wird eine Fehlermeldung angezeigt, die den Nutzer davon in Kenntnis setzt, dass das Decodierungsverfahren abgebrochen wurde. Wenn jedoch Schritt **508** ausgeführt wird, das heißt, wenn es sich um den ersten Steuercode handelt, wird er als Code mit fester Wortlänge decodiert, um Koeffizienten zu regenerieren.

[0071] Wenn einer der Schritte **504**, **506** und **508** ausgeführt wird, wird im nachfolgenden Schritt **510** festgestellt, ob der verarbeitete Eingabe-Code mit variabler Wortlänge der letzte ist oder nicht, und wenn er der letzte ist, wird im Schritt **511** die Codierung beendet. Wenn er jedoch nicht der letzte ist, erfolgt der Rücksprung zum Schritt **502**, nachdem j um 1 erhöht worden ist, wodurch ein nächster eingegebener Code mit variabler Wortlänge in der gleichen Weise verarbeitet wird. So wird das Verfahren bis zum letzten Code mit variabler Wortlänge fortgesetzt.

[0072] Wenn, wie vorstehend beschrieben, gemäß dem in [Fig. 1](#) dargestellten ersten Codierungsverfahren der ersten Ausführungsform beim Vergleichen in den Schritten **103** bis **104** das betreffende Bezugs-

ereignis nicht in der Tabelle von Codes mit variabler Wortlänge enthalten ist, wird im Schritt **106** das Eingabe-Ereignis transformiert und in den Schritten **107** bis **108** wird der Vergleich mit dem resultierenden transformierten Eingabe-Ereignis durchgeführt. Somit kann der Anteil der Durchführung der Codierung mit fester Wortlänge im Schritt **109** verringert werden und die Möglichkeit, dass kurze Codes bei der Codierung mit variabler Wortlänge zugewiesen werden, wird durch Verwendung des transformierten Eingabe-Ereignisses erhöht, was zu einer besseren Komprimierungsrate des Ergebnisses der Codierung insgesamt führt.

[0073] Da die Bildcodierungsvorrichtung der ersten Ausführungsform den Codierer für Codes mit variabler Wortlänge **211** aufweist, der die vorstehend beschriebene Codierung durchführt, kann mit einer einzigen Tabelle von Codes mit variabler Wortlänge ein Codierungsergebnis mit einer hohen Komprimierungsrate erhalten werden, gleichgültig, ob die Komprimierungsrate der Kompressionscodierung im Codierer **203** hoch oder niedrig ist.

[0074] Das Bilddecodierungsverfahren und die Bilddecodierungsvorrichtung gemäß der ersten Ausführungsform können eine zweckmäßige Decodierung entsprechend dem in der vorstehend beschriebenen Weise erhaltenen Codierungsergebnis durchführen, um ein wiederhergestelltes Bild zu erhalten.

[0075] Obwohl beschrieben wird, dass bei der in der ersten Ausführungsform dargestellten Codierung die diskrete Kosinus-Transformation für die Kompressionscodierung durchgeführt wird, kann stattdessen auch die Wellenformcodierung, wie etwa Wavelet-Codierung, angewendet werden, mit der die gleiche Codierung durchgeführt werden kann.

[0076] Obwohl bei der in der ersten Ausführungsform beschriebenen Codierung ein Level-Wert eines Eingabe-Ereignisses bei der Transformation für das Eingabe-Ereignis transformiert wird, kann stattdessen auch ein Run-Wert des Eingabe-Ereignisses transformiert werden. In diesem Fall wird der maximale Run-Wert, der der Gruppe (Last, Level) jedes Bezugsereignisses in der Tabelle von Codes mit variabler Wortlänge der [Fig. 19](#) bis [Fig. 22](#) entspricht, als Offset-Wert verwendet, und (Offset + 1) wird vom Run-Wert eines Eingabe-Ereignisses deduziert, um ein transformiertes Eingabe-Ereignis zu erhalten. Die Transformation kann auch nach einem anderen Verfahren wie Multiplikation eines Level-Werts oder eines Run-Werts, einem Verfahren unter Verwendung einer quadratischen Funktion oder durch Ändern des dynamischen Bereiches durchgeführt werden. Es ist auch möglich, einen Level oder Run in einer Tabelle von Codes mit variabler Wortlänge mit einer vorgegebenen Funktion zu transformieren und dann die Komprimierung durchzuführen, statt eine vorgegebene

Funktion an einem Level oder Run eines Eingabe-Ereignisses auszuführen.

[0077] Wenn, wie vorstehend beschrieben, bei der Codierung ein Run-Wert statt eines Level-Werts transformiert wird, wird (Offset + 1) zu dem der Decodierung mit variabler Wortlänge unterzogenen Run-Wert addiert, um einen Run-Ausgangswert zu erzeugen. In diesem Fall ist der Offset-Wert der maximale Run-Wert, der einer Gruppe (Last, Level) entspricht, wie vorstehend dargelegt.

[0078] Obwohl bei der ersten Ausführungsform die Abläufe der Codierungs- und Decodierungsverfahren anhand der Ablaufdiagramme der [Fig. 1](#) und [Fig. 5](#) beschrieben werden, sind die Codierungs- und Decodierungsverfahren nicht auf diese Abläufe beschränkt, und Abläufe, die die gleichen Verfahren ermöglichen, können ebenfalls die Codierungsleistung verbessern und ein geeignetes Decodierungsverfahren für das Codierungsergebnis implementieren.

Ausführungsform 2

[0079] Bei einem Bildcodierungsverfahren gemäß einer zweiten Ausführungsform wird wie bei der ersten Ausführungsform ein Ereignis als zu codierendes Objekt transformiert, und die Anzahl der Bits kann durch Verwendung eines Moduscodes verringert werden.

[0080] [Fig. 6](#) ist ein Ablaufdiagramm, das den Ablauf eines Verfahrens zur Codierung mit variabler Wortlänge gemäß der zweiten Ausführungsform zeigt. Die Bildcodierungsvorrichtung der zweiten Ausführungsform hat die gleiche Konfiguration wie die der ersten Ausführungsform, und daher wird [Fig. 2](#) für die Beschreibung verwendet.

[0081] Das Bildcodierungsverfahren gemäß der zweiten Ausführungsform unterscheidet sich von dem der ersten Ausführungsform nur hinsichtlich der Funktionsweise des in [Fig. 2](#) gezeigten Codierers für Codes mit variabler Wortlänge **211**, und daher wird die Funktionsweise des Codierers für Codes mit variabler Wortlänge **211** bei der Codierung nachstehend unter Bezugnahme auf [Fig. 6](#) beschrieben.

[0082] Die Schritte **601** bis **608** werden in der gleichen Weise wie die Schritte **101** bis **108** bei der ersten Ausführungsform ausgeführt, und im Schritt **108** wird eine Entscheidung aufgrund eines Vergleichs zwischen einem transformierten Eingabe-Ereignis und einem Bezugsereignis einer Tabelle von Codes mit variabler Wortlänge getroffen. Wenn entschieden wird, dass es ein passendes Bezugsereignis gibt, wird Schritt **601** ausgeführt, und wenn kein solches Ereignis gefunden wird, wird Schritt **609** ausgeführt.

[0083] Wenn Schritt **610** ausgeführt wird, wird ein

Codierungsergebnis ausgegeben, das ein Code mit variabler Wortlänge ist, der dem Bezugsereignis entspricht, zu dem ein vorher eindeutig festgelegter Steuercode und ein zweiter Moduscode addiert werden. Dann schließt sich Schritt **611** an. Hier wird „0000011“, der mit dem in [Fig. 22](#) gezeigten ESC-Code identisch ist, als Steuercode verwendet und „0“ wird als zweiter Moduscode verwendet.

[0084] Wenn jedoch Schritt **609** ausgeführt wird, wird das Eingabe-Ereignis dem in [Fig. 23](#) dargestellten Codierungsverfahren unterzogen, um einen Code mit fester Wortlänge zu erzeugen. Dann wird ein Codierungsergebnis ausgegeben, das der Code mit fester Wortlänge ist, zu dem ein Steuercode und ein erster Moduscode addiert werden. „1“ wird als erster Moduscode verwendet.

[0085] Wenn Schritt **610** oder Schritt **609** ausgeführt wird, wird in einem nachfolgenden Schritt **611** entschieden, ob ein verarbeitetes Eingabe-Ereignis ein letztes Eingabe-Ereignis ist oder nicht, und wenn das Ereignis das letzte ist, wird der Codierungsprozess im Schritt **612** beendet. Wenn das Ereignis jedoch nicht das letzte ist, erfolgt der Rücksprung zum Schritt **602**, nachdem im Schritt **613** j um 1 erhöht worden ist, wodurch ein nächstes Eingabe-Ereignis auf die gleiche Weise verarbeitet wird. So wird das Verfahren bis zum letzten Koeffizienten ungleich Null fortgeführt.

[0086] Bei der zweiten Ausführungsform wird mit einem Offset, der einem Level eines Eingabe-Ereignisses entspricht, die Transformation durchgeführt, um ein transformiertes Eingabe-Ereignis zu erzeugen. Wie bei der ersten Ausführungsform kann der Offset einen festen oder einen variablen Wert annehmen, der einer Gruppe aus einem Last und einem Run entspricht. Bei dieser Ausführungsform wird als Offset der maximale Level-Wert, der jeweils einer Gruppe (Last, Run) entspricht, die aus der in den [Fig. 19](#) bis [Fig. 22](#) dargestellten Tabelle von Codes mit variabler Wortlänge erhalten werden kann, so verwendet, dass Offset = 12 für (Last, Run) = (0, 0), Offset = 6 für (Last, Run) = (0, 1) und Offset = 4 für (Last, Run) = (0, 2) ist.

[0087] Nachstehend wird die Decodierung der codierten Daten beschrieben, die bei dem in der zweiten Ausführungsform erläuterten Bildcodierungsverfahren erhalten werden. Da die Konfiguration der Bilddecodierungsvorrichtung gemäß der zweiten Ausführungsform mit der der ersten Ausführungsform identisch ist, wird [Fig. 4](#) für die Beschreibung verwendet.

[0088] [Fig. 7](#) ist ein Ablaufdiagramm, das den Ablauf eines Verfahrens zur Decodierung mit variabler Wortlänge gemäß der zweiten Ausführungsform zeigt. Das Bilddecodierungsverfahren gemäß der

zweiten Ausführungsform unterscheidet sich von dem der ersten Ausführungsform nur hinsichtlich der Funktionsweise des in [Fig. 4](#) gezeigten Decodierers für Codes mit variabler Wortlänge **408**, und daher wird die Funktionsweise des Decodierers für Codes mit variabler Wortlänge **408** bei der Decodierung nachstehend unter Bezugnahme auf [Fig. 7](#) beschrieben.

[0089] Die Schritte **701** bis **704** werden in der gleichen Weise wie die Schritte **501** bis **504** bei der ersten Ausführungsform (siehe [Fig. 5](#)) ausgeführt, und dann wird im Schritt **705** eine Entscheidung getroffen. Wenn im Schritt **703** entschieden wird, dass ein Steuercode vorhanden ist, wird festgestellt, ob ein nächstes Bit nach dem Steuercode „0“ oder „1“ ist. Wenn das Bit „0“ ist, wird Schritt **706** ausgeführt, und wenn es „1“ ist, wird Schritt **708** ausgeführt.

[0090] Wenn Schritt **706** ausgeführt wird, wird unter Verwendung einer Tabelle von Codes mit variabler Wortlänge eine Gruppe (Last, Run, Level), die einem eingegebenen Code mit variabler Wortlänge entspricht, erhalten, und mit einer vorgegebenen Funktion, die dem Level-Wert entspricht, wird eine Transformation durchgeführt, und dann wird das Decodierungsergebnis ausgegeben. Bei der Transformation der zweiten Ausführungsform wird ein vorgegebener Offset zu einem erhaltenen Level-Wert addiert. Der Offset-Wert nimmt einen variablen Wert an, der einer Gruppe (Last, Run) entspricht, wie vorstehend beschrieben.

[0091] Wenn jedoch Schritt **708** ausgeführt wird, wird ein eingegebener Code mit variabler Wortlänge als Code mit fester Wortlänge decodiert, um Koeffizienten zu regenerieren. Wenn einer der Schritte **704**, **706** und **708** ausgeführt wird, wird in einem nachfolgenden Schritt **710** festgestellt, ob ein verarbeiteter eingegebener Code mit variabler Wortlänge der letzte Eingabe-Code mit variabler Wortlänge ist oder nicht, und wenn der Code der letzte ist, wird die Codierung im Schritt **711** beendet. Wenn er jedoch nicht der letzte Code ist, erfolgt der Rücksprung zum Schritt **702**, nachdem im Schritt **712** j um 1 erhöht worden ist, wodurch ein nächster eingegebener Code mit variabler Wortlänge auf die gleiche Weise verarbeitet wird. So wird das Verfahren bis zum letzten Code mit variabler Wortlänge fortgeführt.

[0092] Wenn, wie vorstehend beschrieben, bei dem Bildcodierungsverfahren der zweiten Ausführungsform beim Vergleichen in den Schritten **603** bis **604** ein passendes Bezugsereignis in der Tabelle von Codes mit variabler Wortlänge nicht vorhanden ist, wird im Schritt **606** das Eingabe-Ereignis transformiert und in den Schritten **607** bis **608** wird unter Verwendung des erhaltenen transformierten Eingabe-Ereignisses ein Vergleich durchgeführt, wie in [Fig. 6](#) gezeigt. Dadurch wird der Anteil der Durchführung der

Codierung mit fester Wortlänge im Schritt **609** verringert und die Möglichkeit, dass kurze Codes bei der Codierung mit variabler Wortlänge zugewiesen werden, wird durch Verwendung des transformierten Eingabe-Ereignisses erhöht, was zu einer besseren Gesamtkomprimierungsrate des Codierungsergebnisses führt. Da im Schritt **609** oder **610** ein Ein-Bit-Moduscode verwendet wird, um zwischen der Codierung mit fester Wortlänge und einem dem Offset-Verfahren unterzogenen Level zu unterscheiden, kann vorteilhafterweise die Gesamtanzahl der Bits des Codierungsergebnisses gegenüber der ersten Ausführungsform, wo erste und zweite Steuercodes verwendet werden, verringert werden.

[0093] Da die Bildcodierungsvorrichtung gemäß der zweiten Ausführungsform einen Codierer für Codes mit variabler Wortlänge **211** aufweist, der die vorstehend beschriebene Codierung durchführt, kann unter Verwendung einer einzigen Tabelle von Codes mit variabler Wortlänge ein Codierungsergebnis mit einer besseren Komprimierungsrate erhalten werden, gleichgültig, ob die Komprimierungsrate der Kompressionscodierung im Codierer **203** hoch oder niedrig ist.

[0094] Das Bilddecodierungsverfahren und die Bilddecodierungsvorrichtung gemäß der zweiten Ausführungsform ermöglichen die Durchführung einer zweckmäßigen Decodierung entsprechend dem in der vorstehend beschriebenen Weise gewonnenen Codierungsergebnis, um ein wiederhergestelltes Bild zu erhalten.

[0095] Obwohl bei der zweiten Ausführungsform nur ein Moduscode verwendet wird, können auch zwei oder mehr Moduscodes zur Unterscheidung mehrerer Modi verwendet werden. So kann beispielsweise ein Modus, bei dem der Run-Wert Null geändert wird, als dritter Modus hinzugefügt werden.

[0096] Obwohl bei der zweiten Ausführungsform ein Level eines Eingabe-Ereignisses transformiert wird, um wie bei der ersten Ausführungsform ein transformiertes Eingabe-Ereignis zu erhalten, kann auch ein Run des Eingabe-Ereignisses transformiert werden. Obwohl hier die Transformation unter Verwendung eines Offsets durchgeführt wird, können auch andere Verfahren wie Multiplikation eines Levels, Verwendung einer quadratischen Funktion und Ändern des dynamischen Bereiches verwendet werden.

Ausführungsform 3

[0097] Bei einem Bildcodierungsverfahren gemäß einer dritten Ausführungsform wird, wie bei der ersten und zweiten Ausführungsform, ein Ereignis transformiert, das ein zu codierendes Objekt ist, und die Codierungsleistung kann durch wiederholte Durchführung der Transformation verbessert werden.

[0098] Die **Fig. 8** sind Ablaufdiagramme, die den Ablauf eines Verfahrens zur Codierung mit variabler Wortlänge gemäß der dritten Ausführungsform zeigen. Die Bildcodierungsvorrichtung der dritten Ausführungsform hat die gleiche Konfiguration wie die der ersten Ausführungsform, und daher wird **Fig. 2** für die Beschreibung verwendet.

[0099] Das Bildcodierungsverfahren gemäß der dritten Ausführungsform unterscheidet sich von dem der ersten Ausführungsform nur hinsichtlich der Funktionsweise des in **Fig. 2** gezeigten Codierers für Codes mit variabler Wortlänge **211**, und daher wird die Funktionsweise des Codierers für Codes mit variabler Wortlänge **211** bei der Codierung nachstehend unter Bezugnahme auf die **Fig. 8** beschrieben.

[0100] Im Ablaufdiagramm von **Fig. 8(a)** werden die Schritte **801** bis **805** in der gleichen Weise wie die Schritte **101** bis **105** bei der ersten Ausführungsform ausgeführt. Bei der dritten Ausführungsform wird bei der Transformation im Schritt **806** ein erstes transformiertes Eingabe-Ereignis durch Transformation unter Verwendung einer vorgegebenen Funktion erhalten. Hier wird die Transformation in der gleichen Weise wie bei der ersten Ausführungsform durchgeführt und ein erstes transformiertes Eingabe-Ereignis wird durch Deduktion eines vorgegebenen Offsets von einem im Eingabe-Ereignis enthaltenen Level-Wert erhalten. In diesem Fall wird der maximale Level-Wert, der jeweils einer in der in den **Fig. 19** bis **Fig. 22** gezeigten Tabelle von Codes mit variabler Wortlänge vorhandenen Gruppe (Last, Run) entspricht, als Offset verwendet: beispielsweise Offset = 12 für (Last, Run) = (0, 0), Offset = 6 für (Last, Run) = (0, 1) und Offset = 4 für (Last, Run) = (0, 2).

[0101] Im Schritt **807** wird ein Vergleich zwischen dem ersten transformierten Eingabe-Ereignis und einem in einer Tabelle von Codes mit variabler Wortlänge enthaltenen Bezugseignis durchgeführt, und im Schritt **808** wird aufgrund dieses Vergleichs eine Entscheidung getroffen. Wenn im Schritt **810** entschieden wird, dass es ein passendes Bezugseignis gibt, wird Schritt **810** ausgeführt, und wenn kein solches Ereignis gefunden wird, läuft das in **Fig. 8(b)** gezeigte Verfahren ab. Wenn Schritt **810** ausgeführt wird, wird das Codierungsergebnis ausgegeben, das ein Code mit variabler Wortlänge ist, zu dem ein vorher eindeutig festgelegter Steuercode und ein zweiter Moduscode addiert werden. Dann folgt Schritt **811**. Hier wird „0000011“, der mit dem in **Fig. 22** gezeigten ESC-Code identisch ist, als Steuercode verwendet und „0“ wird als zweiter Moduscode verwendet.

[0102] Wenn jedoch das in **Fig. 8(b)** gezeigte Verfahren durchgeführt wird, wird der Run des Eingabe-Ereignisses unter Verwendung einer vorgegebenen Funktion transformiert, um ein zweites transfor-

miertes Eingabe-Ereignis zu erhalten. Bei der dritten Ausführungsform wird ein vorgegebener Offset-Wert von einem Run-Wert deduziert, um ein zweites transformiertes Eingabe-Ereignis mit einem neuen Run-Wert zu erzeugen. In diesem Fall wird der maximale Run-Wert, der jeweils einer Gruppe (Last, Level) in der Tabelle von Codes mit variabler Wortlänge entspricht, als Offset verwendet.

[0103] Im Schritt **822** wird ein Vergleich zwischen dem zweiten transformierten Eingabe-Ereignis und einem in der Tabelle von Codes mit variabler Wortlänge enthaltenen Bezugseignis durchgeführt, und im Schritt **823** wird aufgrund des Vergleichs eine Entscheidung getroffen. Wenn entschieden wird, dass ein passendes Bezugseignis vorhanden ist, wird Schritt **824** ausgeführt, und wenn kein solches Ereignis gefunden wird, wird Schritt **825** ausgeführt. Wenn jetzt Schritt **824** ausgeführt wird, wird das Codierungsergebnis ausgegeben, das ein Code mit variabler Wortlänge ist, der dem Bezugseignis entspricht, zu dem ein Steuercode und ein dritter Moduscode addiert werden. Dann folgt der in **Fig. 8(a)** gezeigte Schritt **811**. Es wird festgelegt, dass der dritte Moduscode „10“ ist.

[0104] Wenn jedoch Schritt **825** ausgeführt wird, wird das Eingabe-Ereignis dem in den **Fig. 23** dargestellten Codierungsverfahren unterzogen, um einen Code mit fester Wortlänge zu erzeugen. Dann wird das Codierungsergebnis ausgegeben, das ein Code mit fester Wortlänge ist, zu dem ein Steuercode und ein erster Moduscode addiert werden. Es wird festgelegt, dass der erste Moduscode „11“ ist.

[0105] Wenn einer der Schritte **810** und der in **Fig. 8(b)** gezeigten Schritte **824** und **825** ausgeführt wird, wird in einem nachfolgenden Schritt **811** festgestellt, ob das verarbeitete Eingabe-Ereignis das letzte Eingabe-Ereignis ist oder nicht, und wenn das Ereignis das letzte ist, wird im Schritt **812** die Codierung beendet. Wenn es jedoch nicht das letzte ist, erfolgt der Rücksprung zum Schritt **802**, nachdem im Schritt **813** j um 1 erhöht worden ist, wodurch das nächste Eingabe-Ereignis in gleicher Weise verarbeitet wird. So wird das Verfahren bis zum letzten Koeffizienten ungleich Null fortgesetzt.

[0106] Nachstehend wird ein Decodierungsverfahren für die codierten Daten beschrieben, die bei dem bei der dritten Ausführungsform beschriebenen Bildcodierungsverfahren erhalten werden. Da die Konfiguration der Bilddecodierungsvorrichtung gemäß der dritten Ausführungsform mit der der Bilddecodierungsvorrichtung der ersten Ausführungsform identisch ist, wird **Fig. 4** für die Beschreibung verwendet.

[0107] **Fig. 9** ist ein Ablaufdiagramm, das den Ablauf eines Verfahrens zur Decodierung mit variabler Wortlänge gemäß der dritten Ausführungsform zeigt.

Das Bildcodierungsverfahren gemäß der dritten Ausführungsform unterscheidet sich von dem der ersten Ausführungsform nur hinsichtlich der Funktionsweise des in [Fig. 4](#) gezeigten Decodierers für Codes mit variabler Wortlänge **408**, und daher wird die Funktionsweise des Codierers für Codes mit variabler Wortlänge **408** bei der Decodierung nachstehend unter Bezugnahme auf [Fig. 9](#) beschrieben.

[0108] Die Schritte **901** bis **906** werden in der gleichen Weise wie die Schritte **701** bis **706** der in [Fig. 7](#) gezeigten zweiten Ausführungsform ausgeführt. Wenn im Schritt **905** festgestellt wird, dass ein Zielbit „1“ ist, wird Schritt **907** ausgeführt, in dem entschieden wird, ob das nachfolgende Bit „0“ oder „1“ ist. Wenn es „0“ ist, wird Schritt **908** aus, und wenn es „1“ ist, wird Schritt **909** ausgeführt.

[0109] Wenn Schritt **908** ausgeführt wird, wird unter Verwendung einer Tabelle von Codes mit variabler Wortlänge eine Gruppe (Last, Run, Level), die einem eingegebenen Code mit variabler Wortlänge entspricht, erhalten, und mit einer vorgegebenen Funktion wird der Run-Wert transformiert, und dann wird das Decodierungsergebnis ausgegeben. Bei der Transformation gemäß der dritten Ausführungsform wird ein vorgegebener (Offset + 1) zu einem resultierenden Run-Wert addiert.

[0110] Wenn jedoch Schritt **909** ausgeführt wird, wird ein eingegebener Code mit variabler Wortlänge als Code mit fester Wortlänge decodiert, um Koeffizienten zu regenerieren. Wenn einer der Schritte **904**, **906**, **908** und **900** ausgeführt wird, wird in einem nachfolgenden Schritt **910** festgestellt, ob ein eingegebener Code mit variabler Wortlänge der letzte Eingabe-Code mit variabler Wortlänge ist oder nicht, und wenn der Code der letzte ist, wird die Codierung im Schritt **911** beendet. Wenn er jedoch nicht der letzte Code ist, erfolgt der Rücksprung zum Schritt **902**, nachdem im Schritt **912** j um 1 erhöht worden ist, wodurch ein nächster eingegebener Code mit variabler Wortlänge auf die gleiche Weise verarbeitet wird. So wird das Verfahren bis zum letzten Code mit variabler Wortlänge fortgeführt.

[0111] Wenn, wie vorstehend beschrieben, bei dem Bildcodierungsverfahren gemäß der dritten Ausführungsform beim Vergleichen in den in [Fig. 8\(a\)](#) gezeigten Schritten **803** bis **804** ein betreffendes Bezugsereignis in der Tabelle von Codes mit variabler Wortlänge nicht gefunden wird, wird im Schritt **806** das Eingabe-Ereignis transformiert, und in den Schritten **807** bis **808** wird ein Vergleich unter Verwendung des resultierenden ersten transformierten Eingabe-Ereignisses durchgeführt. Wenn zu diesem Zeitpunkt kein passendes Bezugsereignis in der Tabelle von Codes mit variabler Wortlänge gefunden wird, wird das in [Fig. 8\(b\)](#) gezeigte Verfahren durchgeführt, bei dem im Schritt **821** das Eingabe-Ereignis

transformiert wird und in den Schritten **822** bis **823** ein Vergleich unter Verwendung des resultierenden zweiten transformierten Eingabe-Ereignisses durchgeführt wird. So wird durch Ausführung von Schritt **825** der Anteil der Durchführung der Codierung mit fester Wortlänge gegenüber dem jeweiligen Anteil bei der ersten und zweiten Ausführungsform weiter verringert und die Möglichkeit, dass bei der Codierung mit variabler Wortlänge kurze Codes zugewiesen werden, wird weiter erhöht, was zu einer besseren Gesamtkomprimierungsrate des Codierungsergebnisses führt.

[0112] Da die Bildcodierungsvorrichtung gemäß der dritten Ausführungsform einen Codierer für Codes mit variabler Wortlänge **211** aufweist, der die vorstehend beschriebene Codierung durchführt, kann unter Verwendung einer einzigen Tabelle von Codes mit variabler Wortlänge ein Codierungsergebnis mit einer besseren Komprimierungsrate erhalten werden, gleichgültig, ob die Komprimierungsrate der Kompressionscodierung im Codierer **203** hoch oder niedrig ist.

[0113] Das Bilddecodierungsverfahren und die Bilddecodierungsvorrichtung gemäß der dritten Ausführungsform ermöglichen die Durchführung einer zweckmäßigen Decodierung entsprechend dem in der vorstehend beschriebenen Weise erhaltenen Codierungsergebnis, um ein wiederhergestelltes Bild zu erhalten.

Ausführungsform 4

[0114] Ein Bildcodierungsverfahren gemäß einer vierten Ausführungsform soll die Möglichkeit der Verwendung einer Tabelle von Codes mit variabler Wortlänge durch Vergleichen unter Verwendung mehrerer Tabellen von Codes mit variabler Wortlänge erhöhen und dadurch die Codierungsleistung verbessern.

[0115] [Fig. 10](#) ist ein Ablaufdiagramm, das den Ablauf eines Verfahrens zur Codierung mit variabler Wortlänge gemäß der vierten Ausführungsform darstellt. Da die Bildcodierungsvorrichtung der vierten Ausführungsform die gleiche Konfiguration wie die der ersten Ausführungsform hat, wird [Fig. 2](#) für die Beschreibung verwendet.

[0116] Das Bildcodierungsverfahren gemäß der vierten Ausführungsform unterscheidet sich von dem der zweiten Ausführungsform nur hinsichtlich der Funktionsweise des in [Fig. 2](#) gezeigten Codierers für Codes mit variabler Wortlänge **211**, und daher wird die Funktionsweise des Codierers für Codes mit variabler Wortlänge **211** bei der Codierung nachstehend unter Bezugnahme auf [Fig. 10](#) beschrieben. Hier werden bei der Codierung gemäß der vierten Ausführungsform erste und zweite Tabellen von Codes mit variabler Wortlänge verwendet. Für die erste Tabelle

von Codes mit variabler Wortlänge wird, wie bei der ersten bis dritten Ausführungsform, die in den [Fig. 19](#) bis [Fig. 22](#) dargestellte Tabelle verwendet. Für die zweite Tabelle von Codes mit variabler Wortlänge wird jedoch eine andere als die erste Tabelle von Codes mit variabler Wortlänge verwendet. Diese andere Tabelle besteht aus Gruppen mit Level-Werten, die nicht kleiner als zehn sind. Wie vorstehend dargelegt, wird bei der vierten Ausführungsform eine erste Tabelle von Codes mit variabler Wortlänge verwendet, in der die meisten Level-Werte kleiner als zehn sind, während die zweite verwendete Tabelle von Codes mit variabler Wortlänge die Level-Werte enthält, die gleich oder größer als zehn sind; das heißt, vorzugsweise wird eine zweite Tabelle von Codes mit variabler Wortlänge verwendet, die aus den Bezugsereignissen besteht, die nicht in der ersten Tabelle von Codes mit variabler Wortlänge enthalten sind.

[0117] Nach Initiierung des Prozesses im Schritt **1001** wird im Schritt **1002** ein j-tes Eingabe-Ereignis erhalten. Im Schritt **1003** wird im Codierer für Codes mit variabler Wortlänge **211** ein Vergleich zwischen dem Eingabe-Ereignis und einem in einer ersten Tabelle von Codes mit variabler Wortlänge enthaltenen Bezugsereignis durchgeführt. Dann wird im Schritt **1004** mit einem Vergleich festgestellt, ob ein Bezugsereignis mit einer passenden Gruppe (Last, Run, Level) vorhanden ist oder nicht. Wenn im Schritt **1004** festgestellt wird, dass dieses Bezugsereignis vorhanden ist, wird Schritt **1005** ausgeführt; wenn keines vorhanden ist, wird Schritt **1006** ausgeführt. Wenn jetzt Schritt **1005** ausgeführt wird, wird ein Code mit variabler Wortlänge, der dem Bezugsereignis entspricht, ausgegeben. Daran schließt sich Schritt **1010** an, der später beschrieben wird.

[0118] Wenn jedoch Schritt **1006** ausgeführt wird, wird das Eingabe-Ereignis mit einem in einer zweiten Tabelle von Codes mit variabler Wortlänge enthaltenen Bezugsereignis verglichen. Dann wird im Schritt **1007** durch Vergleichen festgestellt, ob ein Bezugsereignis mit einem passenden (Last, Run, Level) vorhanden ist oder nicht. Wenn im Schritt **1007** festgestellt wird, dass das Bezugsereignis vorhanden ist, wird Schritt **1008** ausgeführt, und wenn es nicht vorhanden ist, wird Schritt **1009** ausgeführt. Wenn jetzt Schritt **1008** ausgeführt wird, wird ein Code mit variabler Wortlänge erhalten, der dem Bezugsereignis entspricht, und ein eindeutig festgelegter Steuercode und ein zweiter Moduscode werden zu dem Code addiert und ausgegeben. Daran schließt sich Schritt **1010** an, der später beschrieben wird. In diesem Fall wird „0000011“, der mit dem in [Fig. 22](#) gezeigten ESC-Code identisch ist, als Steuercode verwendet und „0“ wird als zweiter Moduscode verwendet.

[0119] Wenn jedoch Schritt **1009** ausgeführt wird, wird das Eingabe-Ereignis dem in den [Fig. 23](#) dargestellten Codierungsverfahren unterzogen, um einen

Code mit fester Wortlänge zu erzeugen. Dann wird das Codierungsergebnis ausgegeben, das ein Code mit fester Wortlänge ist, zu dem ein Steuercode und ein erster Moduscode addiert werden. Dabei wird „1“ als erster Moduscode verwendet.

[0120] Wenn einer der Schritte **1005**, **1008** und **1009** ausgeführt wird, wird in einem nachfolgenden Schritt **1010** festgestellt, ob ein verarbeitetes eingegebenes Eingabe-Ereignis das letzte Eingabe-Ereignis ist oder nicht, und wenn es das letzte ist, wird die Codierung im Schritt **1011** beendet. Wenn es jedoch nicht das letzte ist, erfolgt der Rücksprung zum Schritt **1002**, nachdem im Schritt **1012** j um 1 erhöht worden ist, wodurch ein nächstes Eingabe-Ereignis auf die gleiche Weise verarbeitet wird. So wird das Verfahren bis zum letzten Koeffizienten ungleich Null fortgeführt.

[0121] Nachstehend wird die Decodierung der codierten Daten beschrieben, die bei dem in der vierten Ausführungsform erläuterten Bildcodierungsverfahren erhalten werden. Da die Konfiguration der Bilddecodierungsvorrichtung gemäß der vierten Ausführungsform mit der der ersten Ausführungsform identisch ist, wird [Fig. 4](#) für die Beschreibung verwendet.

[0122] [Fig. 11](#) ist ein Ablaufdiagramm, das den Ablauf eines Verfahrens zur Decodierung mit variabler Wortlänge gemäß der vierten Ausführungsform zeigt. Das Bilddecodierungsverfahren gemäß der vierten Ausführungsform unterscheidet sich von dem der ersten Ausführungsform nur hinsichtlich der Funktionsweise des in [Fig. 4](#) gezeigten Decodierers für Codes mit variabler Wortlänge **408**, und daher wird die Funktionsweise des Decodierers für Codes mit variabler Wortlänge **408** bei der Decodierung nachstehend unter Bezugnahme auf [Fig. 11](#) beschrieben.

[0123] Zunächst wird nach der Initiierung des Prozesses im Schritt **1101** ein Code mit variabler Wortlänge von quantisierten Transformationskoeffizienten, der das j-te zu verarbeitende Objekt ist, als Eingabe-Code mit variabler Wortlänge eingegeben. Im Schritt **1103** wird festgestellt, ob der Eingabe-Code mit variabler Wortlänge einen Steuercode enthält oder nicht, und wenn er keinen Steuercode enthält, wird Schritt **1104** ausgeführt; wenn er den Code enthält, wird Schritt **1105** ausgeführt. In dem Fall, dass Schritt **1104** ausgeführt wird, das heißt, wenn kein Steuercode enthalten ist, wird unter Verwendung einer ersten Tabelle von Codes mit variabler Wortlänge eine Gruppe von (Last, Run, Level), die dem eingegebenen Code mit variabler Wortlänge entspricht, ausgegeben. Daran schließt sich Schritt **1110** an, der später beschrieben wird.

[0124] Wenn jedoch im Schritt **1103** festgestellt wird, dass ein Steuercode enthalten ist, wird im Schritt **1105** entschieden, ob ein nächstes Bit nach

dem Steuercode „0“ oder „1“ ist. Wenn es „0“ ist, wird Schritt **1106** ausgeführt, und wenn es „1“ ist, wird Schritt **1108** ausgeführt.

[0125] Wenn Schritt **1106** ausgeführt wird, wird durch Verwendung der zweiten Tabelle von Codes mit variabler Wortlänge eine Gruppe (Last, Run, Level), die dem eingegebenen Code mit variabler Wortlänge entspricht, ausgegeben. Daran schließt sich Schritt **1110** an, der später beschrieben wird. Wenn jedoch Schritt **1108** ausgeführt wird, wird der Eingabe-Code mit variabler Wortlänge als Code mit fester Wortlänge decodiert, um die Koeffizienten zu regenerieren.

[0126] Wenn einer der Schritte **1104**, **1106** und **1108** ausgeführt wird, wird in einem nachfolgenden Schritt **1110** festgestellt, ob der verarbeitete Eingabe-Code mit variabler Wortlänge der letzte ist oder nicht, und wenn es der letzte ist, wird im Schritt **1111** die Codierung beendet. Wenn es jedoch nicht der letzte ist, erfolgt der Rücksprung zum Schritt **1102**, nachdem im Schritt **1112** j um 1 erhöht worden ist, wodurch ein nächster eingegebener Code mit variabler Wortlänge in der gleichen Weise verarbeitet wird. So wird das Verfahren bis zum letzten Code mit variabler Wortlänge fortgesetzt.

[0127] Wie vorstehend beschrieben, werden bei dem Bildcodierungsverfahren gemäß der vierten Ausführungsform eine erste und eine zweite Tabelle von Codes mit variabler Wortlänge verwendet, und wenn beim Vergleichen in den Schritten **1003** bis **1004** kein betreffendes Bezugsereignis in der ersten Tabelle von Codes mit variabler Wortlänge gefunden wird, kann ein Vergleich mit der zweiten Tabelle von Codes mit variabler Wortlänge durchgeführt werden. Dadurch wird der Anteil der Durchführung der Codierung mit fester Wortlänge im Schritt **1009** verringert, was zu einer besseren Gesamtkomprimierungsrate des Codierungsergebnisses führt. Bei der vierten Ausführungsform müssen mehrere Tabellen von Codes mit variabler Wortlänge erstellt werden, und indem die Tabellen so erstellt werden, dass sich die in einer der Tabellen enthaltenen Ereignisse von denen in den anderen Tabellen unterscheiden, kann die Codierungsleistung verbessert werden.

[0128] Da die Bildcodierungsvorrichtung der vierten Ausführungsform den Codierer für Codes mit variabler Wortlänge **211** aufweist, der die vorstehend beschriebene Codierung durchführt, kann ein Codierungsergebnis mit einer besseren Komprimierungsrate erhalten werden, gleichgültig, ob die Komprimierungsrate der Kompressionscodierung im Codierer **203** hoch oder niedrig ist.

[0129] Mit dem Bilddecodierungsverfahren und die Bilddecodierungsvorrichtung gemäß der vierten Ausführungsform können eine zweckmäßige Decodie-

rung entsprechend dem in der vorstehend beschriebenen Weise erhaltenen Codierungsergebnis durchgeführt werden, um ein wiederhergestelltes Bild zu erhalten.

[0130] Obwohl bei der vierten Ausführungsform zwei Tabellen von Codes mit variabler Wortlänge verwendet werden, kann eine beliebige Anzahl von Tabellen von Codes mit variabler Wortlänge erstellt werden und es kann eine entsprechende Anzahl von Moduscodes zur Unterscheidung der Tabellen von Codes mit variabler Wortlänge verwendet werden. Außerdem können Tabellen von Codes mit variabler Wortlänge nach Quantisierungsskalen unterschieden werden.

Ausführungsform 5

[0131] Bei einem Bildcodierungsverfahren gemäß einer fünften Ausführungsform wird ein Ereignis, das ein zu codierendes Objekt ist, durch Division transformiert und dann mit einer Tabelle von Codes mit variabler Wortlänge verglichen.

[0132] [Fig. 12](#) ist ein Ablaufdiagramm, das den Ablauf eines Verfahrens zur Codierung mit variabler Wortlänge gemäß der fünften Ausführungsform darstellt. Da die Bildcodierungsvorrichtung der fünften Ausführungsform die gleiche Konfiguration wie die der ersten Ausführungsform hat, wird [Fig. 2](#) für die Beschreibung verwendet.

[0133] Das Bildcodierungsverfahren gemäß der fünften Ausführungsform unterscheidet sich von dem der ersten Ausführungsform nur hinsichtlich der Funktionsweise des in [Fig. 2](#) gezeigten Codierers für Codes mit variabler Wortlänge **211**, und daher wird die Funktionsweise des Codierers für Codes mit variabler Wortlänge **211** bei der Codierung nachstehend unter Bezugnahme auf [Fig. 12](#) beschrieben.

[0134] Nach Initiierung des Prozesses im Schritt **1201** wird im Schritt **1202** ein aus (Last, Run, Level) bestehendes j -tes Eingabe-Ereignis erhalten. Dann wird im Schritt **1203** das Eingabe-Ereignis durch Division transformiert. Bei der Transformation wird der im Eingabe-Ereignis enthaltene Level-Wert durch einen anhand einer Quantisierungsskala bestimmten Divisor dividiert, um einen Quotienten des Levels zu erhalten. Die [Fig. 13](#) sind Diagramme, die den Zusammenhang zwischen Quantisierungsskala und Divisor zeigen. [Fig. 13\(a\)](#) zeigt ein Festlegungsbeispiel für diesen Zusammenhang und gibt an, dass der Divisor mit 2 angesetzt wird, wenn eine Quantisierungsskala von 1 bis 7 reicht, und dass der Divisor 1 ist, wenn die Quantisierungsskala gleich oder größer als 8 ist. Das heißt, dass keine Division durchgeführt wird, wenn eine Quantisierungsskala gleich oder größer als 8 ist. [Fig. 13\(b\)](#) zeigt ein weiteres Festlegungsbeispiel für diesen Zusammenhang, und wie in

der Figur gezeigt, kann die Quantisierungsskala in drei Klassen eingeteilt werden und für jede Klasse kann ein Divisor festgelegt werden.

[0135] Die Quantisierung ist im Grunde eine Division, und der Wert der quantisierten Transformationskoeffizienten steigt mit fallender Quantisierungsskala. Folglich ist beim Erzeugen eines Ereignisses der in dem Ereignis enthaltene Level-Wert oftmals hoch. Die in den [Fig. 19](#) bis [Fig. 22](#) gezeigte Tabelle von Codes mit variabler Wortlänge wird so erstellt, dass sie für Koeffizienten mit einer relativ hohen Komprimierungsrate geeignet ist, und sie enthält zahlreiche Ereignisse mit niedrigen Level-Werten. Daher werden der Anteil der in der Tabelle enthaltenen passenden Bezugsereignisse und auch der Anteil der Durchführung der Codierung mit variabler Wortlänge verringert, sodass sich die Codierungsleistung verschlechtert. Wenn also bei der fünften Ausführungsform die Quantisierungsskala klein ist, wird der Level dividiert, um ein transformiertes Eingabe-Ereignis mit einem niedrigen Level-Wert zu erzeugen, das zum Vergleichen verwendet wird. Dadurch wird der Anteil der Durchführung der Codierung mit variabler Wortlänge verbessert und die Möglichkeit der Zuweisung kurzer Codes erhöht.

[0136] Nach Erhalt des Quotienten des Levels wird im Schritt **1204** ein Vergleich zwischen (Last, Run, Quotient des Levels) des transformierten Eingabe-Ereignisses und dem in einer Tabelle von Codes mit variabler Wortlänge enthaltenen Bezugsereignis durchgeführt, und in einem nachfolgenden Schritt **1205** wird eine Entscheidung zum Ergebnis des Vergleichs getroffen. Wenn im Schritt **1205** entschieden wird, dass das Bezugsereignis vorhanden ist, wird Schritt **1206** ausgeführt, und wenn es nicht vorhanden ist, wird Schritt **1208** ausgeführt.

[0137] Wenn Schritt **1207** ausgeführt wird, wird der Code mit variabler Wortlänge für das betreffende Bezugsereignis ausgegeben, und in einem nachfolgenden Schritt **1208** wird der Rest des Levels einer Codierung mit fester Wortlänge unterzogen, und das Codierungsergebnis wird als Eingabe-Hilfscode ausgegeben. Die Codelänge des übrigen Levels wird entsprechend dem Divisor geändert. Beispielsweise ist bei der in [Fig. 13\(a\)](#) gezeigten Festlegung die Bitlänge des Rests mit 1 festgelegt, wenn der Divisor 2 ist, und die Bitlänge des Rests ist mit 0 festgelegt, wenn der Divisor 1 ist. Mit anderen Worten, wenn der Divisor 1 ist, wird der Rest nicht codiert. Wenn die Schritte **1207** bis **1208** ausgeführt werden, werden ein Code mit variabler Wortlänge und ein Eingabe-Hilfscode als Codierungsergebnis ausgegeben.

[0138] Wenn jedoch Schritt **1206** ausgeführt wird, wird das Eingabe-Ereignis einer Codierung mit fester Wortlänge unterzogen, um einen Code mit fester Wortlänge zu erzeugen. Dann wird das Codierungs-

ergebnis ausgegeben, das der Code mit fester Wortlänge ist, zu dem ein Steuercode addiert wird.

[0139] Nach Ausführung der Schritte **1207** bis **1208** oder von Schritt **1206** wird in einem nachfolgenden Schritt **1209** festgestellt, ob das verarbeitete Eingabe-Ereignis das letzte Eingabe-Ereignis ist oder nicht, und wenn es das letzte ist, wird im Schritt **1210** die Codierung beendet. Wenn es jedoch nicht das letzte ist, erfolgt der Rücksprung zum Schritt **1202**, nachdem im Schritt **1211** j um 1 erhöht worden ist, wodurch ein nächstes Eingabe-Ereignis in der gleichen Weise verarbeitet wird. So wird das Verfahren bis zum letzten Koeffizienten ungleich Null fortgesetzt.

[0140] Nachstehend wird die Decodierung der codierten Daten beschrieben, die bei dem in der fünften Ausführungsform erläuterten Bildcodierungsverfahren erhalten werden. Da die Konfiguration der Bilddecodierungsvorrichtung gemäß der fünften Ausführungsform mit der der ersten Ausführungsform identisch ist, wird [Fig. 4](#) für die Beschreibung verwendet.

[0141] [Fig. 14](#) ist ein Ablaufdiagramm, das den Ablauf eines Verfahrens zur Decodierung mit variabler Wortlänge gemäß der fünften Ausführungsform zeigt. Das Bilddecodierungsverfahren gemäß der fünften Ausführungsform unterscheidet sich von dem der ersten Ausführungsform nur hinsichtlich der Funktionsweise des in [Fig. 4](#) gezeigten Decodierers für Codes mit variabler Wortlänge **408**, und daher wird die Funktionsweise des Decodierers für Codes mit variabler Wortlänge **408** bei der Decodierung nachstehend unter Bezugnahme auf [Fig. 14](#) beschrieben.

[0142] Zunächst wird nach der Initiierung des Prozesses im Schritt **1401** ein Code mit variabler Wortlänge von quantisierten Transformationskoeffizienten, der das j-te zu verarbeitende Objekt ist, als Eingabe-Code mit variabler Wortlänge im Schritt **1402** eingegeben. Da das den Eingabe-Hilfscode enthaltende Codierungsergebnis wie bei der Bildcodierung der fünften Ausführungsform beschrieben erhalten wird, wird nun zusammen mit dem Code mit variabler Wortlänge auch der Eingabe-Hilfscode als zu verarbeitendes Objekt eingegeben. Im Schritt **1403** wird festgestellt, ob der Eingabe-Code mit variabler Wortlänge einen Steuercode enthält oder nicht, und wenn er einen Steuercode enthält, wird Schritt **1404** ausgeführt; wenn er keinen Code enthält, wird Schritt **1405** ausgeführt. In dem Fall, dass Schritt **1404** ausgeführt wird, das heißt, wenn der Steuercode enthalten ist, wird der Eingabe-Code mit variabler Wortlänge als Code mit fester Wortlänge decodiert, um Koeffizienten zu regenerieren und auszugeben. Daran schließt sich Schritt **1408** an, der später beschrieben wird.

[0143] Wenn jedoch Schritt **1405** ausgeführt wird, das heißt, wenn der Steuercode nicht enthalten ist,

wird unter Verwendung einer ersten Tabelle von Codes mit variabler Wortlänge eine Gruppe (Last, Run, Level), die dem Eingabe-Code mit variabler Wortlänge entspricht, erhalten. Dann wird Schritt **1406** ausgeführt, bei dem eine Transformation durch Multiplikation des erhaltenen Level-Werts mit einem vorgegebenen Wert durchgeführt wird. Der vorgegebene Wert wird anhand einer Quantisierungsskala festgelegt, und in diesem Fall wird der in den **Fig. 13** angegebene Wert entsprechend dem Codierungsverfahren verwendet. Dann wird im Schritt **1407** der Eingabe-Hilfscode einer Codierung mit fester Wortlänge unterzogen, um den Rest des Levels zu erhalten, und der Rest wird zu dem Produkt addiert, das durch die Transformation im Schritt **1406** erhalten wird, um den richtigen Level-Wert zu erhalten, und die Gruppe (Last, Run, Level), in der dieser Level-Wert verwendet wird, wird als Decodierungsergebnis ausgegeben.

[0144] Nachdem einer der Schritte **1404** oder **1405** bis **1407** ausgeführt worden ist, wird in einem nachfolgenden Schritt **1408** festgestellt, ob der verarbeitete Eingabe-Code mit variabler Wortlänge der letzte Eingabe-Code mit variabler Wortlänge ist oder nicht, und wenn er der letzte ist, wird im Schritt **1409** die Codierung beendet. Wenn er jedoch nicht der letzte ist, erfolgt der Rücksprung zum Schritt **1402**, nachdem im Schritt **1410** j um 1 erhöht worden ist, wodurch ein nächster Eingabe-Code mit variabler Wortlänge in der gleichen Weise verarbeitet wird. So wird das Verfahren bis zum letzten Code mit variabler Wortlänge fortgesetzt.

[0145] Wie vorstehend beschrieben, wird bei dem Bildcodierungsverfahren gemäß der fünften Ausführungsform, wie im Schritt **1203** von **Fig. 12** gezeigt, eine Transformation durchgeführt, bei der der Level-Wert, der im Eingabe-Ereignis enthalten ist, das ein zu verarbeitendes Objekt ist, durch eine anhand einer Quantisierungsskala bestimmte Zahl dividiert wird, und im Schritt **1204** wird ein Vergleich mit dem transformierten Eingabe-Ereignis durchgeführt, das der Transformation unterzogen worden ist. Somit wird der Anteil der in der Tabelle von Codes mit variabler Wortlänge enthaltenen passenden Bezugsereignisse erhöht, der Anteil der Durchführung der Codierung mit fester Wortlänge im Schritt **1206** wird gesenkt und die Möglichkeit der Zuweisung kurzer Codes bei der Codierung mit variabler Wortlänge wird durch Verwendung eines Codes mit variabler Wortlänge erhöht, was zu einer besseren Codierungsleistung führt.

[0146] Da die Bildcodierungsvorrichtung der fünften Ausführungsform den Codierer für Codes mit variabler Wortlänge **211** aufweist, der die vorstehend beschriebene Codierung durchführt, kann durch Verwendung einer einzigen Tabelle von Codes mit variabler Wortlänge ein Codierungsergebnis mit einer

besseren Komprimierungsrate erhalten werden, gleichgültig, ob die Komprimierungsrate der Kompressionscodierung im Codierer **203** hoch oder niedrig ist.

[0147] Mit dem Bilddecodierungsverfahren und die Bilddecodierungsvorrichtung gemäß der fünften Ausführungsform kann eine zweckmäßige Decodierung entsprechend dem in der vorstehend beschriebenen Weise erhaltenen Codierungsergebnis durchgeführt werden, um ein wiederhergestelltes Bild zu erhalten.

[0148] Obwohl bei der fünften Ausführungsform beschrieben wird, dass die Transformation an dem in einem Ereignis enthaltenen Level durchgeführt wird, kann die Transformation anstatt am Level auch am Run erfolgen.

Ausführungsform 6

[0149] Bei einem Bildcodierungsverfahren gemäß einer sechsten Ausführungsform wird die Transformation nicht am Eingabe-Ereignis, sondern am Bezugsereignis durchgeführt, und anschließend wird der Vergleich durchgeführt.

[0150] **Fig. 15** ist ein Ablaufdiagramm, das den Ablauf eines Verfahrens zur Codierung mit variabler Wortlänge gemäß der sechsten Ausführungsform zeigt. Da die Bildcodierungsvorrichtung der sechsten Ausführungsform die gleiche Konfiguration wie die der ersten Ausführungsform hat, wird **Fig. 2** für die Beschreibung verwendet.

[0151] Das Bildcodierungsverfahren gemäß der sechsten Ausführungsform unterscheidet sich von dem der ersten Ausführungsform nur hinsichtlich der Funktionsweise des in **Fig. 2** gezeigten Codierers für Codes mit variabler Wortlänge **211**, und daher wird die Funktionsweise des Codierers für Codes mit variabler Wortlänge **211** bei der Codierung nachstehend unter Bezugnahme auf **Fig. 15** beschrieben.

[0152] Nach Initiierung des Prozesses im Schritt **1501** wird im Schritt **1502** ein aus (Last, Run, Level) bestehendes j -tes Eingabe-Ereignis erhalten. Eine bei der Quantisierung verwendete Quantisierungsskala wird ebenfalls eingegeben. Im Schritt **1503** wird ein i -tes Bezugsereignis aus einer Tabelle von Codes mit variabler Wortlänge geholt, und im Schritt **1504** wird das Bezugsereignis transformiert, um ein transformiertes Bezugsereignis zu erzeugen.

[0153] Bei der sechsten Ausführungsform wird als Transformation ein Offset, der entsprechend der im Schritt **1502** eingegebenen Quantisierungsskala festgelegt wird, zu dem in dem Bezugsereignis enthaltenen Level-Wert addiert. Dabei erfolgt die Festlegung so, dass Offset = 5 ist, wenn die Quantisierungsskala 1 oder 2 ist, dass Offset = 4 ist, wenn die Quantisie-

rungsskala 3 oder 4 ist, dass Offset = 3 ist, wenn die Quantisierungsskala 5 oder 6 ist, dass Offset = 2 ist, wenn die Quantisierungsskala 7 oder 8 ist, dass Offset = 1 ist, wenn die Quantisierungsskala 9 oder 10 ist, und dass Offset = 0 ist, wenn die Quantisierungsskala gleich oder größer als 11 ist. Wenn beispielsweise in dem Fall, dass ein Ereignis als zu codierendes Objekt (Last, Run, Level) = (0, 1, 6) ist, das Bezugsereignis nicht transformiert wird, bedeutet das, dass ein passendes Bezugsereignis in den in [Fig. 19](#) gezeigten Bezugsereignissen gefunden wird. Es wird ein diesem entsprechender Code verwendet, und das erhaltene Codierungsergebnis umfasst 13 Bit. Wenn jedoch die Quantisierungsskala 4 ist und das gleiche Ereignis als Objekt verwendet wird, werden alle Bezugsereignisse einer Transformation unterzogen, bei der 4 zum Level-Wert addiert wird. Es wird ein Code mit variabler Wortlänge, der dem in [Fig. 19](#) angegebenen Bezugsereignis (0, 1, 2) entspricht, verwendet und so werden 7 Bit erhalten, was bedeutet, dass im Gegensatz zu dem Fall, dass die vorstehend beschriebene Transformation nicht durchgeführt wird, 6 Bit gerettet werden. Somit ist das Addieren eines Offsets zum Level des Bezugsereignisses mit dem Verfahren identisch, bei dem ein Offset vom Level eines Ereignisses (Last, Run, Level), das ein zu codierendes Objekt ist, deduziert wird und mit jeder Gruppe einer Tabelle von Codes mit variabler Wortlänge verglichen wird.

[0154] Da, wie bei der fünften Ausführungsform beschrieben, der Wert eines quantisierten Koeffizienten mit abnehmender Quantisierungsskala steigt, steigt auch die Häufigkeit, mit der ein Ereignis mit einem großen Level-Wert auftritt. Andererseits ist die in den [Fig. 19](#) bis [Fig. 22](#) gezeigte Tabelle von Codes mit variabler Wortlänge für Koeffizienten mit einer hohen Komprimierungsrate geeignet und sie enthält zahlreiche Ereignisse mit niedrigen Level-Werten. Somit kann durch Addieren eines Offsets zu einem in einem Bezugsereignis enthaltenen Level-Wert der Anteil der Durchführung der Codierung mit variabler Wortlänge unter Verwendung kurzer Codes erhöht werden und dadurch die Komprimierungsrate verbessert werden.

[0155] Im Schritt **1505** wird aufgrund des Ergebnisses des Vergleichs im Schritt **1504** eine Entscheidung getroffen. Wenn sich bei der Entscheidung im Schritt **1505** eine Übereinstimmung zeigt, wird nach Schritt **1506** Schritt **1507** ausgeführt, wo ein Code mit variabler Wortlänge, der dem i-ten Bezugsereignis entspricht, ausgegeben wird. Daran schließt sich Schritt **1509** an, der später beschrieben wird. Wenn jedoch keine Übereinstimmung gefunden wird, wird Schritt **1507** ausgeführt.

[0156] Im Schritt **1507** wird festgestellt, ob das i-te Bezugsereignis das letzte Bezugsereignis ist, das in der Tabelle von Codes mit variabler Wortlänge ent-

halten ist. Wenn es das letzte Bezugsereignis ist, wird Schritt **1508** ausgeführt. Wenn es nicht das letzte Bezugsereignis ist, erfolgt der Rücksprung zum Schritt **1503**, nachdem im Schritt **1512** i um 1 erhöht worden ist, und die nachfolgenden Schritte, die mit der Entscheidung im Schritt **1507** enden, werden nochmals ausgeführt.

[0157] Mit den Prozessen in den Schritten **1503** bis **1507** wird der Vergleich zwischen einem j-ten Eingabe-Ereignis und einem in der Tabelle von Codes mit variabler Wortlänge enthaltenen Bezugsereignis so lange wiederholt, bis sich bei der Entscheidung im Schritt **1505** eine Übereinstimmung zwischen beiden zeigt oder im Schritt **1507** entschieden wird, dass das verarbeitete Ereignis das letzte ist. Mit anderen Worten, der Vergleich wird so lange fortgesetzt, bis die Codierung mit variabler Wortlänge durchgeführt ist oder das letzte Ereignis der Tabelle von Codes mit variabler Wortlänge verglichen ist.

[0158] Wenn im Schritt **1507** entschieden wird, dass das verarbeitete Ereignis das letzte ist, das heißt, wenn der Vergleich mit der Tabelle für das j-te Eingabe-Ereignis beendet ist, wird festgestellt, dass keine Codierung mit variabler Wortlänge durchgeführt werden kann, und im Schritt **1508** wird das Eingabe-Ereignis einer Codierung mit fester Wortlänge unterzogen, um das Codierungsergebnis mit einem Steuercode auszugeben. Daran schließt sich Schritt **1509** an.

[0159] Im Schritt **1509**, der nach Schritt **1506** oder **1508** ausgeführt wird, wird festgestellt, ob ein verarbeitetes Eingabe-Ereignis das letzte Eingabe-Ereignis ist oder nicht. Wenn es das letzte ist, wird im Schritt **1510** die Codierung beendet. Wenn es jedoch nicht das letzte ist, erfolgt der Rücksprung zum Schritt **1502**, nachdem im Schritt **1511** j um 1 erhöht worden ist, wodurch das nächste Eingabe-Ereignis in der gleichen Weise verarbeitet wird. So wird das Verfahren bis zum letzten Koeffizienten ungleich Null fortgesetzt.

[0160] [Fig. 16](#) ist ein Blockdiagramm, das die innere Konfiguration des Codierers für Codes mit variabler Wortlänge (**211** in [Fig. 2](#)) zeigt, der in einer Bildcodierungsvorrichtung gemäß der sechsten Ausführungsform der vorliegenden Erfindung verwendet wird. Wie in der Figur gezeigt, weist der Codierer für Codes mit variabler Wortlänge eine Tabelle von Codes mit variabler Wortlänge **1601**, eine Level-Transformations-einheit **1602**, einen Komparator **1603**, einen Codierer für Codes mit fester Wortlänge **1604** und einen Schalter **1605** auf, und er empfängt ein Eingabe-Ereignis und eine Quantisierungsskala als Eingangsdaten und gibt ein Codierungsergebnis aus.

[0161] Die Tabelle von Codes mit variabler Wortlänge **1601** wird in den [Fig. 19](#) bis [Fig. 22](#) gezeigt. Die

Level-Transformationseinheit **1602** empfängt eine Quantisierungsskala und ein Bezugsereignis als Eingangsdaten und transformiert den Level-Wert des Bezugsereignisses entsprechend der Quantisierungsskala, um ein transformiertes Bezugsereignis zu erzeugen. Der Komparator **1603** führt einen Vergleich zwischen dem Eingabe-Ereignis und dem transformierten Bezugsereignis durch und gibt ein Steuersignal aus, das dem Ergebnis des Vergleichs entspricht. Der Codierer für Codes mit fester Wortlänge **1604** unterzieht das Eingabe-Ereignis beim Eingeben einer Codierung mit fester Wortlänge und gibt das Codierungsergebnis aus. Der Schalter **1605** führt das Schalten entsprechend den Ausgangssignalen vom Komparator **1603** durch und gibt als Codierungsergebnis entweder Ausgangssignale von der Tabelle von Codes mit variabler Wortlänge **1601** oder Ausgangssignale vom Codierer für Codes mit fester Wortlänge **1604** aus.

[0162] Nachstehend wird die Funktionsweise des so aufgebauten Codierers für Codes mit variabler Wortlänge entsprechend dem Ablaufdiagramm von [Fig. 15](#) beschrieben. Wenn im Schritt **1502** des Ablaufdiagramms von [Fig. 15](#) eine Quantisierungsskala und ein (j-tes) Eingabe-Ereignis in den Codierer für Codes mit variabler Wortlänge (**211** in [Fig. 2](#)) eingegeben werden, wird die Quantisierungsskala über eine Leitung **1611** in die Level-Transformationseinheit **1602** eingegeben und das Eingabe-Ereignis wird über eine Leitung **1612** in den Komparator **1603** eingegeben. Im Schritt **1503** des Ablaufdiagramms von [Fig. 15](#) leitet die Level-Transformationseinheit **1602** über eine Leitung **1613** ein (i-tes) Bezugsereignis aus der Tabelle von Codes mit variabler Wortlänge ab, transformiert den in dem Bezugsereignis enthaltenen Level-Wert unter Verwendung der eingegebenen Quantisierungsskala und gibt das resultierende transformierte Bezugsereignis an den Komparator **1603** aus.

[0163] Im Schritt **1505** des Ablaufdiagramms von [Fig. 15](#) führt der Komparator **1603** einen Vergleich durch und trifft eine Entscheidung. Der Komparator **1603** führt den Vergleich zwischen dem Eingabe-Ereignis und dem transformierten Bezugsereignis, die eingegeben werden, durch und entscheidet, ob diese Ereignisse miteinander übereinstimmen oder nicht. Wenn eine Übereinstimmung festgestellt wird, sendet der Komparator **1603** ein Steuersignal über eine Leitung **1615** an die Tabelle von Codes mit variabler Wortlänge **1601**, sodass ein Code, der dem i-ten Bezugsereignis entspricht, ausgegeben wird, und sendet zugleich ein Steuersignal über eine Leitung **1617** an den Schalter **1605**, sodass das Ausgangssignal von der Tabelle von Codes mit variabler Wortlänge **1601** vom Schalter **1605** ausgegeben wird. Somit wird, wenn im Schritt **1505** eine Übereinstimmung festgestellt wird, im Schritt **1506** des Ablaufdiagramms von [Fig. 15](#) ein Code mit variabler Wortlänge

ge von der Ausgabeleitung **1620** des Schalters **1605** ausgegeben.

[0164] Wenn jedoch beim Vergleichen mit dem Komparator **1603** festgestellt wird, dass diese Ereignisse nicht übereinstimmen, wird in Reaktion auf ein über die Leitung **1615** gesendetes Steuersignal ein nächstes (i + 1-tes) Bezugsereignis von der Tabelle von Codes mit variabler Wortlänge **1601** in die Level-Transformationseinheit **1602** eingegeben. In dem Ablaufdiagramm von [Fig. 15](#) werden die Schritte ab Schritt **1503** nochmals ausgeführt, nachdem i im Schritt **1512** erhöht worden ist.

[0165] Wenn die Entscheidung im Schritt **1507** dazu führt, dass der Vergleich für das letzte Ereignis in der Tabelle von Codes mit variabler Wortlänge beendet wird, gibt der Komparator **1603** über eine Leitung **1618** das (j-te) Eingabe-Ereignis an den Codierer für Codes mit fester Wortlänge **1604** aus und sendet gleichzeitig ein Steuersignal über die Leitung **1617** an den Schalter **1605**, sodass das Ausgangssignal vom Codierer für Codes mit fester Wortlänge **1604** als Code vom Schalter **1605** verwendet wird.

[0166] Im Schritt **1508** des Ablaufdiagramms von [Fig. 15](#) codiert der Codierer für Codes mit fester Wortlänge **1604** das eingegebene Eingabe-Ereignis und gibt das erhaltene Codierungsergebnis über eine Leitung **1619** an den Schalter **1605** aus. Entsprechend dem Steuersignal vom Komparator **1603** wird über die Leitung **1620** ein Code mit fester Wortlänge vom Schalter **1605** ausgegeben.

[0167] Nach der Ausgabe von der Leitung **1620** wird ein nächstes Eingabe-Ereignis über die Leitung **1612** eingegeben, wodurch die Prozesse am Eingabe-Ereignis im Ablaufdiagramm von [Fig. 15](#) nochmals ausgeführt werden.

[0168] Auf diese Weise erfolgt die Bildcodierung gemäß der sechsten Ausführungsform der vorliegenden Erfindung, während die Decodierung der durch diese Bildcodierung erhaltenen codierten Daten in der in [Fig. 4](#) gezeigten Bilddecodierungsvorrichtung durchgeführt wird. Das Bilddecodierungsverfahren gemäß der sechsten Ausführungsform unterscheidet sich von dem der ersten Ausführungsform nur hinsichtlich der Funktionsweise des in [Fig. 4](#) gezeigten Decodierers für Codes mit variabler Wortlänge **408**, und daher wird die Funktionsweise des Decodierers für Codes mit variabler Wortlänge **408** bei der Decodierung nachstehend unter Bezugnahme auf [Fig. 17](#) beschrieben, die die innere Konfiguration des Decodierers zeigt.

[0169] Wie in [Fig. 17](#) gezeigt, weist der Decodierer für Codes mit variabler Wortlänge eine Tabelle von Codes mit variabler Wortlänge **1701**, einen Komparator **1702**, eine Level-Umkehrtransformationseinheit

1703, einen Decodierer für Codes mit fester Wortlänge **1704** und einen Schalter **1705** auf, und er empfängt einen Eingabe-Code mit variabler Wortlänge und eine Quantisierungsskala als Eingangssignale und gibt das aus der Decodierung erhaltene Regenerierungsergebnis aus.

[0170] Die Tabelle von Codes mit variabler Wortlänge **1701** wird in den [Fig. 19](#) bis [Fig. 22](#) gezeigt. Der Komparator **1702** führt einen Vergleich zwischen einem Eingabe-Code mit variabler Wortlänge und einem Bezugscode mit variabler Wortlänge durch und gibt ein Steuersignal aus, das dem Ergebnis des Vergleichs entspricht. Die Level-Umkehrtransformationseinheit **1703** empfängt eine Quantisierungsskala und ein Bezugsereignis als Eingangssignale und transformiert den im Bezugsereignis enthaltenen Level-Wert entsprechend der Quantisierungsskala, um ein transformiertes Bezugsereignis zu erzeugen. Der Decodierer für Codes mit fester Wortlänge **1704** unterzieht den Eingabe-Code mit variabler Wortlänge beim Eingeben einer Decodierung mit fester Wortlänge und gibt das Decodierungsergebnis aus. Der Schalter **1705** führt das Schalten entsprechend den Ausgangssignalen vom Komparator **1702** durch und gibt als Regenerierungsergebnis entweder Ausgangssignale von der Tabelle von Codes mit variabler Wortlänge **1701** oder Ausgangssignale vom Decodierer für Codes mit fester Wortlänge **1704** aus.

[0171] Nachstehend wird die Funktionsweise des so aufgebauten Decodierers für Codes mit variabler Wortlänge beschrieben. Wenn eine Quantisierungsskala und ein Eingabe-Code mit variabler Wortlänge in den Decodierer für Codes mit variabler Wortlänge (**408** in [Fig. 4](#)) eingegeben werden, wird die Quantisierungsskala über eine Leitung **1711** in die Level-Umkehrtransformationseinheit **1703** eingegeben und der Eingabe-Code mit variabler Wortlänge wird über eine Leitung **1712** in den Komparator **1702** eingegeben. Der Komparator **1702** leitet über eine Leitung **1713** einen Bezugscode mit variabler Wortlänge von der Tabelle von Codes mit variabler Wortlänge **1701** ab und vergleicht den Eingabe-Code mit variabler Wortlänge mit dem Bezugscode mit variabler Wortlänge.

[0172] Wenn der Vergleich zeigt, dass diese Codes übereinstimmen, sendet der Komparator **1702** ein Steuersignal über eine Leitung **1714** an die Tabelle von Codes mit variabler Wortlänge **1701**, sodass ein Bezugsereignis, das dem Bezugscode mit variabler Wortlänge entspricht, ausgegeben wird, und sendet gleichzeitig ein Steuersignal über eine Leitung **1717** an den Schalter **1705**, sodass der Code aus der Tabelle von Codes mit variabler Wortlänge **1701** vom Schalter **1705** ausgegeben wird.

[0173] Aus der Tabelle von Codes mit variabler Wortlänge wird ein zu transformierendes Bezugs-

ereignis über eine Leitung **1715** in die Level-Umkehrtransformationseinheit **1703** eingegeben. Die Level-Umkehrtransformationseinheit **1703** führt entsprechend dem Level, der in dem auf der Quantisierungsskala beruhenden Bezugsereignis enthalten ist, die für die Codierung durchgeführte Transformation in Umkehrung aus. Bei der sechsten Ausführungsform wird ein Offset, der entsprechend einer Quantisierungsskala festgelegt wird, zu dem in einem Bezugsereignis enthaltenen Level-Wert addiert.

[0174] Im Schritt **1503** des Ablaufdiagramms von [Fig. 15](#) leitet die Level-Transformationseinheit **1602** über eine Leitung **1613** ein (i-tes) Bezugsereignis aus der Tabelle von Codes mit variabler Wortlänge ab, transformiert den in dem Bezugsereignis enthaltenen Level-Wert unter Verwendung der eingegebenen Quantisierungsskala und gibt das resultierende transformierte Bezugsereignis an den Komparator **1603** aus. Das geschieht so, dass Offset = 5 ist, wenn die Quantisierungsskala 1 oder 2 ist, dass Offset = 4 ist, wenn die Quantisierungsskala 3 oder 4 ist, dass Offset = 3 ist, wenn die Quantisierungsskala 5 oder 6 ist, dass Offset = 2 ist, wenn die Quantisierungsskala 7 oder 8 ist, dass Offset = 1 ist, wenn die Quantisierungsskala 9 oder 10 ist, und dass Offset = 0 ist, wenn die Quantisierungsskala gleich oder größer als 11 ist. Wenn beispielsweise ein Eingabe-Code mit variabler Wortlänge „0101000“ ist, wird (Last, Run, Level) = (0, 1, 2) als Bezugsereignis aus der in [Fig. 19](#) gezeigten Tabelle abgeleitet. Die Level-Umkehrtransformationseinheit **1703** addiert 4 zu dem in diesem eingegebenen Bezugsereignis enthaltenen Level, wenn die eingegebene Quantisierungsskala 4 ist, und gibt das erhaltene Ergebnis (0, 1, 6) über eine Leitung **1716** an den Schalter **1705** aus. Dann gibt entsprechend dem Steuersignal vom Komparator **1702** der Schalter **1705** dieses Ergebnis als Regenerierungsergebnis über die Leitung **1720** aus. Wenn sich bei der Entscheidung eine Übereinstimmung zeigt, wird ein Code mit variabler Wortlänge von der Ausgabeleitung **1720** des Schalters **1705** ausgegeben.

[0175] Wenn jedoch kein passender Code mit variabler Wortlänge beim Vergleichen durch den Komparator **1702** gefunden wird, gibt der Komparator **1702** den Eingabe-Code mit variabler Wortlänge über eine Leitung **1718** an den Decodierer für Codes mit fester Wortlänge **1704** aus und sendet gleichzeitig ein Steuersignal über die Leitung **1717** an den Schalter **1705**, sodass ein Code vom Decodierer für Codes mit fester Wortlänge **1704** vom Schalter **1705** ausgegeben wird. Dadurch wird das Ergebnis der Decodierung mit fester Wortlänge von der Ausgabeleitung **1720** des Schalters **1705** ausgegeben.

[0176] Auf diese Weise wird nach dem Bildcodierungsverfahren gemäß der sechsten Ausführungsform die im Schritt **1502** von [Fig. 15](#) gezeigte Trans-

formation an einem in einer Tabelle von Codes mit variabler Wortlänge enthaltenen Bezugseignis durchgeführt. Dadurch werden der Anteil der Durchführung der Codierung mit variabler Wortlänge und der Anteil der Zuweisung kurzer Codes erhöht, wie es bei der ersten Ausführungsform der Fall ist, wo die gleiche Transformation am Eingabe-Ereignis durchgeführt wird, was zu einer besseren Codierungsleistung führt.

[0177] Da die Bildcodierungsvorrichtung gemäß der sechsten Ausführungsform einen Codierer für Codes mit variabler Wortlänge **211** aufweist, der die in **Fig. 16** gezeigte innere Konfiguration hat und das vorstehend beschriebene Codierungsverfahren durchführt, kann ein Codierungsergebnis mit einer besseren Komprimierungsrate mit einer einzigen Tabelle von Codes mit variabler Wortlänge erhalten werden, gleichgültig, ob die Komprimierungsrate der Kompressionscodierung durch den Codierer **203** hoch oder niedrig ist.

[0178] Außerdem kann mit dem Bilddecodierungsverfahren und der Bilddecodierungsvorrichtung gemäß der sechsten Ausführungsform eine zweckmäßige Decodierung entsprechend dem auf die vorstehend beschriebene Weise erhaltenen Codierungsergebnis durchgeführt werden, um ein wiederhergestelltes Bild zu erhalten.

[0179] Obwohl bei der sechsten Ausführungsform ein Verfahren der Addition des Offsets als Transformationsverfahren für das Bezugseignis verwendet wird, ist der Prozess nicht auf dieses Verfahren beschränkt und es können auch andere Funktionen verwendet werden. Die vorstehend beschriebene Wirkung kann beispielsweise durch folgende Verfahren erzielt werden: Ändern eines Vielfachen in einen Level-Wert entsprechend der Quantisierungsskala, Durchführung einer quadratischen Funktion und Ändern des dynamischen Bereiches eines Levels.

[0180] Wie in der Beschreibung der Transformation des Eingabe-Ereignisses bei der ersten Ausführungsform und bei anderen Ausführungsformen dargelegt, kann die Transformation statt am Level-Wert des Bezugseignisses auch am Run-Wert durchgeführt werden.

[0181] Wie in den Ausführungsformen der vorliegenden Erfindung beschrieben, sind bei den einzelnen Ausführungsformen die unter Bezugnahme auf das jeweilige Ablaufdiagramm dargelegten Abläufe stets nur ein Beispiel, und es können auch andere Abläufe, die die gleichen Codierungs- und Decodierungsverfahren darstellen, verwendet werden.

[0182] Die Bildcodierungsvorrichtungen und Bilddecodierungsvorrichtungen der ersten bis sechsten Ausführungsform der vorliegenden Erfindung werden

durch Aufzeichnung eines Bildcodierungsprogramms in Aufzeichnungsmedien, das die Bildcodierungsverfahren der einzelnen Ausführungsformen implementiert, und eines Bilddecodierungsprogramms, das die Bilddecodierungsverfahren der einzelnen Ausführungsformen implementiert, und Abarbeiten der aufgezeichneten Programme auf einem Computersystem mit einem Personalcomputer und einer Workstation realisiert. Als Aufzeichnungsmedien, die solche Programme aufzeichnen können, können beispielsweise Disketten oder optische Platten wie CD-ROMs verwendet werden.

Anwendungsmöglichkeiten in der Industrie

[0183] Wie vorstehend beschrieben, kann mit den erfindungsgemäßen Bildcodierungsverfahren und Bildcodierungsvorrichtungen eine Codierung mit einer hohen Komprimierungsrate an einer breiten Palette von Objekten mit verschiedenen Komprimierungsraten unter Verwendung einer einzigen Tabelle von Codes mit variabler Wortlänge durchgeführt werden, wenn die zu verarbeitenden Objekte die digitalen Bilddaten sind, die durch DCT- oder Wellenformcodierung kompressionscodiert sind.

[0184] Mit dem Bilddecodierungsverfahren und den Bilddecodierungsvorrichtungen der vorliegenden Erfindung kann ein Codierungsergebnis, das der vorstehend beschriebenen effizienten Codierung unterzogen worden ist, entsprechend decodiert werden, um ein wiederhergestelltes Bild zu erhalten.

[0185] Außerdem können ein Aufzeichnungsmedium, das ein Bildcodierungsprogramm enthält, und ein Aufzeichnungsmedium, das ein erfindungsgemäßes Bilddecodierungsprogramm enthält, eine Bildcodierungsvorrichtung, die das Codierungsergebnis mit einer hohen Komprimierungsrate erhalten kann, und eine Bilddecodierungsvorrichtung, die das Codierungsergebnis unter voller Nutzung der Ressourcen der Vorrichtung entsprechend verarbeiten kann, dadurch implementieren, dass die aufgezeichneten Programme auf einem Computersystem abgearbeitet werden.

Patentansprüche

1. Bilddecodierungsverfahren zum Gewinnen eines Codes mit variabler Wortlänge aus kompressionscodierten Daten, zum Erhalten eines dem Code mit variabler Wortlänge entsprechenden Ereignisses unter Verwendung einer Tabelle von Codes mit variabler Wortlänge, wobei jedem Bezugscode mit variabler Wortlänge ein aus [Last, Level, Run) bestehendes Bezugseignis zugeordnet wird, und zum Ableiten eines Ausgangstransformationskoeffizienten aus dem Ereignis, wobei der Begriff „Run“ die Anzahl von vorhergehenden Null-Transformationskoeffizienten, deren Wert null ist, bezeichnet, der Begriff „Level“ ei-

nen Wert eines Nicht-Null-Transformationskoeffizienten, dessen Wert nicht null ist und der auf die Null-Transformationskoeffizienten folgt, bezeichnet, und der Begriff „Last“ angibt, ob ein durch Level dargestellter Transformationskoeffizient der letzte Nicht-Null-Koeffizient in einem Block ist oder nicht, mit den Schritten:

Entscheiden, ob der Code mit variabler Wortlänge einen Steuercode enthält (S903);

Erhalten eines dem Code mit variabler Wortlänge entsprechenden Ereignisses durch Verwendung der Tabelle von Codes mit variabler Wortlänge, wenn entschieden wird, dass der Code mit variabler Wortlänge den Steuercode nicht enthält (S904);

Erhalten eines dem Code mit variabler Wortlänge entsprechenden Ereignisses durch Verwendung der Tabelle von Codes mit variabler Wortlänge und anschließend Erhalten eines transformierten Ereignisses durch Transformieren des Level-Werts des dem Code mit variabler Wortlänge entsprechenden Ereignisses unter Verwendung einer vorgegebenen Funktion, wenn entschieden wird, dass der Code mit variabler Wortlänge den Steuercode und einen zweiten Moduscode enthält (S906);

Erhalten eines dem Code mit variabler Wortlänge entsprechenden Ereignisses durch Verwendung der Tabelle von Codes mit variabler Wortlänge und anschließend Erhalten eines transformierten Ereignisses durch Transformieren des Run-Werts des dem Code mit variabler Wortlänge entsprechenden Ereignisses unter Verwendung einer vorgegebenen Funktion, wenn entschieden wird, dass der Code mit variabler Wortlänge den Steuercode und einen dritten Moduscode enthält (S908); und

Erhalten eines Ereignisses dadurch, dass der Code mit variabler Wortlänge einer Decodierung mit fester Wortlänge unterzogen wird, wenn entschieden wird, dass der Code mit variabler Wortlänge den Steuercode und einen ersten Moduscode enthält (S909).

transformierter Run-Wert = resultierender Run-Wert + (ein vorgegebener Relativzeigerwert + 1).

Es folgen 22 Blatt Zeichnungen

2. Bilddecodierungsverfahren nach Anspruch 1, dadurch gekennzeichnet, dass
der Steuercode „0000011“ ist,
der erste Moduscode „11“ ist,
der zweite Moduscode „0“ ist und
der dritte Moduscode „10“ ist.

3. Bilddecodierungsverfahren nach Anspruch 1, dadurch gekennzeichnet, dass die vorgegebene Funktion zum Transformieren des Level-Werts wie folgt definiert ist:

transformierter Level-Wert = Vorzeichen(resultierender Level-Wert) × [abs.(resultierender Level-Wert) + ein vorgegebener Relativzeigerwert].

4. Bilddecodierungsverfahren nach Anspruch 1, dadurch gekennzeichnet, dass die vorgegebene Funktion zum Transformieren des Run-Werts wie folgt definiert ist:

Anhängende Zeichnungen

Fig.1

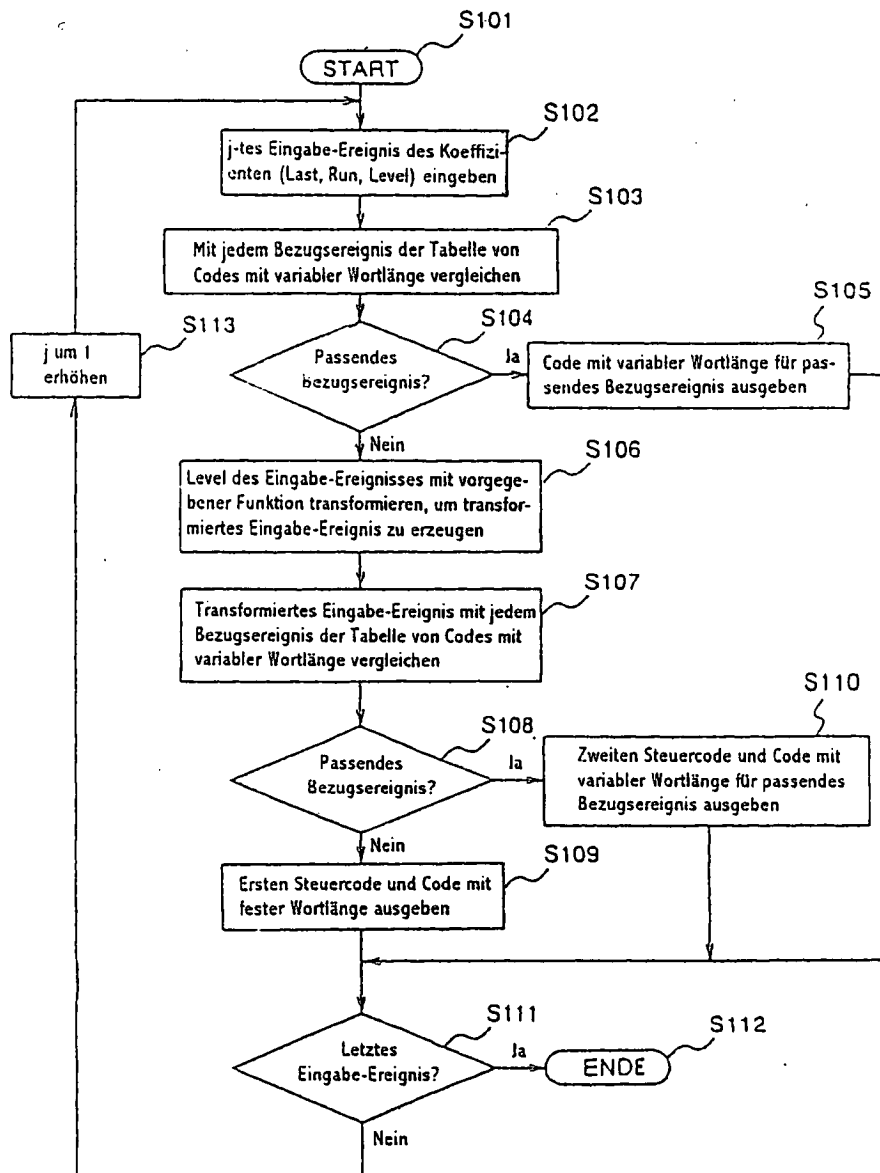


Fig.2

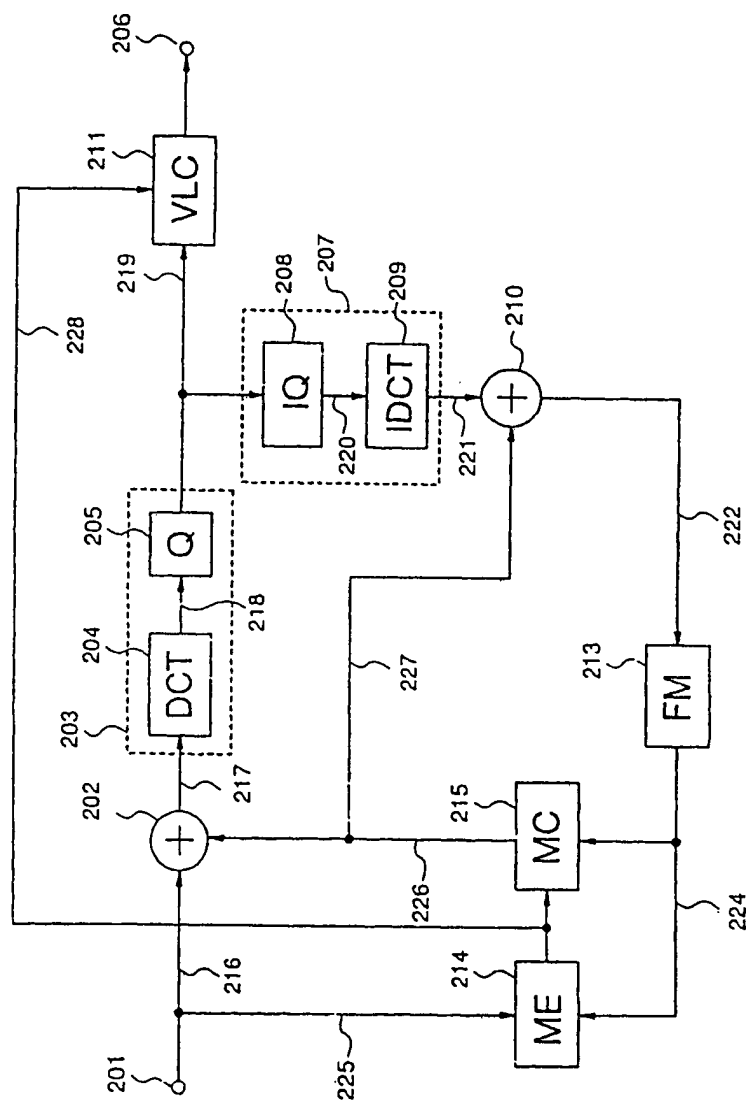


Fig.3 (a)

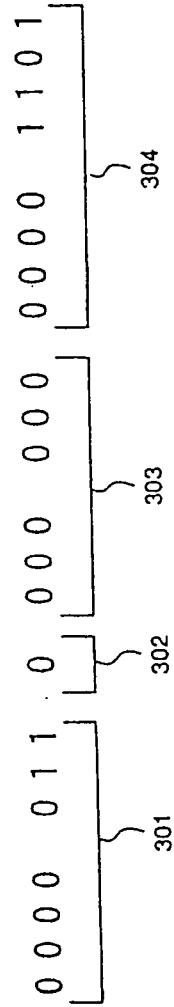


Fig.3 (b)

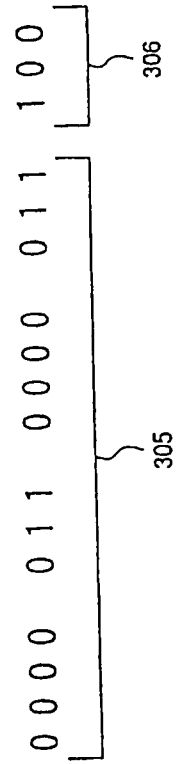


Fig.4

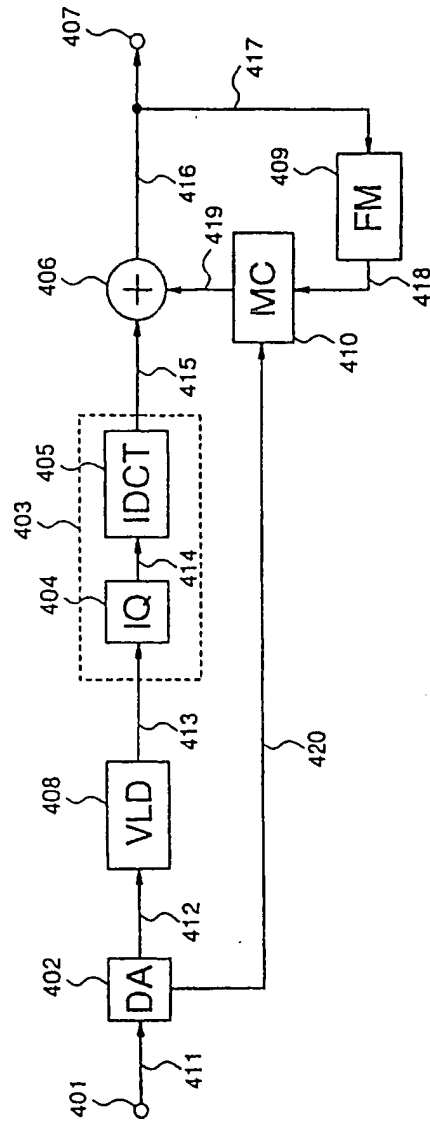


Fig.5

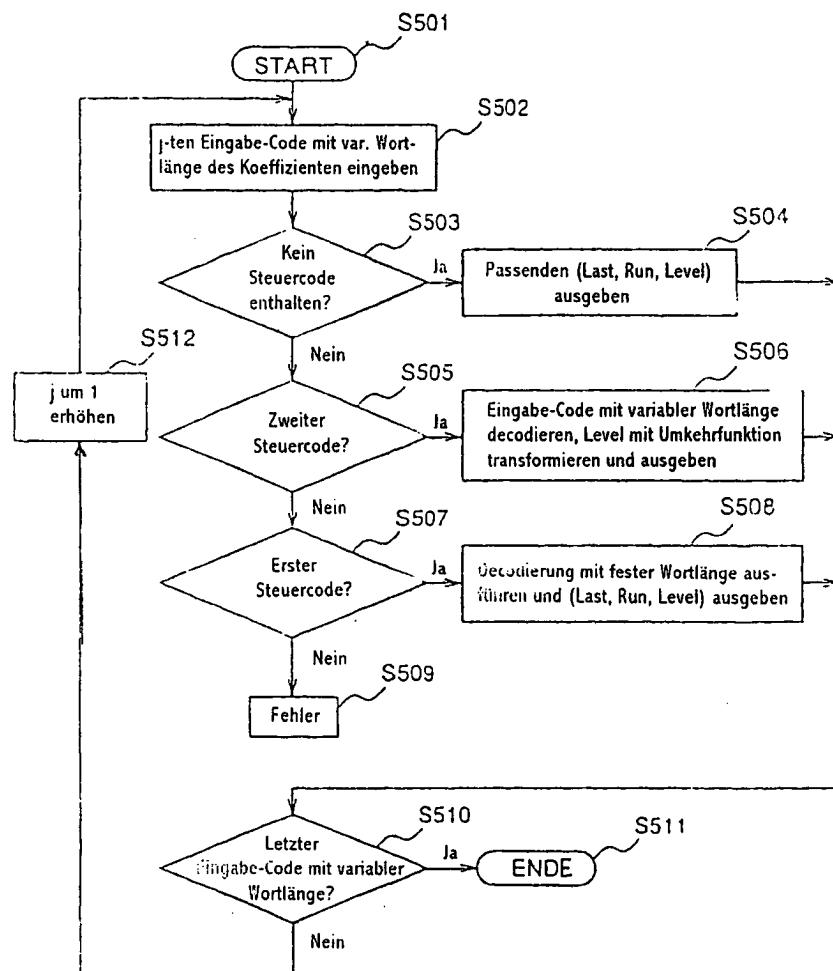


Fig.6

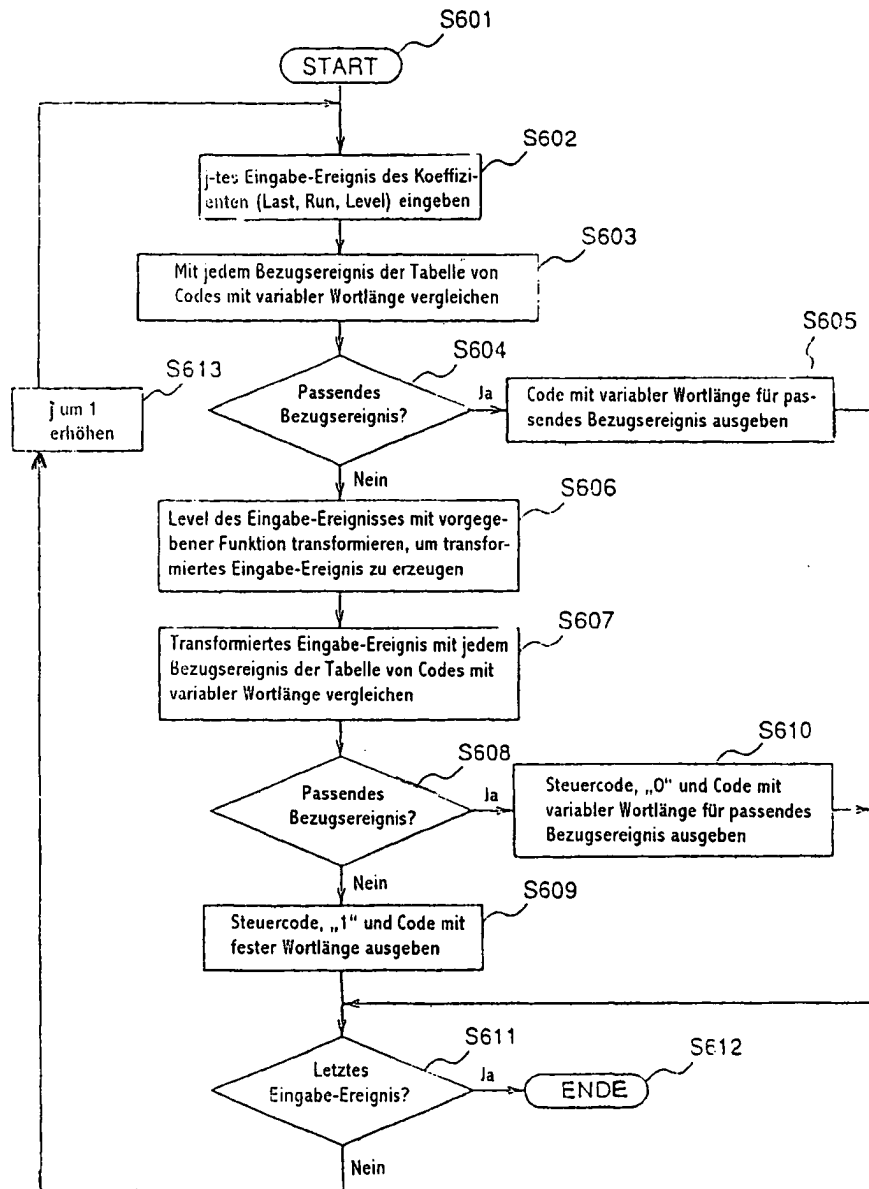
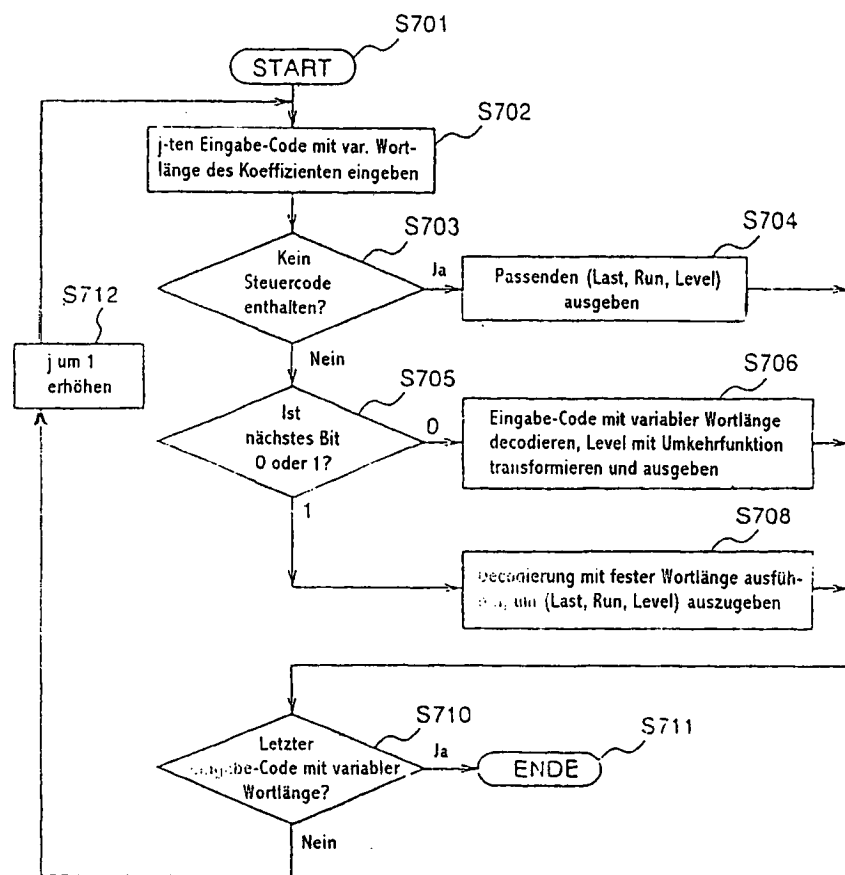


Fig.7



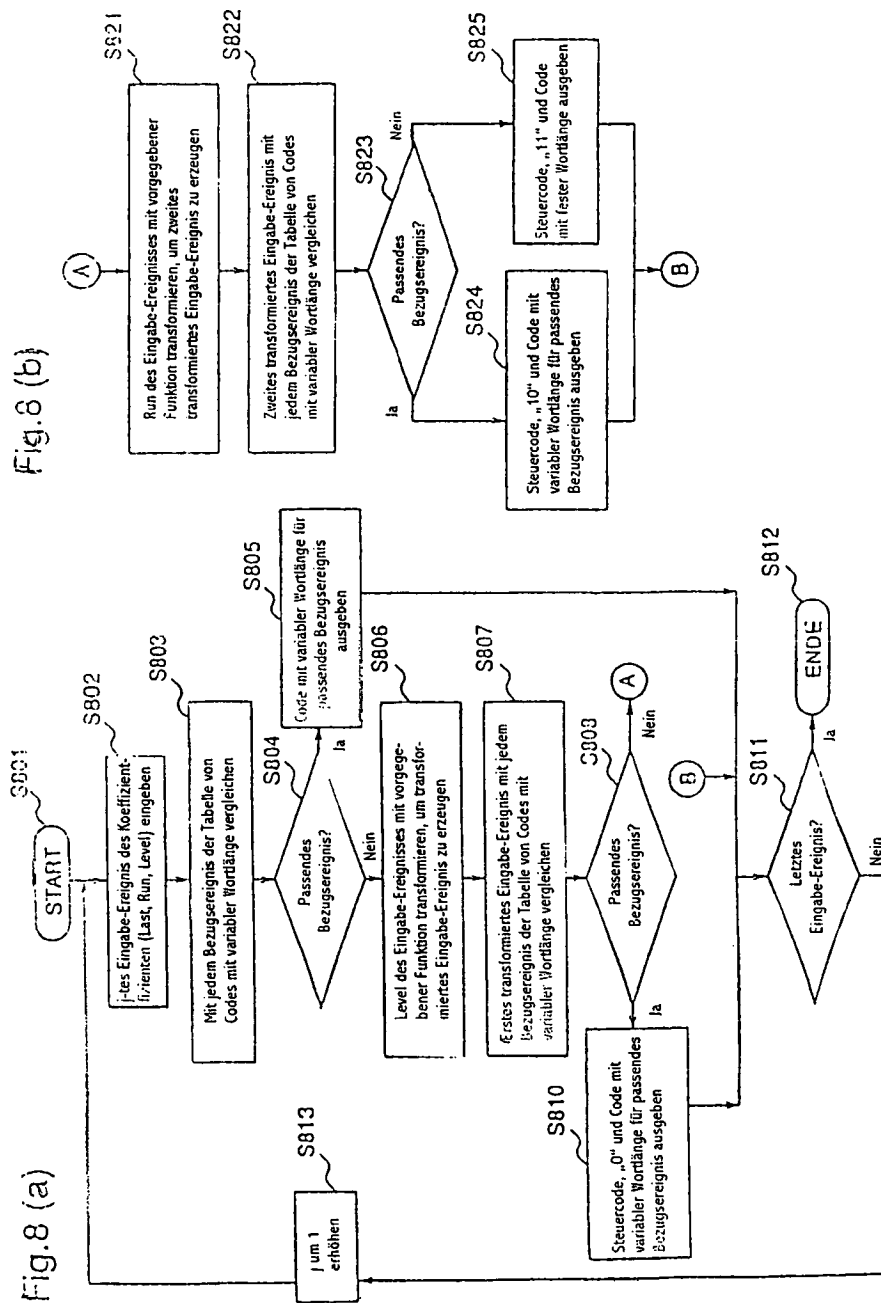


Fig.9

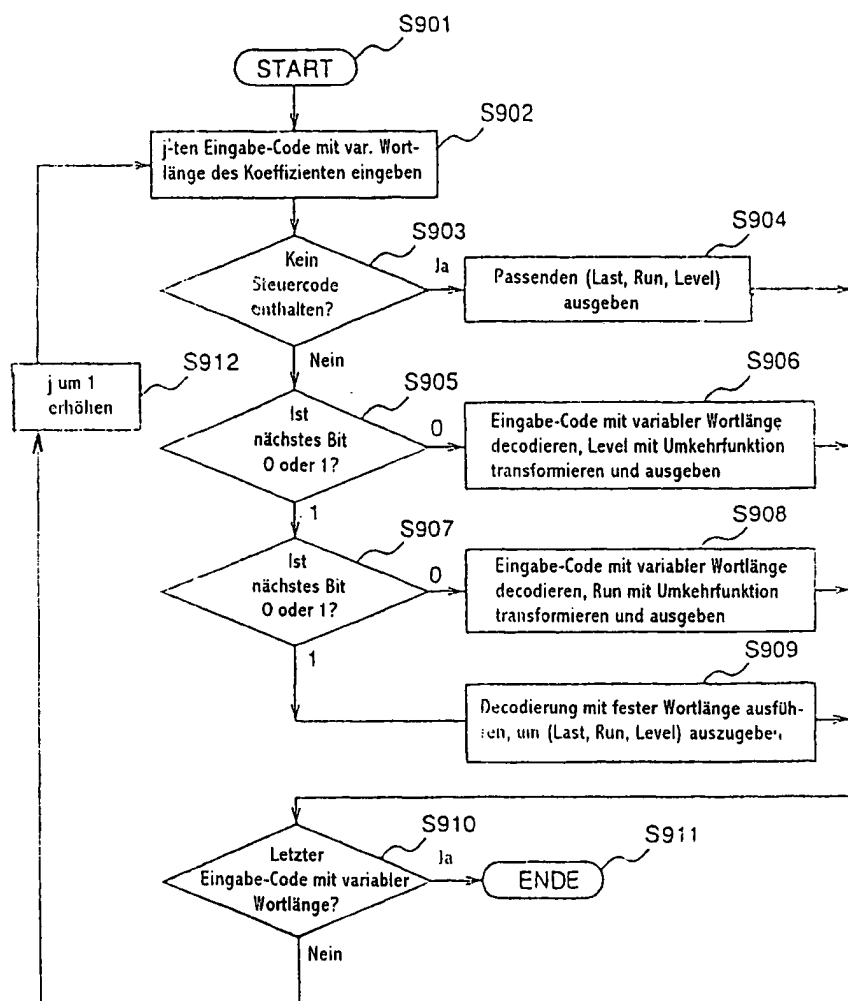


Fig.10

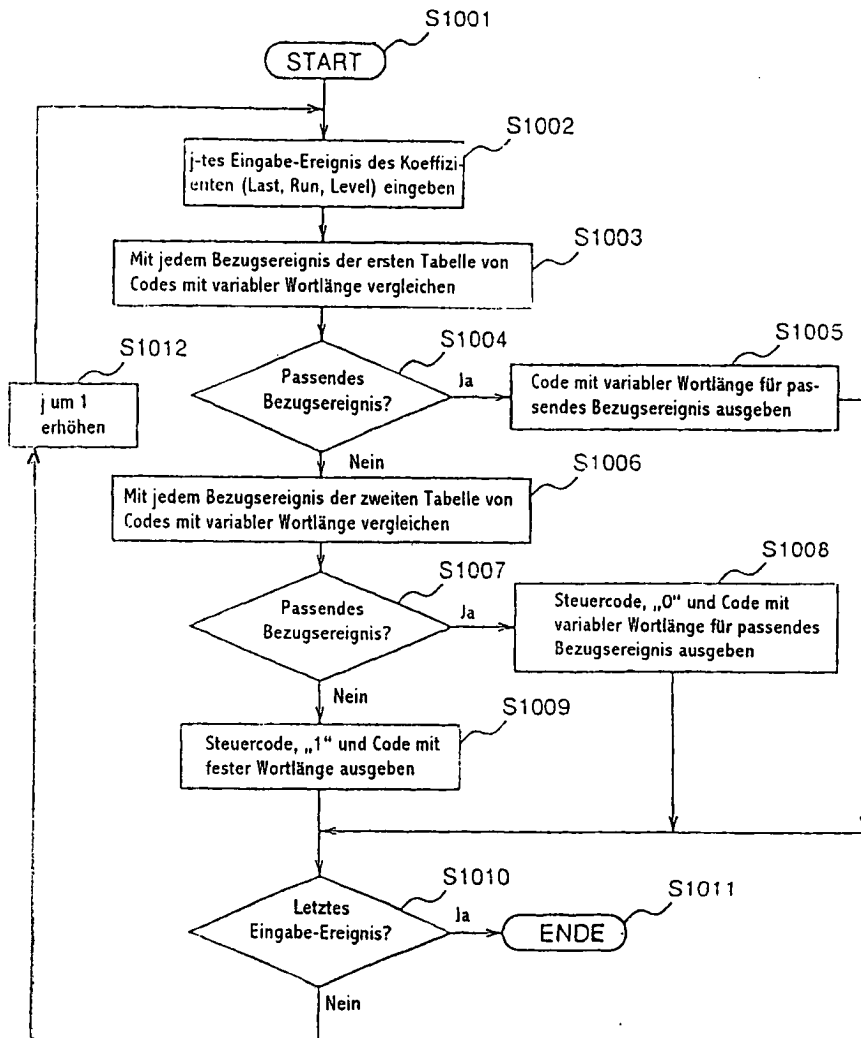


Fig.11

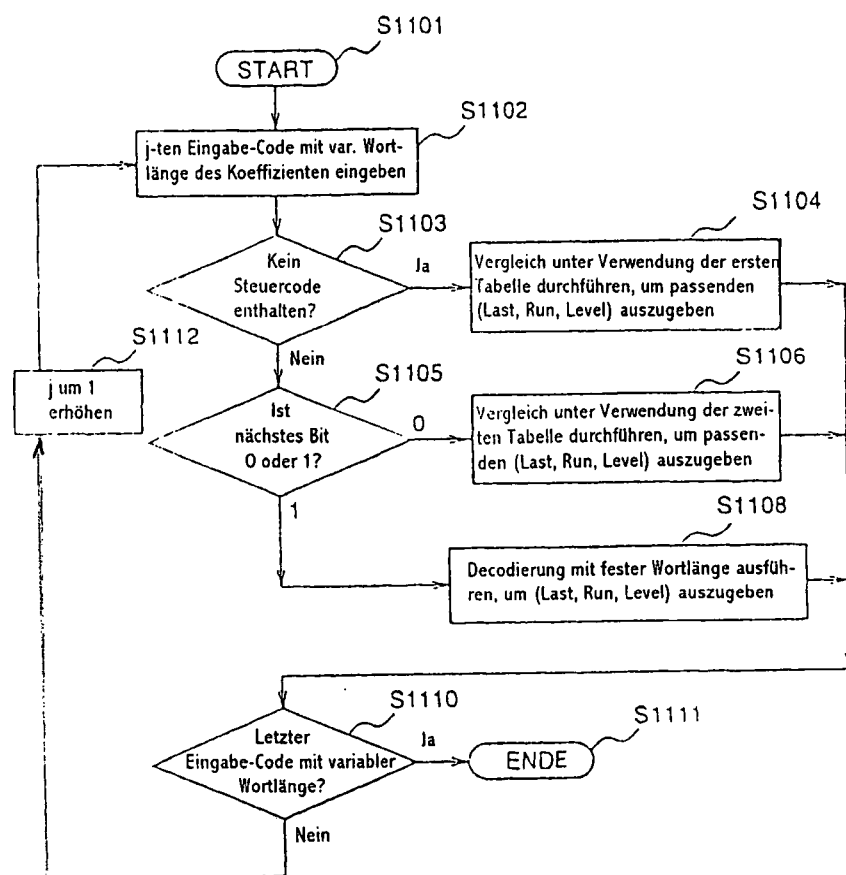


Fig.12

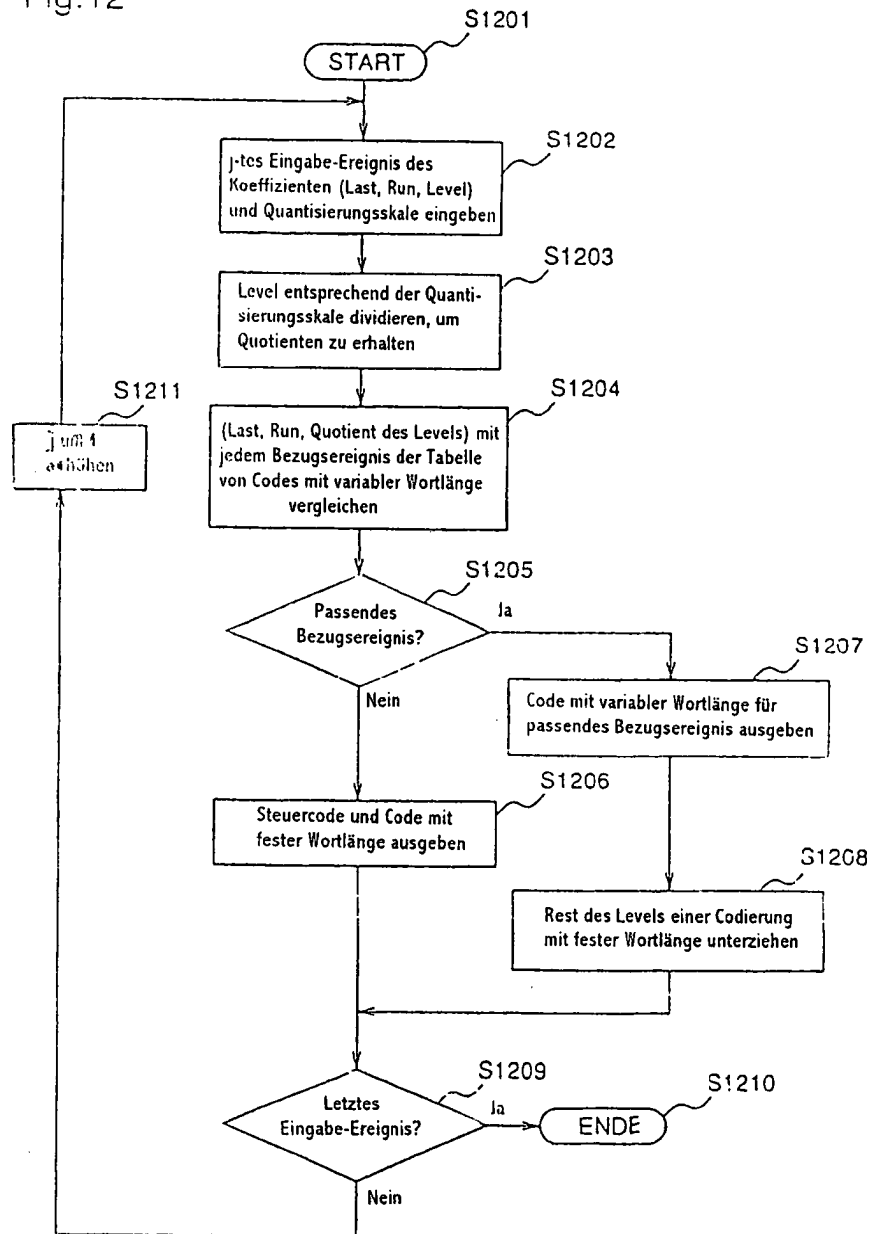


Fig. 13(a)

QUANTISIERUNGS- SKALE	DIVISOR	BIT-ANZAHL DES RESTS
1 ~ 7	2	1
≥ 8	1	0

Fig. 13(b)

QUANTISIERUNGS- SKALE	DIVISOR	BIT-ANZAHL DES RESTS
1 ~ 3	4	2
4 ~ 11	2	1
≥ 12	1	0

Fig.14

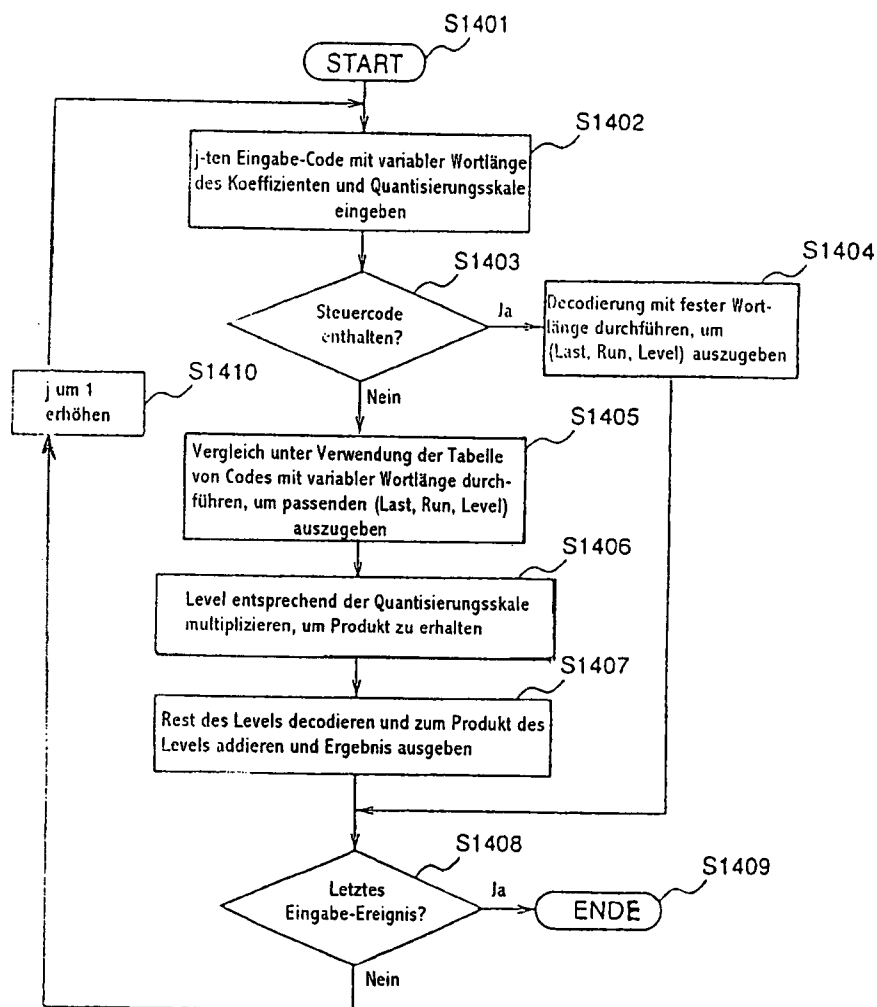


Fig.15

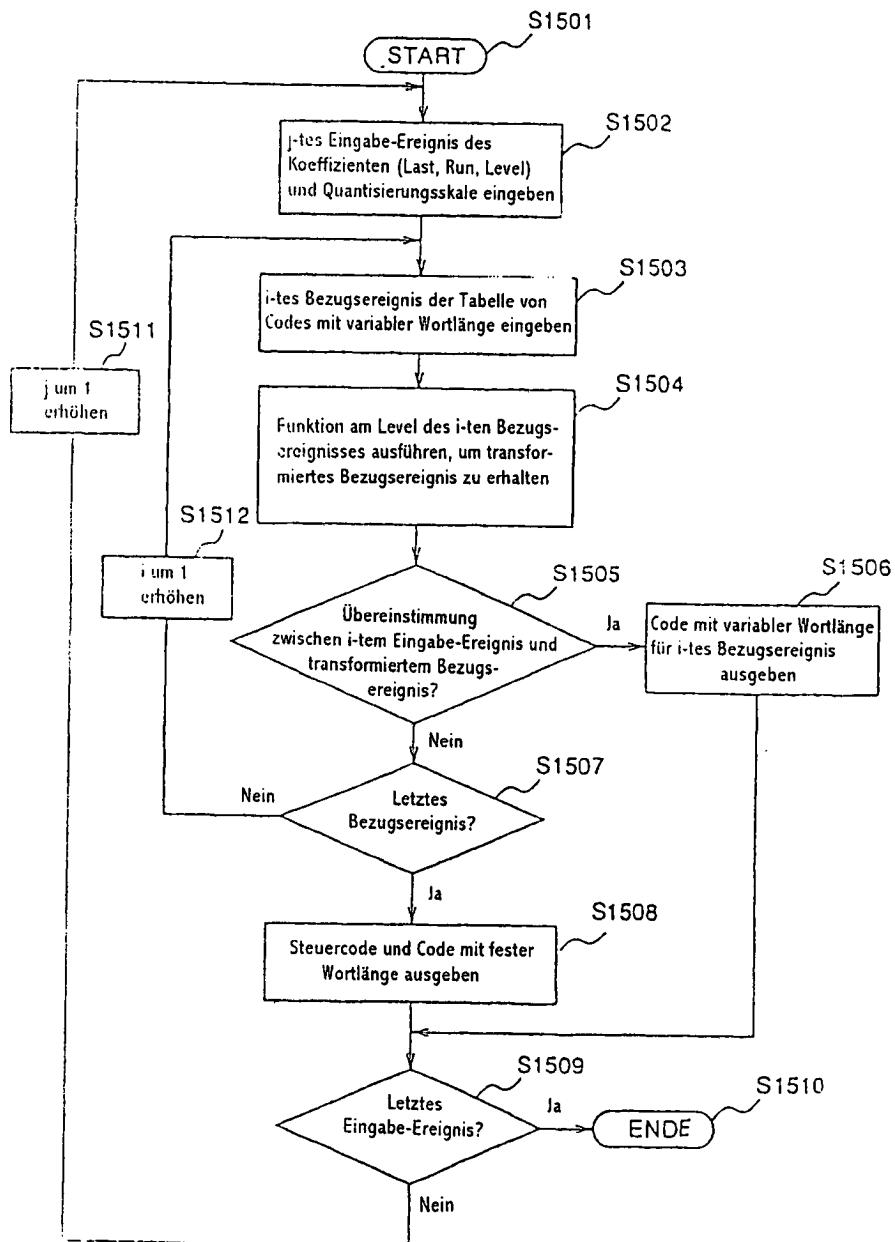


Fig.16

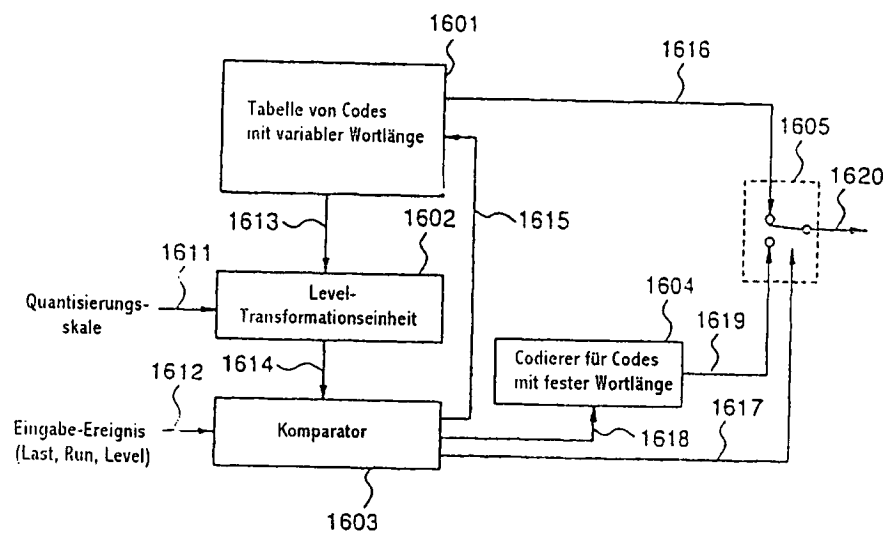


Fig.17

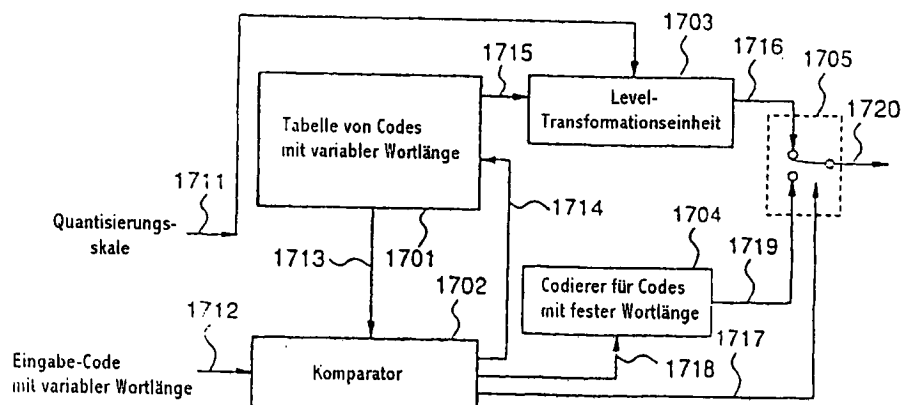


Fig.18

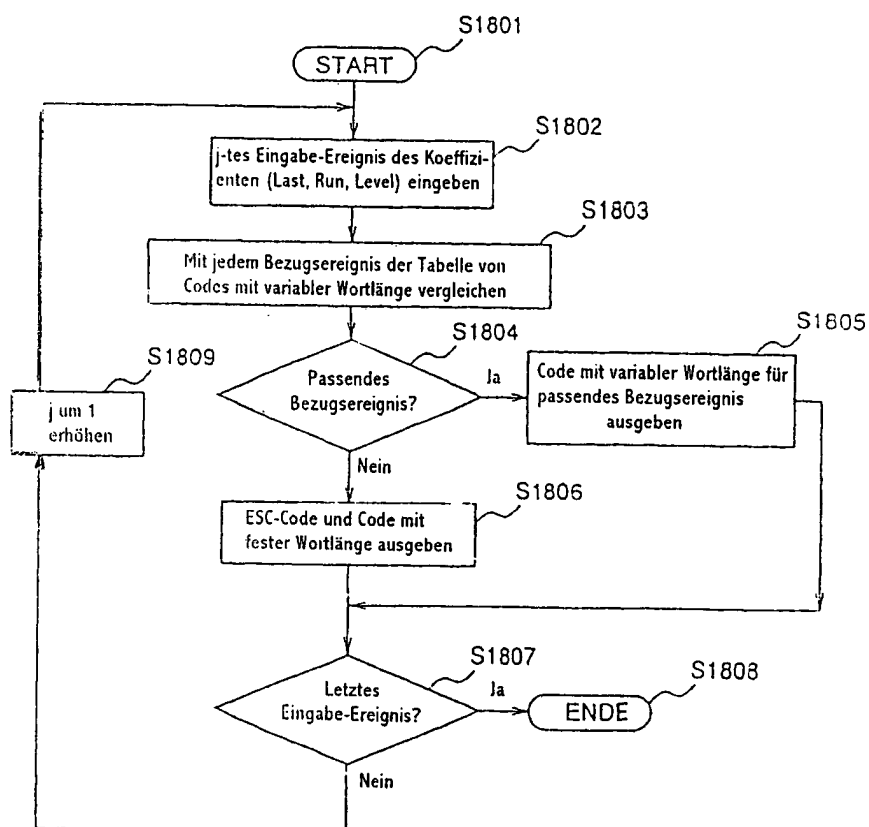


Fig.19

INDEX	LAST	RUN	LEVEL	BITS	VLC-CODE
0	0	0	1	3	10s
1	0	0	2	5	1111s
2	0	0	3	7	0101 01s
3	0	0	4	8	0010 111s
4	0	0	5	9	0001 1111s
5	0	0	6	10	0001 0010 1s
6	0	0	7	10	0001 0010 0s
7	0	0	8	11	0000 1000 01s
8	0	0	9	11	0000 1000 00s
9	0	0	10	12	0000 0000 111s
10	0	0	11	12	0000 0000 110s
11	0	0	12	12	0000 0100 000s
12	0	1	1	4	110s
13	0	1	2	7	0101 00s
14	0	1	3	8	0001 1110s
15	0	1	4	11	0000 0011 11s
16	0	1	5	12	0000 0100 001s
17	0	1	6	13	0000 0101 0000s
18	0	2	1	5	1110s
19	0	2	2	9	0001 1101s
20	0	2	3	11	0000 0011 10s
21	0	2	4	13	0000 0101 0001s
22	0	3	1	6	0110 1s
23	0	3	2	10	0001 0001 1s
24	0	3	3	11	0000 0011 01s
25	0	4	1	6	0110 0s
26	0	4	2	10	0001 0001 0s
27	0	4	3	13	0000 0101 0010s
28	0	5	1	6	0101 1s
29	0	5	2	11	0000 0011 00s
30	0	5	3	13	0000 0101 0011s

Fig.20

31	0	6	1	7	0100 11s
32	0	6	2	11	0000 0010 11s
33	0	6	3	13	0000 0101 0100s
34	0	7	1	7	0100 10s
35	0	7	2	11	0000 0010 10s
36	0	8	1	7	0100 01s
37	0	8	2	11	0000 0010 01s
38	0	9	1	7	0100 00s
39	0	9	2	11	0000 0010 00s
40	0	10	1	8	0010 110s
41	0	10	2	13	0000 0101 0101s
42	0	11	1	8	0010 101s
43	0	12	1	8	0010 100s
44	0	13	1	9	0001 1100s
45	0	14	1	9	0001 1011s
46	0	15	1	10	0001 0000 1s
47	0	16	1	10	0001 0000 0s
48	0	17	1	10	0000 1111 1s
49	0	18	1	10	000 1111 0s
50	0	19	1	10	0000 1110 1s
51	0	20	1	10	0000 1110 0s
52	0	21	1	10	0000 1101 1s
53	0	22	1	10	0000 1101 0s
54	0	23	1	12	0000 0100 010s
55	0	24	1	12	0000 0100 011s
56	0	25	1	13	0000 0101 0110s
57	0	26	1	13	0000 0101 0111s

Fig.21

INDEX	LAST	RUN	LEVEL	BITS	VLC-CODE
58	1	0	1	5	011s
59	1	0	2	10	0000 1100 1s
60	1	0	3	12	0000 0000 101s
61	1	1	1	7	0011 11s
62	1	1	2	12	0000 0000 100s
63	1	2	1	7	0011 10s
64	1	3	1	7	0011 01s
65	1	4	1	7	0011 00s
66	1	5	1	8	0010 011s
67	1	6	1	8	0010 010s
68	1	7	1	8	0010 001s
69	1	8	1	8	0010 000s
70	1	9	1	9	0001 1010s
71	1	10	1	9	0001 1001s
72	1	11	1	9	0001 1000s
73	1	12	1	9	0001 0111s
74	1	13	1	9	0001 0110s
75	1	14	1	9	0001 0101s
76	1	15	1	9	0001 0100s
77	1	16	1	9	0001 0011s
78	1	17	1	10	0000 1100 0s
79	1	18	1	10	0000 1011 1s
80	1	19	1	10	0000 1011 0s
81	1	20	1	10	0000 1010 1s
82	1	21	1	10	0000 1010 0s
83	1	22	1	10	0000 1001 1s
84	1	23	1	10	0000 1001 0s
85	1	24	1	10	0000 1000 1s
86	1	25	1	11	0000 0001 11s
87	1	26	1	11	0000 0001 10s
88	1	27	1	11	0000 0001 01s

Fig.22

89	1	28	1	11	0000 0001 00s
90	1	29	1	12	0000 0100 100s
91	1	30	1	12	0000 0100 101s
92	1	31	1	12	0000 0100 110s
93	1	32	1	12	0000 0100 111s
94	1	33	1	13	0000 0101 1000s
95	1	34	1	13	0000 0101 1001s
96	1	35	1	13	0000 0101 1010s
97	1	36	1	13	0000 0101 1011s
98	1	37	1	13	0000 0101 1100s
99	1	38	1	13	0000 0101 1101s
100	1	39	1	13	0000 0101 1110s
101	1	40	1	13	0000 0101 1111s
102	ESC			7	0000 011

Fig.23 (a)

INDEX	RUN	CODE
0	0	000 000
1	1	000 001
2	2	000 010
.	.	.
.	.	.
63	63	111 111

Fig.23 (b)

INDEX	LAST	CODE
-	-128	UNZULÄSSIG
0	-127	1000 0001
.	.	.
.	.	.
125	-2	1111 1110
126	-1	1111 1111
-	0	UNZULÄSSIG
127	1	0000 0001
128	2	0000 0010
.	.	.
.	.	.
253	127	0111 1111