

(12) NACH DEM VERTRAG ÜBER DIE INTERNATIONALE ZUSAMMENARBEIT AUF DEM GEBIET DES PATENTWESENS (PCT) VERÖFFENTLICHTE INTERNATIONALE ANMELDUNG

(19) Weltorganisation für geistiges Eigentum  
Internationales Büro



(43) Internationales Veröffentlichungsdatum  
27. März 2003 (27.03.2003)

PCT

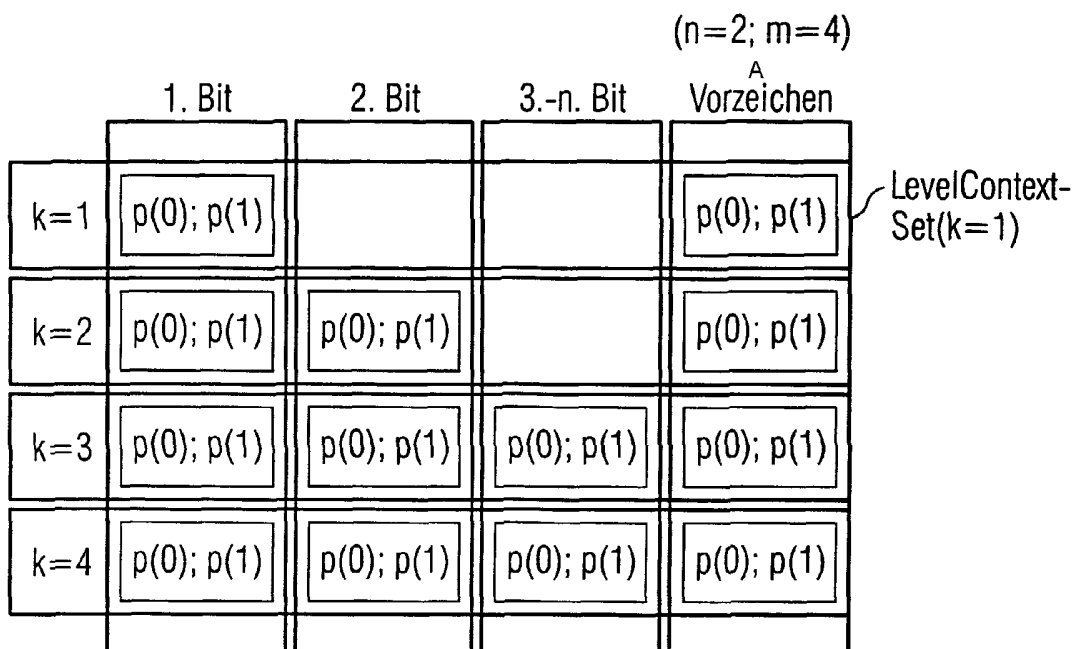
(10) Internationale Veröffentlichungsnummer  
**WO 03/026307 A2**

- (51) Internationale Patentklassifikation<sup>7</sup>: **H04N 7/26** (71) **Anmelder** (für alle Bestimmungsstaaten mit Ausnahme von US): **SIEMENS AKTIENGESELLSCHAFT** [DE/DE]; Wittelsbacherplatz 2, 80333 München (DE).
- (21) Internationales Aktenzeichen: PCT/DE02/03385
- (22) Internationales Anmeldedatum: 11. September 2002 (11.09.2002) (72) **Erfinder; und**  
(75) **Erfinder/Anmelder** (nur für US): **OERTEL, Norbert** [DE/DE]; Unterhachinger Str. 49, 81737 München (DE).  
**BÄSE, Gero** [DE/DE]; Thalkirchner Str. 184, 81371 München (DE).
- (25) Einreichungssprache: Deutsch
- (26) Veröffentlichungssprache: Deutsch (74) **Gemeinsamer Vertreter:** **SIEMENS AKTIENGESELLSCHAFT**; Postfach 22 16 34, 80506 München (DE).
- (30) Angaben zur Priorität:  
101 45 373.6 14. September 2001 (14.09.2001) DE  
102 18 541.7 25. April 2002 (25.04.2002) DE (81) **Bestimmungsstaaten** (national): AU, BR, CN, KR, US.

[Fortsetzung auf der nächsten Seite]

(54) Title: METHOD FOR PRODUCING VIDEO CODING AND PROGRAMME-PRODUCT

(54) Bezeichnung: VERFAHREN ZUR VIDEOCODIERUNG UND COMPUTERPROGRAMMPRODUKT



A... MARK

(57) **Abstract:** According to the invention, there are provided sets of contexts specifically adapted to encode spectral coefficients of a prediction error matrix, on the basis of previously encoded values of level k. Furthermore, the number of values of levels other than 0 is explicitly encoded and numbers of appropriate contexts are selected on the basis of the number of spectral coefficients other than 0.

[Fortsetzung auf der nächsten Seite]



WO 03/026307 A2



**(84) Bestimmungsstaaten (regional):** europäisches Patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, SK, TR).

— *Erfindererklärung (Regel 4.17 Ziffer iv) nur für US*

**Erklärungen gemäß Regel 4.17:**

— *hinsichtlich der Berechtigung des Anmelders, ein Patent zu beantragen und zu erhalten (Regel 4.17 Ziffer ii) für die folgenden Bestimmungsstaaten AU, BR, CN, KR, europäisches Patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, SK, TR)*

**Veröffentlicht:**

— *ohne internationalen Recherchenbericht und erneut zu veröffentlichen nach Erhalt des Berichts*

*Zur Erklärung der Zweibuchstaben-Codes und der anderen Abkürzungen wird auf die Erklärungen ("Guidance Notes on Codes and Abbreviations") am Anfang jeder regulären Ausgabe der PCT-Gazette verwiesen.*

---

**(57) Zusammenfassung:** Es wird vorgeschlagen, zur Codierung der Spektralkoeffizienten eine Prädiktionsfehlermatrix spezifisch angepasste Kontextsätze in Abhängigkeit von den zuvor codierten Pegelwerten  $k$  zu verwenden. Ferner wird vorgeschlagen, die Anzahl der von 0 verschiedenen Pegelwerte explizit zu codieren und geeignete Kontextmengen in Abhängigkeit von der Zahl der von 0 verschiedenen Spektralkoeffizienten auszuwählen.

Beschreibung

Verfahren zur Videocodierung und Computerprogrammprodukt

5 Die Erfindung betrifft ein Verfahren zur Videocodierung sowie ein Computerprogrammprodukt.

Prädiktive Videocodierverfahren schätzen aus bereits übertra-  
genen Bildbereichen die aktuelle Bildinformation und berech-  
10 nen einen davon abweichenden Prädiktionsfehler. Die Ermitt-  
lung der aktuellen Bildinformation aus dem bereits übertrage-  
nen Bildbereichen erfolgt in der Regel dadurch, dass das ak-  
tuelle Eingabebild in Blöcke unterteilt wird und zu jedem  
dieser Blöcke in vorangegangenen Bildern ein korrespondieren-  
15 der Block gesucht wird, der ein Fehler- oder Abstandsmaß mi-  
nimiert. Das so konstruierte Blockbild wird von dem aktuellen  
Eingabebild subtrahiert und der so berechnete Prädiktionsfeh-  
ler wird durch eine diskrete Cosinustransformation oder eine  
Integertransformation vom Orts- in den Frequenzbereich trans-  
20 formiert. Anschließend werden die transformierten Prädikti-  
onsfehlerdaten quantisiert und die sich daraus ergebenden Ko-  
effizienten mittels kontextadaptiver binärer arithmetischer  
Codierung (CABAC = Context Adaptive Binary Arithmetic Coder)  
komprimiert und zum Empfänger gesendet.

25

Für die arithmetische Codierung werden die in der Prädikti-  
onsfehlermatrix enthaltenen Koeffizienten durch eine Koeffi-  
zientenabtafung linearisiert und in einer Folge von Pegeln  
und Längen von Nullfolgen umgewandelt. Sowohl die Pegelwerte  
30 als auch die Längenwerte werden dabei unär dargestellt und  
unabhängig voneinander Bit für Bit codiert. Die Vorzeichen  
der Pegel werden getrennt codiert. Zur Codierung der Pegel-  
werte wird jeweils ein spezifischer Kontext für die ersten  
beiden Bits benutzt und ein weiterer spezifischer Kontext für  
35 alle folgenden Bits. Unter Kontext wird in diesem Zusammen-  
hang die Verteilung der Häufigkeiten der logischen 0 und 1  
verstanden. Der Kontext gibt daher die Wahrscheinlichkeit da-

für an, dass ein Bit gesetzt oder nicht gesetzt ist. Zur Codierung der Längenwerte wird ein spezifischer Kontext für das erste Bit und ein weiterer spezifischer Kontext für alle folgenden Bits benutzt. Auch für die Codierung des Vorzeichens, welches durch ein einzelnes Bit dargestellt wird, wird ein eigener Kontext verwendet. Die sechs verwendeten spezifischen Kontexte bilden zusammen eine Kontextmenge.

Ausgehend von diesem Stand der Technik liegt der Erfindung die Aufgabe zugrunde, die kontextadaptive binäre arithmetische Codierung weiter zu verbessern.

Diese Aufgabe wird erfindungsgemäß durch ein Verfahren mit den folgenden Verfahrensschritten gelöst:

- Bereitstellen einer Prädiktionsfehlermatrix;
- Umwandeln der Prädiktionsfehlermatrix durch eine Koeffizientenabtastung in eine Folge von Symbolen;
- Durchführen einer kontextadaptiven arithmetischen Codierung der Symbole auf der Grundlage von Symbolhäufigkeiten, deren Verteilung in Abhängigkeit von einem bereits codierten Symbol ausgewählt wird.

Der Erfindung liegt die Erkenntnis zugrunde, dass zwischen aufeinanderfolgenden Symbolen statistische Abhängigkeiten bestehen, da große Koeffizientenwerte überwiegend zu Beginn der Koeffizientenabtastung auftreten. Die statistischen Abhängigkeiten zwischen den Symbolen können ausgenutzt werden, indem je nach den zuvor übertragenen Symbolen spezifische Verteilungen der Symbolhäufigkeiten der kontextadaptiven arithmetischen Codierung zugrundegelegt werden. Im Gegensatz zum Stand der Technik erfolgt die Auswahl der zur Codierung verwendeten Verteilung der Symbolhäufigkeit nicht allein in Abhängigkeit von der Stellung des zu codierenden Symbols innerhalb der Symbolfolge, sondern auch in Abhängigkeit von einem jeweils zuvor konkret übertragenen Symbol.

Bei einer bevorzugten Ausführungsform wird die Prädiktions-  
fehlermatrix durch die Koeffizientenabtastung in einer Folge  
von Pegeln und Längen von Nullfolgen umgewandelt und die Pe-  
gelwerte in Abhängigkeit vom Wert eines zuvor übertragenen  
5 Pegels und die Längenwerte in Abhängigkeit vom Wert einer zu-  
vor codierten Länge codiert.

Es sei angemerkt, dass in diesem Zusammenhang unter Pegelwert  
der Betrag eines Pegels verstanden werden soll. Da die Längen  
10 keine negativen Werte annehmen können, werden im folgenden  
die Längenwerte gelegentlich auch kurz als Längen bezeichnet.

Da die statistischen Abhängigkeiten jeweils zwischen den Pe-  
gelwerten und Längenwerten untereinander besonders ausgeprägt  
15 sind, ist bei dieser Vorgehensweise eine besonders effiziente  
Codierung möglich.

Bei einer weiteren bevorzugten Ausführungsform werden die Pe-  
gel nach Größe sortiert und so die statistischen Abhängigkei-  
20 ten zwischen den Pegelwerten verstärkt.

Schließlich ist vorgesehen, die Anzahl der Koeffizienten zu  
bestimmen und zu codieren. Diese Vorgehensweise erlaubt, die  
Auswahl der Häufigkeitsverteilung für die Codierung der Sym-  
25 bole auch in Abhängigkeit von der Anzahl der Koeffizienten zu  
wählen.

Weitere Einzelheiten der Erfindung sind Gegenstand der abhän-  
30 gigen Ansprüche.

Nachfolgend werden Ausführungsbeispiele der Erfindung anhand  
der beigefügten Zeichnung im Einzelnen erläutert. Es zeigen:

Figur 1 ein Blockdiagramm eines Encoders;

- Figur 2 ein Blockdiagramm, das die Codierung von Koeffizienten einer Prädiktionsfehlermatrix veranschaulicht;
- 5 Figur 3 ein Diagramm, das den Abtastvorgang der Prädiktionsfehlermatrix zeigt;
- Figur 4 eine globale Häufigkeitsverteilung der Pegelwerte;
- 10 Figur 5 eine Häufigkeitsverteilung von Pegelwerten in Abhängigkeit von einem vorhergehenden Pegelwert;
- Figur 6 eine Darstellung einer zur Codierung der Pegelwerte verwendeten Kontextmenge; und
- 15 Figur 7 eine Darstellung einer zur Codierung der Längen verwendeten Kontextmenge.

Figur 1 zeigt einen Encoder 1, der nach dem Prinzip der bewegungskompensierenden Hybrid-Codierung arbeitet. Der Encoder 1 verfügt über einen Eingang 2, über den Videodaten dem Encoder 1 zugeführt werden. Eine dem Eingang 2 nachgeordnete Bewegungsschätzeinheit 3 segmentiert ein aktuell zu codierendes Bild des Videodatenstroms in rechteckige Makroblöcke. Für jeden dieser Makroblöcke sucht die Bewegungsschätzeinheit 3 aus den bereits übertragenen Bildern passende Makroblöcke heraus und berechnet deren Bewegungsvektoren. Die Bewegungsvektoren können dann mit Hilfe einer Codiereinheit 4 codiert und über einen Multiplexer 5 in einem an einem Ausgang 6 ausgegebenen Bitstrom eingebettet werden. Die von der Bewegungsschätzeinheit 3 berechneten Bewegungsvektoren der Makroblöcke werden auch einem Bewegungskompensator 7 mitgeteilt, der ausgehend von den in einem Bildspeicher 8 abgespeicherten, bereits übertragenen Bildern das sich durch die Verschiebung der Makroblöcke der bereits übertragenen Bildern ergebende Prädiktionsbild berechnet. Dieses Prädiktionsbild wird in einem Subtrahierer 9 von dem ursprünglichen Bild subtrahiert, um

20

25

30

35

einen Vorhersagefehler zu erzeugen, der einem diskreten Cosinus-Transformator 10 mit nachfolgendem Quantisierer 11 zugeführt wird. Der Vorhersagefehler wird auch als Prädiktionsfehler oder Textur bezeichnet. Der transformierte und quantisierte Vorhersagefehler wird an eine weitere, kontext-sensitive Codiereinheit 12 weitergeleitet, die die Prädiktionsfehlermatrix mit Hilfe einer kontextadaptiven, binären arithmetischen Codierung in einen Bitstrom wandelt, der in den Multiplexer 5 eingespeist und in den am Ausgang 6 ausgegebenen Bitstrom eingebettet wird.

Durch die Verarbeitung im diskreten Cosinus-Transformator 10 werden die Makroblöcke mit beispielsweise 8 x 8 Bildpunkte in eine Prädiktionsfehlermatrix mit 8 x 8 Spektralkoeffizienten umgewandelt. Dabei enthält der erste Koeffizient die mittlere Helligkeit des Makroblocks. Der erste Koeffizient wird daher auch als Gleichanteil bezeichnet. Die restlichen Spektralkoeffizienten geben mit steigender Indexzahl höherfrequente Anteile der Helligkeitsverteilung wieder, weshalb sie als Wechselanteile bezeichnet werden.

Durch den nachfolgenden Quantisierer 11 wird die Datenrate weiter verringert. Denn bei flächigen Bildelementen ändert sich der Vorhersagefehler von Bildpunkt zu Bildpunkt nur langsam, so dass nach der Verarbeitung im Quantisierer 11 die meisten hochfrequenten Spektralkoeffizienten gleich Null sind und daher gar nicht erst übertragen werden müssen.

Der Quantisierer 11 berücksichtigt darüber hinaus auch psychovisuelle Effekte. Denn das menschliche Gehirn nimmt niederfrequente Bildkomponenten, nämlich flächenmäßig ausgedehnte Bildkomponenten, wesentlich deutlicher wahr als hochfrequente Bildkomponenten, nämlich Details. Daher werden die hochfrequenten Spektralkoeffizienten gröber quantisiert als die niederfrequenten Spektralkoeffizienten.

Um die im Bildspeicher 8 abgelegten, bereits übertragenen Bilder nachzuführen, werden die Spektralkoeffizienten einem inversen Quantisierer 13 und einem inversen diskreten Cosinus-Transformator 14 zugeführt und die so rekonstruierten Daten des Prädiktionsfehlers auf das von dem Bewegungskompensator 7 erzeugte Prädiktionsbild in einem Addierer 15 aufaddiert. Das so erzeugte Bild entspricht demjenigen Bild, das beim Decodieren entsteht. Dieses Bild wird im Bildspeicher 8 abgelegt und dient der Bewegungsschätzeinheit 3 als Basis für die Berechnung der Bewegungsvektoren der folgenden Bilder.

Die Funktion der kontextsensitiven Codiereinheit 12 wird nachfolgend anhand der Figuren 2 bis 7 beschrieben.

Die in Figur 2 dargestellte kontextsensitive Codiereinheit verfügt über eine Abtastvorrichtung 16, durch die die Spektralkoeffizienten in der transformierten Prädiktionsfehlermatrix in eine Folge von Pegeln und Längen von Nullen umgewandelt werden. Diese Art der Darstellung wird auch als Run/Level-Darstellung bezeichnet. Durch die Abtastvorrichtung 16 wird beispielsweise die Folge von Spektralkoeffizienten 20-1001000000 in die Folge (0/2) (1/-1) (2/1) (0/0) umgewandelt. Dabei gibt die Zahl vor dem Schrägstrich jeweils die Zahl der Nullen vor dem nach dem Schrägstrich angegebenen Pegel an. Die Zahlen vor dem Schrägstrich werden als Längen bezeichnet. Die im letzten Zahlenpaar hinter dem Schrägstrich angegebene Null kennzeichnet den Umstand, dass der Rest der Ziffernfolge aus lauter Nullen besteht. Das letzte Ziffern paar kann daher auch als Angabe betrachtet werden, die das Ende der Blockübertragung kennzeichnet (EOB = End of Block).

Die von der Abtastvorrichtung 16 erzeugte Folge von Pegeln und Längen wird einem Wandler 17 zugeführt, der die binäre Darstellung in eine unäre (einwertige) Darstellung umwandelt. Dabei werden die im Beispiel angeführten Zahlenpaare jeweils in der Reihenfolge Pegel, Länge der Nullfolge und Vorzeichen des Pegels codiert. Das Zahlenpaar (0/2) wird im vorliegenden



Fall also in die unäre Folge 110/0/0 und das Zahlenpaar (1/-1) in die Folge 10/10/1 umgewandelt.

Die unäre Ziffernfolge wird abschließend einem arithmetischen  
5 Codierer 18 zugeführt, der die eigentliche kontextadaptive  
arithmetische Codierung durchführt. Für die Durchführung der  
kontextadaptiven arithmetischen Codierung benötigt der arith-  
metische Codierer 18 jeweils die Häufigkeit, mit der die Ein-  
sen und Nullen in dem vom Wandler 17 gelieferten unären Da-  
10 tenstrom auftreten. Diese Wahrscheinlichkeiten und damit die  
Verteilung der Häufigkeiten der Einsen und Nullen werden dem  
arithmetischen Codierer 18 von einem Analysator 19 geliefert,  
der von der Abtastvorrichtung 16 mit der Folge von Pegeln und  
Längen beaufschlagt wird und daraus aktuelle Verteilungen der  
15 Wahrscheinlichkeit für das Auftreten von logischen Nullen und  
Einsen bestimmt.

In der hier beschriebenen Codiereinheit 12 werden auf beson-  
dere Weise statistische Abhängigkeiten zwischen den Pegeln  
20 und Längen berücksichtigt. Diese statistischen Abhängigkeiten  
beruhen auf dem Umstand, dass große Pegelwerte meist zu Be-  
ginn des Abtastvorgangs auftreten. Umgekehrt sind große Län-  
genwerte gegen Ende des Abtastvorgangs zu häufiger. Denn ge-  
mäß Figur 3 wird die transformierte Prädiktionsfehlermatrix  
25 20 mit einem Zick-Zack-Muster 21 abgetastet, durch das zu-  
nächst die niederfrequenten Spektralkoeffizienten und dann  
die höherfrequenten Spektralkoeffizienten ausgelesen werden.  
Große Pegelwerte treten daher vor allem zu Beginn des Abtast-  
vorgangs, also zu Beginn der Folge von Pegeln und Längen auf.

30

In Figur 4 ist ein Diagramm dargestellt, in dem eine Vertei-  
lungskurve 22 die Häufigkeitsverteilung  $P(k)$  in Abhängigkeit  
von den Pegelwerten  $k$  angibt. Allgemein gilt, dass die Wahr-  
scheinlichkeit für kleine Pegelwerte größer ist als für große  
35 Pegelwerte. Die Verteilungskurve 22 fällt daher ausgehend von  
einem Maximalwert bei dem Pegelwert  $k = 0$  monoton ab.

Figur 5 enthält ein weiteres Diagramm, in das eine Verteilungskurve 23 eingezeichnet ist, die die Häufigkeitsverteilung  $P(k|k = k_x)$  für die Wahrscheinlichkeit dafür angibt, dass nach einem Pegelwert  $k = k_x$  der Pegelwert  $k$  auftritt.

5 Diese Verteilungskurve 23 weist ein Maximum auf, das bei einem Wert  $k < k_x$  liegt. Das bedeutet, dass nach dem Auftreten eines Pegelwerts  $k = k_x$  kleinere Pegelwerte sehr wahrscheinlich sind. Dies entspricht der Tatsache, dass die Pegelwerte gegen Ende des Abtastvorgangs hin abnehmen.

10

Aufgrund der statistischen Abhängigkeit der Pegelwerte untereinander ist es sinnvoll, die Häufigkeiten für die Nullen und Einsen, die der kontextadaptiven arithmetischen Codierung im arithmetischen Codierer 18 zugrundeliegen, auch in Abhängigkeit vom Wert des zuvor codierten Pegels auszuwählen.

15

Die statistischen Abhängigkeiten können noch verstärkt werden, wenn die Pegelwerte der Größe nach sortiert werden. Geeignete Sortierverfahren, bei dem die Pegelwerte der Größe nach sortiert werden und bei denen die Sortierinformation separat übertragen wird, sind dem Fachmann bekannt und als solche nicht Gegenstand der Erfindung. Daneben kommen auch zukünftige Sortierverfahren für die Anwendung in diesem Zusammenhang in Frage.

25

Durch die Sortierung der Pegel wird ein in Figur 5 schraffiert eingezeichneter Kurventeil 24 gewissermaßen abgeschnitten. Dadurch wird die statistische Abhängigkeit der Pegelwerte weiter verstärkt.

30

Die statistischen Abhängigkeiten betreffen nicht nur die Pegelwerte, sondern auch die Längen. Wie bereits erwähnt, treten große Längenwerte, insbesondere gegen Ende des Abtastvorgangs hin, auf. Es ist daher auch sinnvoll, die Verteilung der Häufigkeiten für die Nullen und Einsen, die der arithmetischen Codierung in dem arithmetischen Codierer 18 zugrunde-

35

liegt, in Abhängigkeit vom Wert der zuvor codierten Längen auszuwählen.

In den Figuren 6 und 7 sind die Kontextmengen, die für die Codierung der Pegelwerte und Längenwerte verwendet werden, graphisch dargestellt.

In Figur 6 sind insbesondere die Kontextmengen dargestellt, die der Codierung der sortierten Pegelwerte zugrundeliegen. Im vorliegenden Fall wurde für die Zahl der individuell codierten Bitstellen  $n = 2$  und für die Zahl der individuell codierten Pegelwerte  $m = 4$  gewählt. Wie bereits erwähnt, wird die Codierung eines Pegelwerts  $l$  abhängig vom Betrag des zuvor codierten Pegels  $k$  durchgeführt.

Es sei angemerkt, dass bei der unären Darstellung von sortierten Pegelwerten die für die unäre Darstellung von  $l$  verwendete abschließende Null entfallen kann, wenn  $l == k$  gilt, da der maximale Wert für  $l$  gleich  $k$  ist.

Für die ersten  $\min(n, k)$  Bit wird jeweils ein eigener Kontext verwendet. Für alle möglichen folgenden Bits bis zur maximalen Länge der unären Darstellung von  $l$  wird ein Sammelkontext verwendet. In Figur 5 ist dies der Sammelkontext für das dritte und die folgenden Bits. Im Versuch hat sich 2 als günstiger Wert für  $n$  erwiesen. Das Vorzeichen wird mit einem einzelnen Bit dargestellt und es wird ein spezifischer Vorzeichenkontext zur Codierung des Vorzeichens verwendet. Die Kontextmenge zur Codierung eines Pegels mit Betrag  $l$  unter der Voraussetzung, dass der zuletzt codierte Pegel den Betrag  $k$  hat, umfasst also  $\min(k+1, \min(n,k)+2)$  Kontexte und wird im Folgenden mit  $\text{LevelContextSet}(k)$  bezeichnet.

Bei der Codierung des ersten aus der transformierten Prädiktionsfehlermatrix ausgelesenen Pegels kann nicht auf einen zuvor codierten Pegel zurückgegriffen werden, so dass der erste ausgelesene Pegelwert separat codiert werden muss. Als

Kontext für den ersten zu codierenden Pegelwert wird der Kontext des größtmöglichen Pegelwerts angenommen, der bedingt durch die Transformation und die nachfolgende Quantisierung möglich ist.

5

Im übrigen kann anstelle der verschiedenen Kontexte für ein  $k > m$  der Kontext für  $k = m$  als Sammelkontext verwendet werden. Für die Codierung eines Pegelwerts  $l$  abhängig vom vorhergehenden Pegelwert  $k$  wird daher gemäß Figur 6 folgende Kontextmenge verwendet: `LevelContextSet(min(m,k))`. Im Versuch hat sich  $m = 4$  als günstiger Wert erwiesen.

Grundsätzlich besteht allerdings auch die Möglichkeit, die Parameter  $n$  und  $m$  dynamisch zu gestalten und die dafür gewählten Werte in einen Header zu übertragen.

Die Codierung einer Länge  $r$  wird abhängig von der zuvor codierten Länge  $p$  durchgeführt. Für die erste Länge gilt:  $p = 0$ . Wie für die Pegelwerte wird auch für die ersten  $n$  Bits jeweils ein eigener Kontext verwendet. Für alle folgenden Bits wird ein gemeinsamer Kontext benutzt. Im Versuch hat sich  $n = 3$  bewährt. Die Kontextmenge zur Codierung einer Länge  $r$  unter der Voraussetzung, dass die zuletzt codierte Länge den Betrag  $p$  hat, wird im Folgenden mit `RunContextSet(p)` bezeichnet. Für die Codierung einer Länge  $r$  abhängig von  $p$  wird folgende Kontextmenge verwendet: `RunContextSet(min(m,p))`. Im Versuch erwies sich  $m = 5$  als günstiger Wert.

Grundsätzlich besteht ebenfalls die Möglichkeit, den Parameter  $m$  in diesem Zusammenhang dynamisch zu gestalten und den dafür ausgewählten Wert mit Hilfe eines geeigneten Informationselements zu übertragen.

Durch das hier beschriebene Verfahren, bei dem die zur Codierung verwendeten Kontexte in Abhängigkeit vom zuvor codierten Pegelwert oder Längenwert ausgewählt werden, kann die Datenrate um 2 bis 3 % reduziert werden.

Eine weitere Reduzierung kann erzielt werden, wenn das Ende der aus der Prädiktionsfehlermatrix ausgelesenen Folge von Pegeln und Längen nicht über den Pegelwert 0 codiert wird, sondern wenn der Folge die Anzahl der von 0 verschiedenen ausgelesenen Pegeln vorangestellt wird. Im oben angeführten Beispiel ergäbe sich dann nicht die Folge (0/2)(1/-1)(2/1)(0/0), sondern 3(0,2)(1/-1)(2/1).

10 In diesem Fall kann die Information über die Anzahl der von 0 verschiedenen Pegelwerten für eine effiziente Codierung der Pegelwerte benutzt werden. Denn transformierte Prädiktionsfehlermatrizen mit wenigen von 0 verschiedenen Spektralkoeffizienten haben in der Regel nur Spektralkoeffizienten mit sehr kleinen Pegelwerten. Es ist daher sinnvoll, in Abhängigkeit von der Anzahl der von 0 verschiedenen Spektralkoeffizienten nun zwischen verschiedenen Kontextmengen umzuschalten und damit die Effizienz des Codiervorgangs zu steigern.

20 Alternativ ist es auch möglich, in Abhängigkeit vom Betrag des ersten übertragenen Pegelwerts zwischen verschiedenen Kontextmengen umzuschalten.

25 Die Berücksichtigung der Anzahl der von 0 verschiedenen Koeffizienten bei der Codierung durch entsprechend angepasste Kontextmengen reduziert die Datenrate des am Ausgang 6 ausgegebenen Bitstroms um weitere 5 bis 6 %.

30 Die hier beschriebenen Vorrichtungen und Verfahren eignen sich zur Anwendung im Rahmen existierender Videostandards wie H.263 und MPEG-4 sowie H.26L. Die hier beschriebenen Verfahren und Vorrichtungen sind jedoch auch für die Anwendung von zukünftigen, den heutigen Standards entsprechenden Standards geeignet.

35

Das hier beschriebene Verfahren und die hier beschriebenen Vorrichtungen sind besonders bei niedrigen Datenraten effizient, da sehr viele Pegel bei niedrigen Datenraten betragsmäßig den Wert 1 aufweisen. Dies ließ sich jedoch bisher nicht effizient statistisch modellieren, da der Schluss vom zuletzt codierten Pegelwert auf den Wertebereich des aktuell zu codierenden Pegelwerts ohne ein Vorwissen über die Folge der Koeffizienten nicht möglich ist.

10 Abschließend sei darauf hingewiesen, dass die hier beschriebenen Vorrichtungen und Verfahren sowohl in Hardware- als auch in Software realisiert werden können.

## Patentansprüche

1. Verfahren zur Videocodierung mit den Verfahrensschritten:
  - 5 - Bereitstellen einer Prädiktionsfehlermatrix;
  - Umwandeln der Prädiktionsfehlermatrix durch eine Koeffizientenabtastung in eine Folge von Symbolen;
  - Durchführen einer kontextadaptiven arithmetischen Codierung der Symbole auf der Grundlage von Symbolhäufigkeiten, deren  
10 Verteilung in Abhängigkeit von einem bereits codierten Symbol ausgewählt wird.
  
2. Verfahren nach Anspruch 1,  
bei dem die Verteilung der Häufigkeitssymbole in Abhängigkeit  
15 von dem unmittelbar vorher codierten Symbol ausgewählt wird.
  
3. Verfahren nach Anspruch 1 oder 2,  
bei dem die Prädiktionsfehlermatrix durch die Koeffizienten-  
abtastung in eine Folge von Pegeln und Längen von Nullfolgen  
20 umgewandelt wird.
  
4. Verfahren nach Anspruch 3,  
bei dem die Pegel und Längen unär repräsentiert werden.
  
- 25 5. Verfahren nach Anspruch 3 oder 4,  
bei dem die Pegel jeweils kontextadaptiv arithmetisch codiert  
werden, wobei die Verteilung der zugrundegelegten Pegelwert-  
häufigkeiten in Abhängigkeit vom jeweils zuvor codierten Pe-  
gelwert ausgewählt wird.  
30
  
6. Verfahren nach einem der Ansprüche 3 bis 5,  
bei dem die Längen kontextadaptiv arithmetisch codiert wer-  
den, wobei die zugrundegelegte Verteilung der Längenwerthäu-  
figkeiten in Abhängigkeit vom zuvor codierten Längenwert aus-  
35 gewählt wird.

7. Verfahren nach einem der Ansprüche 3 bis 6,  
bei dem die Pegelwerte vor der kontextadaptiven arithmeti-  
schen Codierung nach Pegelwerten sortiert werden.
- 5 8. Verfahren nach einem der Ansprüche 1 bis 7,  
bei dem die Anzahl der bei der Koeffizientenabtastung ausge-  
lesenen Symbole codiert und übertragen wird.
9. Verfahren nach Anspruch 8,  
10 bei dem die Auswahl der Verteilung der Symbolhäufigkeiten in  
Abhängigkeit von der Anzahl der ausgelesenen Symbole durchge-  
führt wird.
- 15 10. Computerprogrammprodukt, das Programmcodes zur Ausfüh-  
rung eines Verfahrens nach einem der Ansprüche 1 bis 9 ent-  
hält.



FIG 1

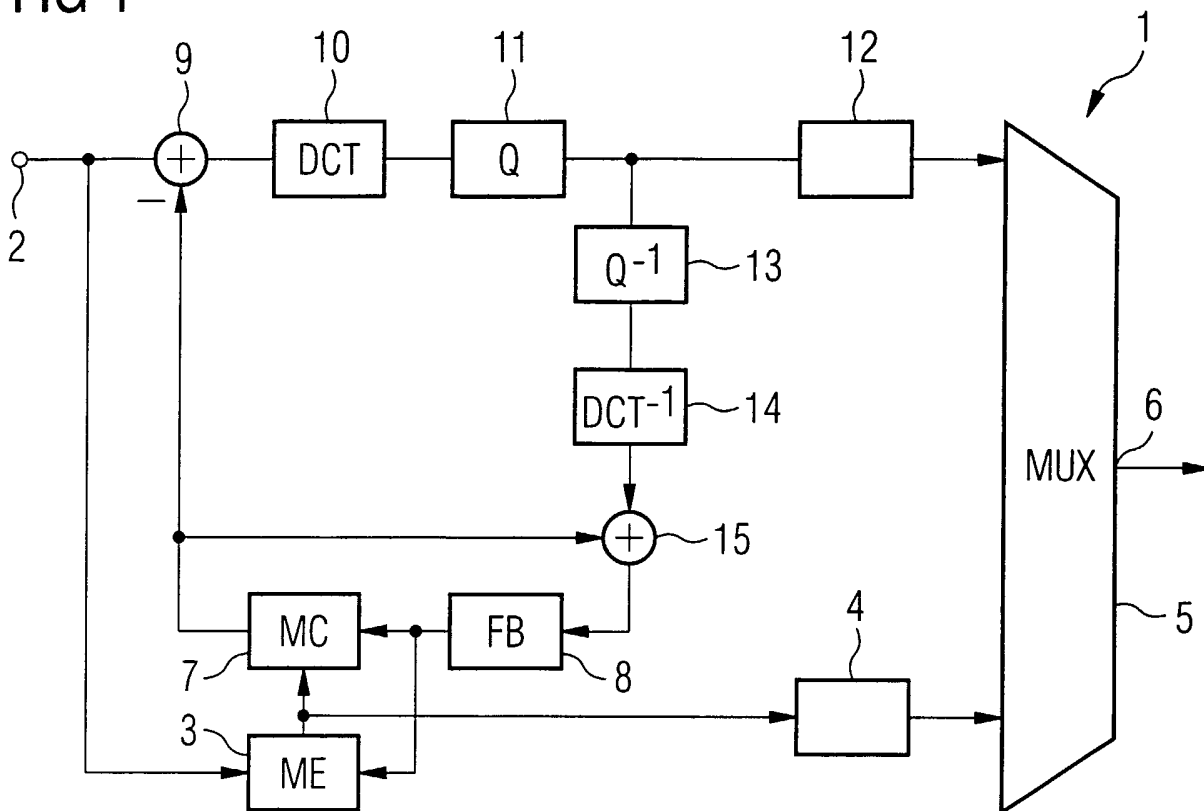


FIG 2

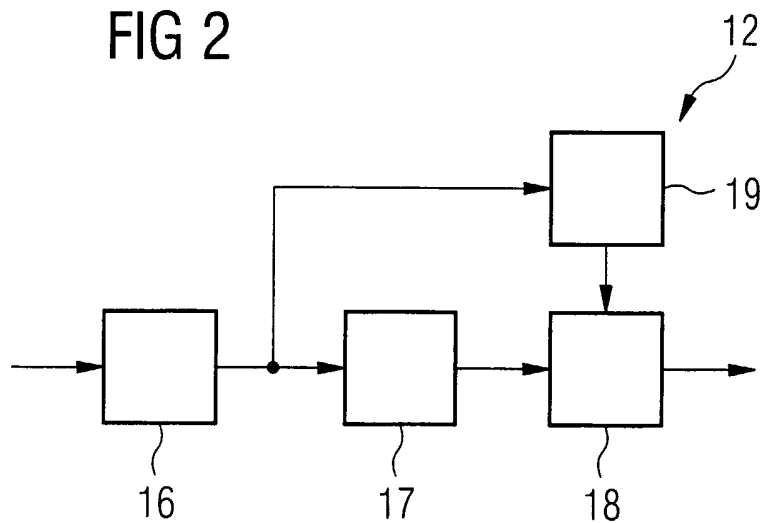


FIG 3

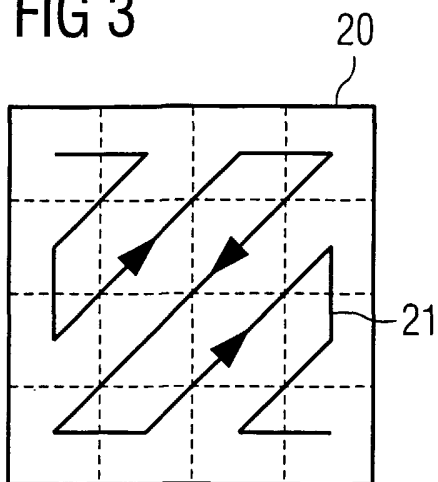


FIG 4

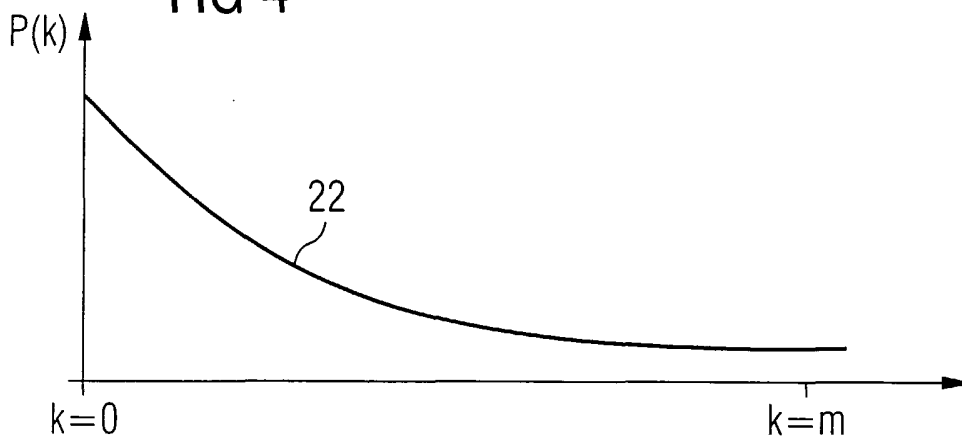


FIG 5

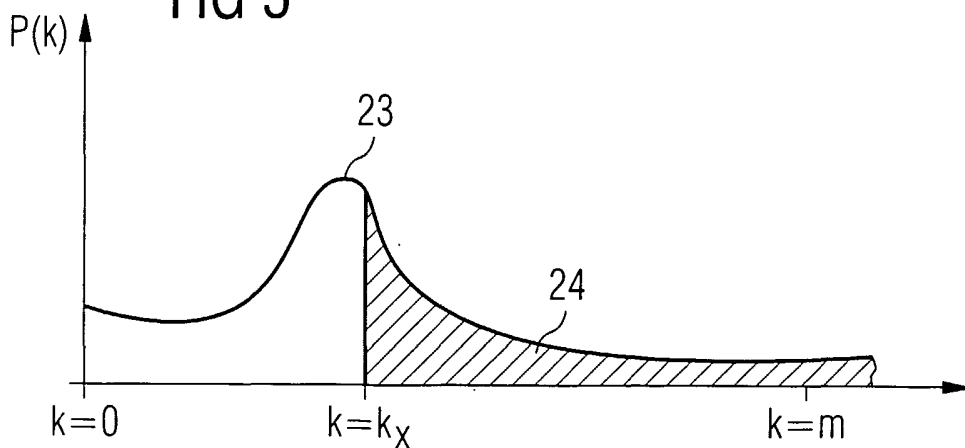


FIG 6

(n=2; m=4)

	1. Bit	2. Bit	3.-n. Bit	Vorzeichen
k=1	p(0); p(1)			p(0); p(1)
k=2	p(0); p(1)	p(0); p(1)		p(0); p(1)
k=3	p(0); p(1)	p(0); p(1)	p(0); p(1)	p(0); p(1)
k=4	p(0); p(1)	p(0); p(1)	p(0); p(1)	p(0); p(1)

LevelContext-Set(k=1)

FIG 7

(n=4; m=15)

	1. Bit	2. Bit	3. Bit	4.-n. Bit
p=0	p(0); p(1)	p(0); p(1)	p(0); p(1)	p(0); p(1)
p=1	p(0); p(1)	p(0); p(1)	p(0); p(1)	p(0); p(1)
p=2	p(0); p(1)	p(0); p(1)	p(0); p(1)	p(0); p(1)
⋮				
p=15	p(0); p(1)	p(0); p(1)	p(0); p(1)	p(0); p(1)

RunContext-Set(p=0)