



(19)
Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) **DE 603 13 205 T2** 2007.10.25

(12) **Übersetzung der europäischen Patentschrift**

(97) **EP 1 559 274 B1**

(51) Int Cl.⁸: **H04N 7/24** (2006.01)

(21) Deutsches Aktenzeichen: **603 13 205.7**

(86) PCT-Aktenzeichen: **PCT/FI03/00832**

(96) Europäisches Aktenzeichen: **03 810 480.8**

(87) PCT-Veröffentlichungs-Nr.: **WO 2004/043071**

(86) PCT-Anmeldetag: **06.11.2003**

(87) Veröffentlichungstag
der PCT-Anmeldung: **21.05.2004**

(97) Erstveröffentlichung durch das EPA: **03.08.2005**

(97) Veröffentlichungstag
der Patenterteilung beim EPA: **11.04.2007**

(47) Veröffentlichungstag im Patentblatt: **25.10.2007**

(30) Unionspriorität:

424409 P	06.11.2002	US
429953 P	29.11.2002	US
430712 P	03.12.2002	US

(84) Benannte Vertragsstaaten:

AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LI, LU, MC, NL, PT, RO, SE, SI, SK, TR

(73) Patentinhaber:

Nokia Corp., Espoo, FI

(72) Erfinder:

TIAN, Dong, FIN-33720 TAMPERE, FI; WANG, Ye-Kui, FIN-33710 Tampere, FI; HANNUKSELA, Miska, FIN-36110 RUUTANA, FI

(74) Vertreter:

COHAUSZ & FLORACK, 40211 Düsseldorf

(54) Bezeichnung: **BILDPUFFERUNG FÜR PRÄDIKTIONSREFERENZEN UND ANZEIGE**

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 eine Bilderpufferung für Vorhersagebezüge und Anzeige, wobei das Verfahren einen Codierschritt zum Ausbilden von codierten Bildern in einem Codierer, einen optionalen hypothetischen Decodierschritt zum Decodieren der codierten Bilder in dem Codierer, einen Übertragungsschritt zum Übertragen der codierten Bilder an einen Decodierer, einen Decodierschritt zum Decodieren der codierten Bilder zum Ausbilden decodierter Bilder und einen Neuordnungsschritt zum Anordnen der decodierten Bilder in einer Anzeigeanordnung umfasst.

Allgemeiner Stand der Technik

[0002] Zu veröffentlichten Videocodierstandards gehören ITU-T H.261, ITU-T H.263, ISO/IEC MPEG-1, ISO/IEC MPEG-2 und ISO/IEC MPEG 4 Teil 2. Diese Standards sind hierin als herkömmliche Videocodierstandards bezeichnet.

[0003] Es geht eine Standardisierungsbemühung in einem Joint Video Team (JVT) aus ITU-T und ISO/IEC voran. Die Arbeit von JVT basiert auf einem früheren Standardisierungsprojekt in ITU-T namens H.26L. Das Ziel der JVT-Standardisierung ist, denselben Standardtext wie ITU-T Recommendation H.264 und ISO/IEC International Standard 14496-10 (MPEG-4 Teil 10) herauszugeben. Der Entwurfsstandard ist in diesem Papier als der JVT-Codierstandard bezeichnet, und der Code gemäß dem Entwurfsstandard als JVT-Codec.

[0004] Der optionale Bezugsbildauswahlmodus von H.263 und das NEWPRED-Codierwerkzeug von MPEG-4 Teil 2 ermöglichen die Auswahl des Bezugsrahmens zur Bewegungskompensierung für jedes Bildsegment, z.B. für jeden Slice in H.263. Ferner ermöglichen der verbesserte Bezugsbildauswahlmodus von H.263 und der JVT-Codierstandard die Auswahl des Bezugsrahmens für jeden Makroblock separat.

[0005] Bezugsbildauswahl ermöglicht viele Arten temporaler Skalierbarkeitsmodelle. [Fig. 1](#) zeigt ein Beispiel eines temporalen Skalierbarkeitsmodells, das hierin als rekursive temporale Skalierbarkeit bezeichnet ist. Das Beispielmodell kann mit drei konstanten Bildwechselfrequenzen decodiert werden. [Fig. 2](#) stellt ein Modell dar, das als Video Redundancy Coding bezeichnet ist, wobei eine Bildfolge überlappend in zwei oder mehr unabhängig codierte Threads geteilt ist. Die Pfeile in diesen und allen folgenden Figuren geben die Bewegungskompensierungsrichtung an, und die Werte unter den Frames entsprechen den relativen Erfassungs- und Anzeigzeiten der Frames.

[0006] Bei herkömmlichen Videocodierstandards ist die Decodierordnung von Bildern dieselbe wie die Anzeigeanordnung, außer bei B-Bildern. Ein Block in einem herkömmlichen B-Bild kann bidirektional temporal aus zwei Bezugsbildern vorhergesagt werden, wobei ein Bezugsbild in der Anzeigeanordnung temporal vorausgeht und das andere Bezugsbild temporal nachfolgt. Nur das letzte Bezugsbild in der Decodierordnung kann dem B-Bild in der Anzeigeanordnung nachfolgen (Ausnahme: Interlace-Codierung bei H.263, wo beide Feldbilder eines temporal nachfolgenden Bezugs-Frames einem B-Bild in der Decodierordnung vorausgehen können). Ein herkömmliches B-Bild kann nicht als Bezugsbild zur temporalen Voraussage benutzt werden, und daher kann ein herkömmliches B-Bild abgelegt werden, ohne die Decodierung jeglicher anderer Bilder zu beeinflussen.

[0007] Der JVT-Codierstandard beinhaltet die folgenden, im Vergleich zu früheren Standards neuartigen technischen Merkmale:

- Die Bilderdecodierordnung ist aus der Anzeigeanordnung ausgekoppelt. Die Bildzahl zeigt Decodierordnung und die Bildordnungszählung die Anzeigeanordnung an.
- Bezugsbilder für einen Block in einem B-Bild können entweder vor oder nach dem B-Bild in der Anzeigeanordnung vorliegen. Infolgedessen steht ein B-Bild für ein biprädictives Bild anstatt eines bidirektionalen Bilds.
- Bilder, die nicht als Bezugsbilder benutzt sind, sind ausdrücklich markiert. Ein Bild jeglicher Art (intra, inter, B usw.) kann entweder ein Bezugsbild oder ein Nichtbezugsbild sein. (Daher kann ein B-Bild als Bezugsbild für temporale Vorhersage von anderen Bildern benutzt sein.)
- Ein Bild kann Slices enthalten, die mit einer anderen Codierart codiert sind. Anders gesagt kann ein codiertes Bild beispielsweise aus einem intracodierten Slice und einem B-codierten Slice bestehen.

[0008] Das Auskoppeln der Anzeigeanordnung von der Decodierordnung kann vom Gesichtspunkt der Kompressionseffizienz und Federelastizität her günstig sein.

[0009] Ein Beispiel einer Vorhersagestruktur, die die Kompressionseffizienz potentiell verbessert, ist in [Fig. 3](#) dargestellt. Kästen geben Bilder an, Großbuchstaben in Kästen geben Codierarten an, Zahlen in Kästen sind Bildnummern gemäß dem JVT-Codierstandard, und Pfeile geben Vorhersageabhängigkeiten an. Es ist zu beachten, dass Bild B17 ein Bezugsbild für Bilder B18 ist. Die Kompressionseffizienz ist im Vergleich zu herkömmlicher Codierung potentiell verbessert, da die Bezugsbilder für Bilder B18 temporal näher im Vergleich zu herkömmlicher Codierung mit PDBP- oder PBBBP-codierten Bildmustern sind. Die Kompressionseffizienz ist im Vergleich zu herkömmlichen PBP-codierten Bildmustern potentiell verbessert, da ein Teil von Bezugsbildern bidirektional vorhergesagt ist.

[0010] [Fig. 4](#) stellt ein Beispiel des Intra-Bildverschiebungsverfahrens dar, das zum Verbessern der Fehlertoleranz benutzt sein kann. Herkömmlicherweise wird ein Intra-Bild beispielsweise umgehend nach einem Szenenschnitt oder in Reaktion auf eine abgelaufene Intra-Bildaktualisierungsperiode codiert. Bei dem Intra-Bildverschiebungsverfahren wird ein Intra-Bild nicht umgehend codiert, nachdem ein Erfordernis zum Codieren eines Intra-Bilds auftritt, sondern es wird stattdessen ein nachfolgendes Intra-Bild als Intra-Bild ausgewählt. Jedes Bild zwischen dem codierten Intra-Bild und dem herkömmlichen Standort eines Intra-Bilds wird von dem nächsten temporal folgenden Bild vorhergesagt. Wie [Fig. 4](#) zeigt, erstellt das Intra-Bildverschiebungsverfahren zwei unabhängige Inter-Bildvorhersageketten, während herkömmliche Codieralgorithmen eine einzige Inter-Bildkette erzeugen. Es ist intuitiv klar, dass der Zweikettenansatz gegenüber Lösungsfehlern robuster ist als der herkömmliche Einkettenansatz. Wenn eine Kette einen Paketverlust erleidet, kann die andere Kette weiterhin korrekt empfangen werden. Bei der herkömmlichen Codierung bewirkt ein Paketverlust stets eine Fehlerausbreitung auf den Rest der Inter-Bildvorhersagekette.

[0011] Beim JVT-Codierstandard müssen decodierte Bilder aus zwei Gründen gepuffert werden: erstens werden decodierte Bilder als Bezugsbilder zum Vorhersagen folgender codierter Bilder benutzt. Zweitens müssen decodierte Bilder aufgrund des Auskoppelns der Decodierordnung aus der Anzeigereihenfolge aufgezeichnet sein.

[0012] Das folgende Beispiel wird zum Erläutern des Problems separaten Pufferns benutzt, das die vorliegende Erfindung löst.

[0013] Betrachten Sie die folgende Bildersequenz, in der P ein vorhergesagtes Bild, BS ein biperädictives Bezugsbild und BN ein biperädictives Nichtbezugsbild ist und die Zahl die Anzeigereihenfolge betrifft:

Anzeigereihenfolge P1 BN2 BN3 BS4 BN5 BN6 P7 ...

Decodierordnung P1 P7 BS4 BN2 BN3 BN5 BN6

[0014] Dies kann mit drei Bildspeichern im Bezugsbildpuffer decodiert werden, aber wenn BN5 decodiert wird, ist es noch nicht Zeit, es anzuzeigen:

Decodierte P1 P7 BS4 BN2 BN3 BN5 BN6

Zeit

Anzeigezeit P1 BN2 BN3 BS4 BN5 BN6 P7 ...

[0015] Daher muss BN5 zum Neuordnen der Bilder in der Anzeigereihenfolge gespeichert werden.

[0016] Dieses Problem besteht bei herkömmlichen Videocodierstandards nicht, da die Anzeigereihenfolge für alle Bezugsbilder dieselbe wie ihre Decodierordnung ist, und da nur das letzte decodierte Bezugsbild zum Neuordnen von Bildern in der Anzeigereihenfolge gepuffert werden muss, wenn B-Bilder in Benutzung sind. Die herkömmlichen Videocodierstandards, die Bezugsbildauswahl unterstützen, weisen einen Bezugsbildpuffer auf, aber keinen Bildpuffer zur Anzeigeneuordnung.

[0017] Der folgende unkomplizierte Vorschlag wurde für den JVT-Codierstandard gemacht: Aufweisen eines Bildpuffers für Bezugsbilder, der von einem Bildpuffer zur Anzeigeneuordnung getrennt ist. Separates Spezifizieren der Höchstanzahl von Bildern für beide Puffer.

[0018] Betrachten wir nun das oben beschriebene Beispiel erneut. Ein Bezugsbild kommt in den Bezugsbildpuffer, sobald es decodiert ist. Ein Nichtbezugsbild kommt nicht in den Bezugsbildpuffer. Ein decodiertes Bild wird aus dem Bezugsbildpuffer entfernt, sobald es nicht weiter als Bezug benötigt wird. Beispielsweise kann Bild P1 nach dem Decodieren von Bild BN3 entfernt werden. Ein Bild kommt in den Anzeigeneuordnungspuffer, sobald es decodiert ist. Ein decodiertes Bild wird aus dem Anzeigeneuordnungspuffer entfernt, wenn es angezeigt werden kann. Die Figur unten zeigt den Inhalt der Puffer gerade nach dem Decodieren jeden Bilds.

Decodierte	P1	P7	BS4	BN2	BN3	BN5	BN6		
Zeit									
Anzeigezeit				P1	BN2	BN3	BS4	BN5	BN6 P7
Bezugspuffer	P1	P1	P1	P1	BS4	BS4	P7		
				P7	BS4	BS4	P7	P7	
				P7	P7				
Anzeigepuffer		P1	P1	BS4	BS4	BS4	BN5	BN6	P7
				P7	P7	P7	P7	P7	P7

[0019] Es ist ersichtlich, dass die erforderlichen Größen des Bezugsbildpuffers und des Anzeigeneuordnungspuffers 3 Bilder bzw. zwei Bilder betragen.

[0020] Die Arbeitsschrift JVT-B050, die für die zweite Zusammenkunft des Joint Video Teams (JVT) von ISO/IEC MPEG & ITU-T VCEG (ISO/IEC JTC1/SC29/WG11 und ITU-T SG16 Q.6) in Genf, Schweiz, vom 29. Januar bis 1. Februar 2002, erstellt wurde, schlägt einen neuen Videokomplexitätsprüfer vor. Dieser Prüfer kennzeichnet, wenn er als Teil des hypothetischen Bezugsdecodierers (HRD) benutzt wird, die Verzögerungsmenge und Puffergröße, die zum Decodieren und Präsentieren eines gegebenen Bitstroms auf einer bestimmten Rechenkapazitätsebene im Decodierer benötigt ist.

Kurzdarstellung der Erfindung

[0021] Bei der vorliegenden Erfindung sind der Bezugsbildpuffer und der Anzeigebildpuffer kombiniert, die Bezugsbilder sowie die Bilder, die neu geordnet werden sollen, im selben Puffer gespeichert werden, um das Erfordernis zu vermeiden, dasselbe Bild an zwei verschiedenen Stellen zu speichern.

[0022] Die vorliegende Erfindung stellt hypothetischen Decodierer, Signal, Codierer, Decodierer, Verfahren, Gerät, System und Computerprogramm gemäß den beiliegenden Ansprüchen zum Gebrauch in jeder Art Netzwerkgerät und Endgerät bereit, wobei derselbe Puffer zum Speichern am Decodierer der Bezugsbilder sowie der Bilder benutzt sein kann, die auf ihre Anzeige warten. Ferner sind die Bilder nicht zweimal im Speicher gespeichert.

[0023] Die Erfindung löst das Problem des decodierten Bildpuffers im JVT-Codierstandard, das bei herkömmlichen Videocodierstandards nicht bestand. Die Erfindung spart im Vergleich zu der anderen, für den JVT-Codierstandard vorgeschlagenen Lösung Speicher. Der Vorteil, dass der zum Puffern der Bilder benötigte Speicher minimiert sein kann, ist ein erheblicher Vorteil bei kleinen Geräten wie etwa mobilen Endgeräten.

Beschreibung der Zeichnungen

[0024] [Fig. 1](#) zeigt ein Beispiel eines rekursiven, temporalen Skalierbarkeitsmodells,

[0025] [Fig. 2](#) stellt ein Modell dar, das als Video Redundancy Coding bezeichnet wird, wobei eine Bildfolge überlappend in zwei oder mehr unabhängig codierte Threads geteilt ist,

[0026] [Fig. 3](#) zeigt ein Beispiel einer Vorhersagestruktur, die potentiell die Kompressionseffizienz verbessert,

[0027] [Fig. 4](#) zeigt ein Beispiel des Intra-Bildverschiebungsverfahrens dar, das zum Verbessern der Fehlertoleranz benutzt sein kann,

[0028] [Fig. 5](#) stellt eine vorteilhafte Ausführungsform des Systems gemäß der vorliegenden Erfindung dar,

[0029] [Fig. 6](#) stellt eine vorteilhafte Ausführungsform des Codierers gemäß der vorliegenden Erfindung dar,

[0030] [Fig. 7](#) stellt eine vorteilhafte Ausführungsform des Decodierers gemäß der vorliegenden Erfindung dar.

DETAILLIERTE BESCHREIBUNG DER ERFINDUNG

[0031] Bei der Einheitspufferlösung verbleiben alle decodierten Bilder in dem Einheitspuffer, bis sie nicht weiter als Bezugsbilder benutzt werden und bis ihre Anzeigereihenfolge kommt. Der Einheitspuffer spart im Vergleich zu separaten Bildpuffern für Bezugsbilder und zum Neuordnen in der Anzeigereihenfolge Speicher.

[0032] Betrachten wir das vorhergehende Beispiel erneut, wenn ein Einheitspuffer in Gebrauch ist.

Decodierte	P1	P7	BS4	BN2	BN3	BN5	BN6		
Zeit									
Anzeigezeit			P1	BN2	BN3	BS4	BN5	BN6	P7
Einheitspuffer	P1	P1	P1	P1	BS4	BS4	BN6	P7	
			P7	BS4	BS4	P7	BN5	P7	
P7	P7	P7							

[0033] Es gibt jedoch einige Fragen, die bei der Einheitspufferlösung berücksichtigt werden sollten. Die Größe des Puffers sollte groß genug zum Speichern aller der Bilder eingestellt sein, die noch benötigt werden, d.h. alle Bezugsbilder und Bilder, die neu geordnet werden sollen. Die Speicherkapazität ist häufig begrenzt, so dass die Größe des Puffers eine Art Kompromiss darstellt. Der Codierer muss überprüfen, ob der Decodierer imstande ist, genügend Bilder zum Decodieren und Ausgeben aller Bilder ohne Fehler zu speichern. Diese Überprüfung kann durch einen hypothetischen Bezugsdecodierer (HRD) in dem Codiergerät ausgeführt werden. Der HRD umfasst einen codierten Bildpuffer zum Speichern eines codierten Datenstroms und einen decodierten Bildpuffer zum Speichern von decodierten Bezugsbildern und zum Neuordnen von decodierten Bildern in Anzeigereihenfolge. Der HRD verschiebt Daten so ähnlich zwischen den Puffern wie ein Decodierer eines Decodiergeräts. Der HRD muss jedoch weder die codierten Bilder völlig decodieren noch die decodierten Bilder ausgeben, sondern der HRD überprüft nur, dass das Decodieren des Bilderstroms unter den in dem Codierstandard gegebenen Einschränkungen ausgeführt werden kann. Wenn der HRD arbeitet, empfängt er einen codierten Datenstrom und speichert ihn in dem codierten Bilderpuffer. Zudem entfernt der HRD codierte Bilder aus dem codierten Bilderpuffer und speichert zumindest einige der entsprechenden hypothetisch decodierten Bilder in dem decodierten Bilderpuffer. Der HRD kennt die Eingabegeschwindigkeit, gemäß welcher die codierten Daten in den codierten Bildpuffer strömen, die Entfernungsgeschwindigkeit der Bilder aus dem codierten Bildpuffer und die Ausgabegeschwindigkeit der Bilder aus dem decodierten Bildpuffer. Der HRD prüft codierte oder decodierte Bildpufferüberläufe und zeigt an, falls das Decodieren mit den derzeitigen Einstellungen nicht möglich ist. Dann informiert der HRD den Codierer über die Pufferverletzungen, wobei der Codierer die Codierparameter beispielsweise durch Reduzieren der Anzahl von Bezugs-Frames ändern kann, um eine Pufferverletzung zu vermeiden. Alternativ oder zusätzlich beginnt der Codierer, die Bilder mit den neuen Parametern zu codieren, und sendet die codierten Bilder an den HRD, der wiederum das Decodieren der Bilder und die notwendigen Prüfungen ausführt. Als wiederum andere Alternative kann der Codierer den letzten codierten Frame verwerfen und spätere Frames codieren, sodass keine Pufferverletzung vorkommt.

Slice-basierter Betrieb

[0034] Die empfohlene Mindesteinheit zum Einkapseln in einem Übertragungspaket ist ein Slice oder Datenpartition. Die verhältnismäßig unabhängig decodierbare Mindesteinheit ist ein Slice. Infolgedessen setzt das Einheitspuffermodell voraus, dass jeweils ein ganzer Slice oder eine ganze Datenpartition übertragen wird. Das Modell setzt außerdem voraus, dass paketbasierte Netzwerke und Übertragungsprotokolle benutzt werden und daher jeweils ein ganzer Slice oder eine ganze Datenpartition empfangen wird.

[0035] Es wird vorausgesetzt, dass jeweils ein Slice decodiert wird. Daher werden Daten Slice für Slice aus dem hypothetischen Eingabepuffer entfernt. Das vorgeschlagene Komplexitätsmodell wird zum Bestimmen der Zeit benutzt, die benötigt wird, einen Slice virtuell zu decodieren. Wenn die Zeit abgelaufen ist, kann der nächste Slice virtuell decodiert werden.

Übersicht des Puffermodells der vorliegenden Erfindung

[0036] Der Bezugsdecodierer basiert auf einem Puffermodell, das in diesem Abschnitt dargestellt ist. Das Modell basiert auf drei Puffern: dem Vordecodiererpuffer, dem Decodierpuffer und dem Nachdecodiererpuffer. Der Vordecodiererpuffer puffert komprimierte Daten. Der Betrieb des Decodierpuffers verkörpert den Betrieb eines

echten Decodierers und stellt die Verarbeitungskomplexitätsbeschränkungen ein. Der Nachdecodiererpuffer enthält rekonstruierte Bilder, die als Bezugsbilder zur Bewegungskompensierung benutzt werden und/oder auf einen richtigen Augenblick zur virtuellen Anzeige warten.

[0037] Die Standardpuffergrößen sind in den Profil- und Ebenendefinitionen angegeben. Die Puffergrößen sind außerdem verhandelbar. Die Größe des Decodierpuffers gibt die Maximalgröße eines Slice an. Die Größe des Nachdecodiererpuffers kann beispielsweise in Byte oder Einheiten von 16×16 angegeben sein.

[0038] Der Betrieb des Vor- und Nachdecodiererpuffers kann durch zwei Parameter gesteuert sein. Die anfängliche Vordecodiererpufferperiode und die anfängliche Nachdecodiererpufferperiode definieren, wie lange die Puffer gefüllt werden, bevor jegliche Daten daraus entfernt werden. Die Perioden werden für jeden Bitstrom separat signalisiert.

[0039] Drei Zeitstempel können einer Dateneinheit (einer Datenpartition, einem Slice oder einem Bild) zugeordnet sein. Jedem Slice und jeder Datenübertragung ist eine Übertragungszeit zugeordnet. Im Falle eines Dateizugriffs ist die Übertragungszeit in der Datei angezeigt. Andernfalls wird die echte Übertragungszeit benutzt. Jedem Slice kann eine Decodierzeit zugeordnet sein. Die Decodierzeit kann zum Verzögern des Decodierens eines Slice benutzt sein, um Überläufe des Nachdecodiererpuffers zu vermeiden. Jedem Bild ist eine Anzeigzeit zugeordnet.

[0040] Zeitbezogene Parameter sind in einer gemeinsamen Zeitskala angegeben, wie etwa Urticken einer 90-kHz- oder einer 27-MHz-Uhr.

[0041] Das Modell kann auf jeden fortlaufenden und unabhängig bestimmbar Teil eines codierten Datenstroms Anwendung finden. Im Falle von Unterbrechungen, wie etwa dem Zugreifen auf einen Bitstrom von einer Zufallsposition, wird das Modell rückgesetzt.

Funktion des Puffermodells

[0042] Das Modell ist definiert wie folgt:

1. Die Puffer sind anfangs leer.
2. Ein Slice oder eine Datenpartition wird unter Ausnahme seines Headers dem Vordecodiererpuffer zu seiner Übertragungszeit hinzugefügt. Eine bestimmte Speichermenge, beispielsweise acht Byte, zusätzlichen Speichers wird für jeden Slice zum Speichern seines Headers reserviert. (Eigentliche Slice-Header können eng mit spezifischen Daten von Nutzdaten-Headern vernetzt werden, wie etwa den Parametereinstellungsindikator im RTP-Entwurfsnutzdatenformat (VCEG-N72R1). Es ist unpraktisch, diese Art Daten nur für HRD-Zwecke aus dem Header-Abschnitt des Slice zu entfernen. Dadurch wird eine konstante Größe des Slice-Headers vorausgesetzt.)
3. während einer anfängliche Vordecodiererpufferperiode genannten Periode werden keine Daten aus dem Vordecodiererpuffer entfernt. Die Periode beginnt, wenn dem Puffer der erste Slice oder Datenpartition hinzugefügt wird.
4. Wenn die anfängliche Vordecodiererpufferperiode abgelaufen ist und falls Decodierzeitstempel in Benutzung sind, wird der Decodierzeitgeber ab der Decodierzeit des frühesten Bilds im Bildpuffer in Gang gesetzt. Andernfalls wird kein Decodierzeitgeber benötigt.
5. Bilder werden Slice für Slice in ansteigender Bildkennungsordnung (unter Anwendung von Modulo-Arithmetik) aus dem Vordecodiererpuffer entfernt. Slices eines bestimmten Bilds werden in ihrer Übertragungsordnung entfernt. Ein Slice mit seinen Datenpartitionen wird jeweils vom Vordecodiererpuffer zum Decodierpuffer verschoben, wenn der Decodierpuffer leer wird. Ein Slice wird jedoch nicht aus dem Vordecodiererpuffer entfernt, bevor der Decodierzeitgeber die Decodierzeit des Slice erreicht.
6. Ein Slice verbleibt so lange wie seine berechnete Decodierdauer im Decodierpuffer. Die Dauer ist die größere von zwei Kandidaten: Kandidat 1 = Anzahl von Makroblöcken in einem Slice/maxMacroblockPerSec und Kandidat 2 = Anzahl von Bit in einem Slice/maxBitsPerSec.
7. Ein dekomprimiertes Bild tritt in den Nachdecodiererpuffer ein, wenn der letzte Slice (in der Decodierordnung) des entsprechenden codierten Bilds aus dem Decodierpuffer entfernt wird.
8. Während einer anfängliche Nachdecodiererpufferperiode genannten Periode werden keine Daten aus dem Nachdecodiererpuffer entfernt. Die Periode beginnt, wenn das erste Bild in den Nachdecodiererpuffer verbracht wird.
9. Wenn die anfängliche Nachdecodiererpufferperiode abgelaufen ist, wird der Wiedergabezeitgeber ab der frühesten Anzeigzeit der Bilder, die zu diesem Zeitpunkt im Nachdecodiererpuffer stehen, in Gang gesetzt.
10. Ein Bild wird virtuell angezeigt, wenn der Wiedergabezeitgeber die planmäßige Anzeigzeit des Bilds

erreicht.

11. Ein Bild wird aus dem Nachdecodiererpuffer entfernt, wenn es virtuell angezeigt wird und wenn es nicht länger als Bezugsbild benötigt wird.

Anforderungen für einen konformen Bitstrom

[0043] Jeglicher übertragene oder gespeicherte Bitstrom sollte folgende Anforderungen erfüllen:

- Die Belegung des Vordecodiererpuffers soll die Standard- oder signalisierte Puffergröße nicht übersteigen.
- Jeglicher Slice soll nicht größer als die Größe des Decodierpuffers sein.
- Die Belegung des Nachdecodiererpuffers soll die Standard- oder signalisierte Puffergröße nicht übersteigen.
- Jedes Bild soll vor oder zu seiner Anzeigzeit in den Nachdecodiererpuffer eingefügt sein.

[0044] Wenn die Standardpuffergrößen gemäß einem/er bestimmten Profil und Ebene benutzt sind, ist ein Bitstrom mit diesem Profil und dieser Ebene konform.

Anforderungen für einen konformen Decodierer

[0045] Ein Decodierer soll imstande sein, alle Bilder in einem konformen Bitstrom zu empfangen und zu decodieren, wenn sowohl im Decodierer als auch im Bitstrom dieselben Puffergrößen vorausgesetzt sind. Ferner soll der Decodierer jedes Bild zur selben Zeit zu einem Anzeigeprozess weiterleiten, zu der der hypothetische Bezugsdecodierer das Bild virtuell anzeigen würde.

[0046] Im Folgenden wird die Erfindung unter Bezugnahme auf das System von **Fig. 5**, den Codierer **1** und hypothetischen Bezugsdecodierer (HRD) **5** von **Fig. 6** und den Decodierer **2** von **Fig. 7** detaillierter beschrieben. Die Bilder, die codiert werden sollen, können beispielsweise Bilder eines Videostroms von einer Videoquelle **3** sein, z.B. einer Kamera, einem Videorecorder usw. Die Bilder (Frames) des Videostroms können in kleinere Abschnitte wie etwa Slices unterteilt sein. Diese Slices können ferner in Blöcke aufgeteilt sein. In dem Codierer **1** wird der Videostrom codiert, um die Information zu reduzieren, die über einen Übertragungskanal **4** oder an ein Speichermittel (nicht gezeigt) übertragen werden soll. Bilder des Videostroms werden in den Codierer **1** eingegeben. Der Codierer weist einen Codierpuffer **1.1** (**Fig. 6**) zum temporalen Speichern von einigen der Bilder auf, die codiert werden sollen. Der Codierer **1** beinhaltet außerdem einen Speicher **1.3** und einen Prozessor **1.2**, in dem die Codieraufgaben gemäß der Erfindung ausgeführt werden können. Der Speicher **1.3** und der Prozessor **1.2** können mit dem Übertragungsgerät **6** gemeinsam sein, oder das Übertragungsgerät **6** kann einen weiteren Prozessor und/oder Speicher (nicht gezeigt) für andere Funktionen des Übertragungsgeräts **6** aufweisen. Der Codierer **1** führt Bewegungsschätzung und/oder andere Aufgaben zum Komprimieren des Videostroms aus. Bei der Bewegungsschätzung werden Ähnlichkeiten zwischen dem Bild, das codiert werden soll (dem derzeitigen Bild), und einem vorhergehenden und/oder späteren Bild gesucht. Wenn Ähnlichkeiten gefunden werden, kann das Vergleichsbild oder ein Teil davon als Bezugsbild für das Bild benutzt werden, das codiert werden soll. Bei JVT ist die Anzeigereihenfolge und die Decodierereihenfolge der Bilder nicht notwendigerweise dieselbe, wobei das Bezugsbild in einem Puffer (z.B. im Codierpuffer **1.1**) gespeichert sein muss, solange es als Bezugsbild benutzt wird. Der Codierer **1** fügt außerdem Information zur Anzeigereihenfolge der Bilder in den Übertragungsstrom ein.

[0047] Vom Codierprozess werden die codierten Bilder bei Bedarf in einen optionalen codierten Bildpuffer **1.5** verschoben. Die codierten Bilder werden vom Codierer **1** über den Übertragungskanal **4** an den Decodierer **2** übertragen. Im Decodierer **2** werden die codierten Bilder zum Ausbilden von dekomprimierten Bildern decodiert, die so weit wie möglich den codierten Bildern entsprechen. Jedes decodierte Bild wird im DPB **2.1** des Decodierers **2** gespeichert, wenn es nicht im Wesentlichen unverzüglich nach dem Decodieren angezeigt wird und nicht als Bezugsbild benutzt wird. In dem System gemäß der vorliegenden Erfindung ist das Bezugsbildpuffer sowie das Anzeigebildpuffer kombiniert, und sie benutzen denselben decodierten Bildpuffer **2.1**. Dies beseitigt das Erfordernis, dieselben Bilder an zwei verschiedenen Stellen zu speichern, wodurch die Speicheranforderungen des Decodierers **2** reduziert sind.

[0048] Der Decodierer **2** beinhaltet außerdem einen Speicher **2.3** und einen Prozessor **2.2**, in dem die Decodieraufgaben gemäß der Erfindung ausgeführt werden können. Der Speicher **2.3** und der Prozessor **2.2** können mit dem Empfangsgerät **8** gemeinsam sein, oder das Empfangsgerät **8** kann einen weiteren Prozessor und/oder Speicher (nicht gezeigt) für andere Funktionen des Empfangsgeräts **8** aufweisen.

[0049] Wir wollen nun den Codier-Decodier-Prozess detaillierter betrachten. Bilder aus der Videoquelle **3** werden in den Codierer **1** eingegeben und vorteilhafterweise im Codierpuffer **1.1** gespeichert. Der Codierprozess wird nicht notwendigerweise unmittelbar nach der Eingabe des ersten Bilds in den Codierer in Gang gesetzt, sondern nachdem eine bestimmte Menge Bilder im Codierpuffer **1.1** verfügbar ist. Dann versucht der Codierer **1**, geeignete Kandidaten aus den Bildern herauszufinden, die als Bezugs-Frames benutzt werden sollen. Dann führt der Codierer **1** das Codieren zum Ausbilden codierter Bilder aus. Die codierten Bilder können beispielsweise vorhergesagte Bilder (P), biprädiktive Bilder (B) oder intracodierte Bilder (I) sein. Die intracodierten Bilder können ohne die Benutzung jeglicher anderer Bilder decodiert werden, aber andere Bilderarten benötigen zumindest ein Bezugsbild, bevor sie decodiert werden können. Bilder von jeglicher der oben angegebenen Bildarten können als Bezugsbild benutzt werden.

[0050] Der Codierer den Bildern zwei Zeitstempel bei: einen Decodierzeitstempel (DTS) und einen Ausgabezeitstempel (OTS). Der Decodierer kann die Zeitstempel zum Bestimmen der korrekten Decodierzeit und Zeit zum Ausgeben (Anzeige) der Bilder benutzen. Diese Zeitstempel werden jedoch nicht notwendigerweise an den Decodierer übertragen, oder er benutzt sie nicht.

[0051] Wenn der Codierer den HRD **5** aufweist, definiert der Codierer die Größe des DPB **5.2** (DPB-Größe). Die Anfangsgröße kann von einigen Parametern bezüglich des Videostroms (z.B. Auflösung, Farbe/SW usw.) abhängen. Es kann außerdem eine Maximalgröße für den DPB **5.2** (DPB-Kapazität) definiert sein. Die Anfangsgröße muss nicht notwendigerweise dieselbe wie die Maximalgröße sein, sondern kann außerdem kleiner als die Maximalgröße sein. Außerdem ist die Höchstzahl von im DPB gespeicherten Bezugs-Frames definiert. Dieser Wert wird in dieser Beschreibung als num_ref_frames bezeichnet.

[0052] Im Decodierer weist der DPB **2.1** eine begrenzte Größe (DPB-Größe) auf, die bei einigen Anwendungen während des Decodierprozesses bei Bedarf verändert werden kann. Die Anfangsgröße des DPB **2.1** kann von einigen Parametern bezüglich des Videostroms (z.B. Auflösung, Farbe/SW usw.) abhängen. Es kann außerdem eine Maximalgröße für den DPB **2.2** (DPB-Kapazität) definiert sein. Die Anfangsgröße muss nicht notwendigerweise dieselbe wie die Maximalgröße sein, sondern kann außerdem kleiner als die Maximalgröße sein. Außerdem ist die Höchstzahl von im DPB gespeicherten Bezugs-Frames definiert. Dieser Wert wird in dieser Beschreibung als num_ref_frames bezeichnet.

[0053] Die Übertragung und/oder Speicherung der codierten Bilder (und des optionalen virtuellen Decodierens) kann, unmittelbar nachdem das erste codierte Bild bereit ist, in Gang gesetzt werden. Dieses Bild ist nicht notwendigerweise das erste in der Decodiererausgabeordnung, da die Decodierordnung und die Ausgabeordnung nicht dieselbe sein könnte. Da die Größe des DPB **2.1** des Decodierers **2** begrenzt ist, ist es jedoch notwendig zu definieren, wie lange die Verzögerung zwischen der Decodierzeit eines Bilds und der Anzeigzeit des Bilds maximal sein kann, d.h. die Maximalzahl der Bilder, die zum Anzeigen neu geordnet werden sollen. Dieser Wert wird in dieser Beschreibung als num_reorder_frames bezeichnet. num_reorder_frames ist die Anzahl aufeinander folgender Frames in Decodierordnung, die zum Wiederherstellen der Bildausgabeordnung zu jeder Zeit erforderlich ist. Die Summe der Höchstanzahl von im DPB gespeicherten Bezugs-Frames und der Höchstanzahl der Bilder, die zum Anzeigen neu geordnet werden sollen, unter Ausnahme jener Bilder, die bereits als Bezugsbilder gespeichert sind, darf nicht größer als die derzeitige Kapazität des DPB sein. Die DPB-Größe stellt dar, wie viele Bilder im DPB **2.1** gespeichert werden können. Sie kann durch Dividieren der Größe des DPB (in Byte) durch die Größe eines Bilds (in Byte) berechnet werden.

[0054] Wenn das erste Bild des Videostroms codiert ist, kann die Übertragung in Gang gesetzt werden. Die codierten Bilder werden optional im codierten Bildpuffer **1.5** gespeichert. Die Übertragung kann außerdem in einer späteren Phase beginnen, nachdem ein bestimmter Teil des Videostroms codiert wurde.

[0055] Der Decodierer **2,5** sollte die decodierten Bilder in korrekter Ordnung ausgeben, beispielsweise durch Benutzung der Ordnung der Bildordnungszählungen, und daher muss der Neuordnungsprozess eindeutig und normativ definiert werden.

[0056] Als nächstes wird die Funktion des Decodierers **2** gemäß einer Ausführungsform der vorliegenden Erfindung beschrieben. Der DPB **2.1** enthält Speicherplätze zum Speichern einer Anzahl von Bildern. Diese Plätze werden in dieser Beschreibung außerdem Frame-Speicher genannt. Ein Attribut FrameStoreUsage wird hier als Indikator des Reservierungsstatus eines Frame-Speichers (benutzt/nicht benutzt) verwendet. Der Wert FrameStoreUsage gibt die derzeitige Frame-Speichernutzung wieder. Ein erster Wert (z.B. 0) bedeutet, dass der Frame-Speicher leer ist und sicher überschrieben werden kann. Ein zweiter Wert (z.B. 1) bedeutet, dass dies ein als Bezug (aber nicht zum Neuordnen) benutzter Frame-Speicher ist, ein dritter Wert (z.B. 2) bedeutet,

dass dies ein zum Neuordnen (aber nicht als Bezug) benutzter Frame-Speicher ist. Ein vierter Wert (z.B. 3) bedeutet, dass dies ein sowohl als Bezug als auch zum Neuordnen benutzter Frame-Speicher ist. Der Anfangswert von FrameStoreUsage soll der erste Wert sein. Es ist außerdem möglich ein separates, zweites Attribut zu benutzen, um zu definieren, ob ein Bild (virtuell) angezeigt wird oder nicht. Es kann „Zur Ausgabe nicht benutzt“- und „Zur Ausgabe benutzt“-Werte haben. Ein als „zur Ausgabe benutzt“ markiertes Bild bedeutet, dass das Bild noch im DPB verbleibt und darauf wartet, zur Ausgabe an die Reihe zu kommen. Es soll infolge der Ausgabe des Bilds als „zur Ausgabe nicht benutzt“ markiert werden. Wenn das zweite Attribut benutzt wird, würden nur der erste und zweite Wert mit FrameStoreUsage benutzt werden, wobei die Kombination des ersten und zweiten Attributs dieselbe Information wie FrameStoreUsage mit den vier verschiedenen Wertalternativen ergibt. Es ist außerdem möglich, ein Attribut für die Gültigkeit eines Bilds (gültig/ungültig) zu definieren. Ein Bild wird als „ungültig“ markiert, wenn es absichtlich zum Ausfüllen der erkannten Lücken zwischen benachbarten frame_num eingefügt ist. Andernfalls wird das Bild als „gültig“ markiert, das korrekt decodiert oder mithilfe von Fehlermaskierung rekonstruiert ist. Die Frame-Speicher können außerdem eindeutig indiziert sein, was die Decodierordnung der gepufferten Bilder einbezieht.

[0057] Am Anfang des Decodiervorgangs ist der Status aller Frame-Speicher auf leer oder frei eingestellt. Dies erfolgt durch Einstellen der FrameStoreUsage auf den ersten Wert für alle Frame-Speicher. Der Decodierer **2** kann außerdem die folgenden Variablen auf einen Anfangswert (z.B. 0) einstellen: num_frames_in_use, das die Anzahl von für entweder Bezug oder Ausgabe benutzten Frame-Speicher enthält; num_frames_use_ref, das die Anzahl von für Bezug benutzten Frame-Speicher enthält; und num_frames_use_output, das die Anzahl von für Ausgabe benutzten Frame-Speicher enthält.

[0058] Ein zusätzlicher Frame-Speicher wird zum Sichern des neu decodierten Bilds benutzt. Der DPB plus der zusätzliche Frame-Speicher wird erweiterter DPB (EDPB) genannt.

Schritt 1: Decodieren

[0059] Der Decodierer **2** beginnt das Decodieren der codierten Bilder beginnend von den Bildern mit dem kleinsten Decodierzeitstempel. Wenn das Bild decodiert ist, wird es im zusätzlichen Frame-Speicher gespeichert. Das Bild, das decodiert werden soll, kann entweder ein Frame oder ein Feld sein. Wenn es ein Bezugsbild ist, wird FrameStoreUsage auf den vierten Wert eingestellt, andernfalls auf den dritten Wert. Es wird zunächst als „zum Neuordnen benutzt“ etikettiert, da bei diesem Schritt ohne die Information über die DPB nicht bestimmt werden kann, ob das neu decodierte Bild eine Neuordnung benötigt.

Schritt 2: Neuordnen

[0060] Vor den folgenden Schritten muss der Decodierer **2** wissen, ob zumindest noch ein Bild, das decodiert werden soll, und Information vom Slice-Header des nächsten Bildes, das decodiert werden soll, vorliegt.

Schritt 2.1

[0061] Wenn zumindest noch ein Bild, das decodiert werden soll, vorliegt und die Anzahl von Frames, die im DPB neu geordnet werden sollen, weniger als num_reorder_frames ist, gibt der Decodierer **2** kein Bild aus. Wenn das neu decodierte Bild als Bezug benutzt wird, dann wird Schritt 3 ausgeführt, andernfalls wird Schritt 4 ausgeführt.

Schritt 2.2

[0062] Wenn zumindest noch ein Bild, das decodiert werden soll, vorliegt und die Anzahl von Frames, die im DPB neu geordnet werden sollen, gleich num_reorder_frames ist, werden folgende Bedingungen geprüft:

- Das neu decodierte Bild ist ein Frame.
- Der zusätzliche Frame-Speicher enthält zwei Felder.
- Wenn ein einzelnes Feld in dem zusätzlichen Frame-Speicher steht, ist das nächste Bild, das decodiert werden soll, nicht seine entgegengesetzte Parität desselben Frames.

[0063] Wenn keine der obigen Bedingungen wahr ist, weiter zu Schritt 1 zum Decodieren des entgegengesetzten Paritätsfelds desselben Frames.

[0064] Wenn zumindest eine der obigen Bedingungen wahr ist, werden dann die Frame-Speicher im DPB und der zusätzliche Frame-Speicher durchsucht, um herauszufinden, welches Bild ausgegeben werden muss. Das

Bild, das das erste Bild in der Ausgabeordnung unter den Bildern ist, die im DPB und dem zusätzlichen Frame-Speicher neu geordnet werden sollen, soll ausgegeben werden. Wie das erste Bild in der Ausgabeordnung zu bestimmen ist, wird später beschrieben. Wenn das ausgegebene Bild ein Feld und das entgegengesetzte Paritätsfeld desselben Frames vorhanden ist, werden beide Felder ausgegeben. Die Ausgabebilder sollten als „zum Neuordnen nicht benutzt“ markiert werden.

[0065] Wenn das neu decodierte Bild ausgegeben wird und nicht als Bezug benutzt wird, dann weiter zu Schritt 1 zum Decodieren des nächsten Bilds. Wenn das neu decodierte Bild als Bezug (jedoch nicht zum Neuordnen) benutzt wird, wird FrameStoreUsage auf den zweiten Wert (1) eingestellt und Schritt 3 folgt. Wenn das neu decodierte Bild zum Neuordnen (aber nicht als Bezug) benutzt wird, wird FrameStoreUsage auf einen dritten Wert (2) eingestellt und weiter zu Schritt 4 zum Speichern des Bilds zur Neuordnung. Wenn das neu decodierte Bild sowohl als Bezug als auch zum Neuordnen benutzt wird, wird FrameStoreUsage auf einen vierten Wert (3) eingestellt und weiter zu Schritt 3.

Schritt 2.3

[0066] Wenn keine Bilder mehr zum Decodieren vorliegen, sollen alle Bilder, die weiterhin als „zum Neuordnen benutzt“ markiert sind, in Ausgabeordnung ausgegeben werden. Dann hört der Decodierer auf zu arbeiten, bis mehr Bilder zum Decodieren vorliegen.

Schritt 3: Bilder als „als Bezug nicht benutzt“ markieren

[0067] Wenn das neu decodierte Bild ein Bezugsbild ist, wird ein „Sliding Window“- oder „Adaptive Memory Control“-Mechanismus auf den Frame-Speichersatz angewendet, der aus mit „als Bezug benutzt“ markierten Bildern besteht.

Schritt 4: Speichern

[0068] Wenn das neu decodierte Bild ein Frame ist, wird Schritt 4.1 ausgeführt, andernfalls Schritt 4.2.

Schritt 4.1

[0069] Es sollte vor dem Speichern des neu decodierten Frames zumindest einen leeren Frame-Speicher im DPB **2.1** geben. FrameStoreUsage wird gleich FrameStoreUsage des zusätzlichen Frame-Speichers eingestellt. Weiter zu Schritt 1 zum Decodieren des nächsten Bilds.

Schritt 4.2

[0070] Für ein neu decodiertes Feld ist der Frame-Zielspeicher ein leerer Frame-Speicher, wenn sein entgegengesetztes Paritätsfeld desselben Frames nicht im DPB **2.1** ist. In diesem Falle wird FrameStoreUsage gleich FrameStoreUsage des zusätzlichen Frame-Speichers eingestellt. Andernfalls sollte das neu decodierte Bild im selben Frame-Speicher wie sein entgegengesetztes Paritätsfeld gespeichert werden und FrameStoreUsage unverändert sein. Weiter zu Schritt 4.3 zum Decodieren des nächsten Bilds.

Schritt 4.3

[0071] Nach dem Speichern des neu decodierten Bilds im DPB wird vorausgesetzt, dass das letzte decodierte Bild in einem Frame-Speicher gespeichert ist, dessen FrameStoreIndex 0 ist, dass das als vorletztes decodierte Bild in einem Frame-Speicher gespeichert ist, dessen FrameStoreIndex 1 ist usw. Zuletzt wird Schritt 1 zum Decodieren des nächsten Bilds ausgeführt.

[0072] Es ist zu beachten, dass das Markieren eines Bilds als „als Bezug nicht benutzt“ und das Markieren eines Bilds als „zum Neuordnen nicht benutzt“ entkoppelt ist.

[0073] Um das Verständnis der Vorgänge in den oben angegebenen Schritten zu unterstützen, sind zwei Beispiele für den Neuordnungsprozess dargestellt. Die kursiv gedruckten Zahlen nach (oder unter) den Bildetiketten geben die frame_num an. Bezugsbilder sind durch Unterstreichen des Buchstabens des Bildetiketts angezeigt. „x“ bedeutet, dass kein Bild vor dem Decodieren des nächsten Bilds neugeordnet wird.

Beispiel 1. num_reorder_frames = 1; 10 Frames sind codiert.

Codierte Bilder in Ausgabeordnung	I0	B1	B2	P3	B4	B5	P6	B7	B8	P9
Decodierordnung:	<u>I</u> 0 0	<u>P</u> 3 1	B1 2	B2 2	<u>P</u> 6 2	B4 3	B5 3	<u>P</u> 9 3	B7 4	B8 4 P9
Nach dem Neuordnen:	x	I0	B1	B2	P3	B4	B5	P6	B7	B8, P9

Beispiel 2. num_reorder_frames = 2; 13 Frames sind codiert.

Codierte Bilder in Ausgabeordnung	I0	B1	B2	B3	P4	B5	B6	B7	P8	B9	B10	B11	P12
Decodier-ordnung	<u>I</u> 0 0	<u>P</u> 4 1	<u>B</u> 2 2	B1 3	B3 3	<u>P</u> 8 3	<u>B</u> 6 4	B5 5	B7 5	<u>P</u> 12 5	<u>B</u> 10 6	B9 7	B11 7 P12
Nach dem Neuordnen	x	x	I0	B1	B2	B3	P4	B5	B6	B7	P8	B9	B10, B11, B12

[0074] Wenn der Decodierer 2 bemerkt, dass die codierten Bilder nicht decodiert werden können, könnte die Größe des DPB 2.1 zu klein sein. Bei einigen Anwendungen kann der Decodierer 2 den Codierer informieren, die Codierparameter zu ändern, beispielsweise weniger Bezugsbilder zu benutzen oder eine größere DPB-Größe zu benutzen. In der Fehlersituation kann der Decodierer 2 weiterhin jene decodierten Bilder ausgeben, die er zu decodieren bewerkstelligt hat, und/oder einige Standardbilder (z.B. leere Frames) ausgeben.

„Sliding Window“-Vorgang zum Neuordnen

[0075] Num_reorder_frames wird zum Definieren des „Sliding Window“-Vorgangs zum Zweck des Neuordnens benutzt. Die Bilder, die neu geordnet werden sollen, werden in Ausgabeordnung angeordnet, um das Bild herauszufinden, das zuerst aus dem Bereich des Fensters springen soll. Wenn die Anzahl der Bilder, die neu geordnet werden sollen, num_reorder_frames erreicht, ist es sicher, dass das erste Bild in Ausgabeordnung infolge eines „Sliding Window“-Vorgangs aus dem Fenster fällt. Dieser Unterabschnitt spezifiziert, wie das früheste Bild in Ausgabeordnung unter den Bildern herauszufinden ist, die als „zum Neuordnen benutzt“ markiert sind.

Schritt 1. Initialisierung.

[0076] Das älteste Bild in Decodierordnung im DPB als Kandidatenbild einstellen.

Schritt 2.

[0077] Schleife vom zweitältesten Bild zum letzten Bild (in Decodierordnung) herstellen und i als Schleifen-zähler benutzen. Der Schleifenkörper ist durch Schritt 2.1 und 2.2 gebildet.

Schritt 2.1

[0078] Wenn Bild i ein IDR-Bild ist, ist das Kandidatenbild das Bild, das gefunden werden soll, Schleife abbrechen.

Schritt 2.2

[0079] Wenn POC von Bild i weniger als POC des Kandidatenbilds ist, Bild i als Kandidatenbild einstellen.

[0080] Das Kandidatenbild ist das älteste Bild in Ausgabeordnung.

[0081] Num_ref_frames und num_reorder_frames werden dem Decodierer **2** auf eine Art und Weise signalisiert, beispielsweise durch einen Sequenzparametersatz. Jedesmal wenn ein anderer Sequenzparametersatz wirksam wird, können sich num_ref_frames oder num_reorder_frames ändern, und daher sollte der DPB besondere Beachtung finden.

1.5.1 Ändern von num_ref_frames

[0082] Es sind keine besonderen Tätigkeiten erforderlich, wenn num_ref_frames zunimmt.

[0083] Wenn num_ref_frames abnimmt, wird eine einfache Einwirkung auf den DPB vorgeschlagen: Lassen wir diff_ref_frames gleich dem neuen num_ref_frames minus alte num_ref_frames sein. Alle diff_ref_frames Bezugs-Frames ab dem ältesten in Decodierordnung sollen als „als Bezug nicht benutzt“ markiert werden.

1.5.2 Ändern von num_reorder_frames

[0084] Wenn num_reorder_frames zunimmt, wird 0 Bild pro ein decodiertes Bild in (neue num_reorder_frames) – (alte num_reorder_frames) Bildern nach (alte num_reorder_frames) Bild ab dem decodierten Bild, das den neuen Sequenzparametersatz benutzt, ausgegeben. Beispielsweise sind die num_reorder_frames von zwei Teilen des Bitstroms 1 bzw. 2. Wenn P3 decodiert wird, wird kein Bild ausgegeben.

Decodierordnung	I0	P2	B1	P4	B3	I0	P3	B2	B1	P6	B5	B4
Nach dem Neuordnen	x	I0	B1	P2	B3	P4	x	I0	B1	B2	P3	B4, B5, B6

[0085] Wenn num_reorder_frames abnimmt, wird mehr als ein Bild am decodierten Bild unter Benutzung des neuen Sequenzparametersatzes ausgegeben. Beispielsweise sind die num_reorder_frames von zwei Teilen des Bitstroms 2 bzw. 1. Wenn I0 decodiert wird, werden zwei Bilder B5 und P6 zur selben Zeit ausgegeben.

Decodierordnung	I0	P3	B2	B1	P6	B5	B4	I0	P2	B1	P4	B3
Nach dem Neuordnen	x	x	I0	B1	B2	P3	B4	B5, P6	I0	B1	P2	B3, P4

[0086] Ein Bild, das zur Vorhersage gespeichert wird, als Bezugbild bezeichnet, wird als „als Bezug benutzt“ markiert und als kurzzeitiges Bezugbild etikettiert, welchenfalls es durch seine frame_number und Parität identifiziert wird. Einem kurzzeitigen Bild kann ein Langzeit-Frameindex zugeordnet werden, welchenfalls es durch seinen Langzeitindex identifiziert wird. Ein Bild, das nicht zur Vorhersage gespeichert wird, wird als „zur Vorhersage nicht benutzt“ etikettiert. Der Bezugsbildpuffer ist ein Bildsatz, der durch Bilder gebildet ist, die als „als Bezug benutzt“ im DPB markiert sind. Das Entfernen eines Bilds aus dem Bezugsbildpuffer bedeutet das Markieren des Bilds als „als Bezug nicht benutzt“. Das Rücksetzen des Bezugsbildpuffers bedeutet, alle Bilder im Bezugsbildpuffer als „als Bezug nicht benutzt“ zu markieren. Kurzzeitige Bezugsbilder können als Bezug über eine begrenzte Dauer markiert bleiben, die durch MaxFrameNum gegeben ist. Langzeitbezugsbilder können bis zum nächsten IDR-Bild als „als Bezug benutzt“ markiert bleiben. Memory_management_control_operation-Befehle können zum Modifizieren des Bezugsbildpuffers benutzt werden.

[0087] In einer anderen Ausführungsform der vorliegenden Erfindung wird der HRD **5** mit dem Codierer **1** zum virtuellen Ausführen des Decodierens der codierten Bilder benutzt, ähnlich wie der Decodierer **3**. Der codierte Bildpuffer **1.5** kann als Vordecodierpuffer für einen HRD **5** benutzt werden. Ein Ziel des virtuellen Decodierens im HRD **5** ist das Beseitigen einer Gefahr des Decodierens von Fehlern aufgrund unausgewogener Codier-/Decodieraufgaben. Der HRD **5** enthält einen decodierten Bildpuffer **5.2** (DPB) zum temporalen Speichern von Bildern, die vom HRD **5** decodiert wurden. Außerdem sind im HRD **5** sowohl das Bezugsbildpuffer als

auch das Anzeigebildpuffern kombiniert, und sie benutzen denselben decodierten Bildpuffer **5.2**, wodurch die Speicheranforderungen des Codierers **1** reduziert sind. Alle decodierten Bilder verbleiben in dem Einheitspuffer, bis sie nicht länger als Bezugsbilder benutzt werden und bis ihre (virtuelle) Anzeigzeit erreicht ist. Der Begriff virtuell bezieht sich auf die Tatsache, dass im HRD **5** des Codierers **1**, in dem das Decodieren in dieser vorteilhaften Ausführungsform ausgeführt wird, die decodierten Bilder nicht angezeigt werden, sondern der HRD **5** nur prüft, ob Fehler im Decodierprozess des Decodierers **2** aufgrund von beispielsweise zu geringer Puffergröße vorliegen könnten. Wenn Fehler vorliegen, kann der HRD **5** den Codierer zum Ändern von Codierparametern informieren, oder die Größe des DPB **2.1** des Decodierers **1** kann erhöht werden, beispielsweise durch Senden einer SEI-(Supplemental Enhancement Information-)Nachricht an den Decodierer **1** bei einigen Implementierungen (wenn nicht die Maximalgröße des DPB bereits in Benutzung ist).

[0088] Das Decodieren kann eine gleichzeitige Funktion mit dem Codieren sein, oder der Codierer erstellt zunächst den Bitstrom, und wenn der Bitstrom bereit ist, wird er in den HRD **5** zum Prüfen eingegeben, ob der Bitstrom die HRD **5**- und Verarbeitungsebenenanforderungen erfüllt. Der HRD kann außerdem HRD-Parameter erstellen und modifizieren, wie etwa die anfängliche Pufferverzögerung im codierten Bildpuffer, gemäß den Kennzeichen des codierten Stroms.

[0089] Um den obigen Zweck zu erfüllen, soll der Neuordnungsprozess nicht von jeglicher HRD-Timinginformation abhängen, da eine derartige Information im Decodierer nicht vorhanden sein könnte. Gleichzeitig sollte es ermöglicht sein, dass der Decodierer HRD, DTS und OTS nicht akkurat Folge leistet. Beispielsweise würde es eine Anwendung wie etwa ein Streaming-Recorder mögen, Bitströme zu empfangen, sie zu decodieren und die decodierte Sequenz so schnell wie möglich zu speichern, oder sie arbeitet aufgrund der Übertragungskapazität langsamer. In diesem Falle kann HRD, DTS und OTS nicht Folge geleistet werden.

[0090] Die vorliegende Erfindung kann in zahlreichen System- und Gerätearten Anwendung finden. Das Übertragungsgerät **6** mit dem Codierer **1** und optional dem HRD **5** beinhalten außerdem einen Sender **7** zum Übertragen der codierten Bilder an den Übertragungskanal **4**. Das Empfangsgerät **8** enthält den Decodierer **2**, einen Empfänger **9** zum Empfangen der codierten Bilder und eine Anzeige **10**, auf der die decodierten Bilder angezeigt werden können. Der Übertragungskanal kann beispielsweise ein Festnetzkommunikationskanal und/oder ein drahtloser Kommunikationskanal sein. Das Übertragungsgerät und das Empfangsgerät enthalten außerdem einen oder mehrere Prozessoren **1.2**, **2.2**, die die notwendigen Schritte zum Steuern des Codier-/Decodierprozesses eines Videostroms gemäß der Erfindung ausführen können. Daher kann das Verfahren gemäß der vorliegenden Erfindung hauptsächlich als maschinenausführbare Schritte des Prozessors implementiert werden. Das Puffern der Bilder kann im Speicher **1.3**, **2.3** der Geräte implementiert sein. Der Programmcode **1.4** des Codierers kann im Speicher **1.3** gespeichert sein. Entsprechend kann der Programmcode **2.4** des Decodierers im Speicher **2.3** gespeichert sein.

Patentansprüche

1. Hypothetischer Decodierer (**5**) zum hypothetischen Decodieren eines codierten Bildstroms, wobei der hypothetische Decodierer (**5**) einen Bildpuffer (**5.2**) zum Puffern von Bezugsbildern und Bildern, die neu geordnet werden sollen, umfasst, **dadurch gekennzeichnet**, dass der hypothetische Decodierer (**5**) zum Empfangen eines Attributs von einem Codierer (**1**) konfiguriert ist, das eine Anzahl zum Puffern angeordneter Bilder definiert, welche zum Wiederherstellen der Ausgabeordnung von Bildern in dem Bildpuffer (**5.2**) genügt, wenn Bilder zum Puffern in Decodierordnung eingegeben sind, und dass der hypothetische Decodierer (**5**) folgendes umfasst:

einen Verarbeitungsblock zum Ausführen des folgenden vor dem Einfügen eines neu decodierten Bilds in den Puffer (**5.2**),

- Prüfen, ob das neu decodierte Bild in den Puffer (**5.2**) eingefügt werden kann, ohne ein anderes Bild aus dem Puffer (**5.2**) zu entfernen,
- Auswählen auf der Grundlage der Prüfung, wenn das neu decodierte Bild nicht in den Puffer (**5.2**) eingefügt werden kann, ohne ein anderes Bild aus dem Puffer (**5.2**) zu entfernen, eines Bilds, das aus dem Puffer (**5.2**) entfernt werden soll,
- Entfernen des ausgewählten Bilds aus dem Puffer (**5.2**).

2. Hypothetischer Decodierer (**5**) nach Anspruch 1, dadurch gekennzeichnet, dass er folgendes umfasst:

- ein erstes Attribut, das die Anzahl Bezugsbilder in dem Puffer (**5.2**) definiert, und
- ein zweites Attribut, das die Anzahl zum Puffern in Decodierordnung angeordneter Bilder definiert, welche zum Wiederherstellen der Ausgabeordnung von Bildern in dem Puffer (**5.2**) genügt.

3. Signal, das einen codierten Bildstrom enthält, wobei das Signal zumindest ein erstes Bezugsbild und ein zweites Bezugsbild enthält, wobei das Signal zumindest ein Signalelement zum Ermöglichen enthält, dass das erste Bezugsbild in der Decodierordnung vor dem zweiten Bezugsbild rangiert und das zweite Bezugsbild in der Ausgabeordnung vor dem ersten Bezugsbild rangiert, dadurch gekennzeichnet, dass das Signal eine Anzeige einer derartigen Anzahl zum Puffern angeordneter Bilder enthält, welche zum Wiederherstellen der Ausgabeordnung von Bildern genügt, wenn Bilder zum Puffern in Decodierordnung eingegeben sind.

4. Signal nach Anspruch 3, dadurch gekennzeichnet, dass, wenn das Signal von einem hypothetischen Decodierer (5) nach Anspruch 1 verarbeitet würde, der hypothetische Decodierer (5) keinen Fehler anzeigen würde.

5. Codierer (1) zum Ausbilden eines codierten Bildstroms, wobei der Codierer (1) zum Beinhaltenden von zumindest einem ersten Bezugsbild und einem zweiten Bezugsbild in dem Bildstrom und zumindest einem Signalelement zum Ermöglichen, dass das erste Bezugsbild in der Decodierordnung vor dem zweiten Bezugsbild rangiert und das zweite Bezugsbild in der Ausgabeordnung vor dem ersten Bezugsbild rangiert, angeordnet ist, dadurch gekennzeichnet, dass der Codierer zum Ausbilden einer Anzeige von einer derartigen Anzahl zum Puffern in einem Decodierer angeordneter Bilder zu dem Bildstrom angeordnet ist, welche zum Wiederherstellen der Ausgabeordnung von Bildern genügt, wenn Bilder zum Puffern in Decodierordnung eingegeben sind.

6. Codierer (1) nach Anspruch 5, dadurch gekennzeichnet, dass der Codierer (1) zum derartigen Ausbilden eines Signals angeordnet ist, dass, wenn das Signal von einem hypothetischen Decodierer (5) nach Anspruch 1 verarbeitet würde, der hypothetische Decodierer (5) keinen Fehler anzeigen würde.

7. Codierer (1) nach einem der Ansprüche 5 oder 6, dadurch gekennzeichnet, dass, wenn das Bild, das entfernt werden soll, noch nicht angezeigt ist, es nicht entfernt wird.

8. Codierer (1) nach einem der Ansprüche 5, 6 oder 7, dadurch gekennzeichnet, dass die Größe des Puffers definiert ist und die Anzahl Bilder, die als Bezugsbilder benutzt sind, definiert ist.

9. Codierer (1) nach Anspruch 8, dadurch gekennzeichnet, dass in jedem bestimmten Moment die Anzahl der zum Puffern in der Decodierordnung angeordneten Bilder, die zum Wiederherstellen der Ausgabeordnung von Bildern genügt, derart definiert ist, dass die Summe der Anzahl als Bezugsbilder benutzter Bilder und der Anzahl Bilder zum Puffern in der Decodierordnung, die nicht als Bezugsbilder benutzt sind, geringer als die oder gleich der Größe des Puffers (5.2; 2.1) ist.

10. Codierer (1) nach einem der Ansprüche 5 bis 9, dadurch gekennzeichnet, dass, da die Bilder als Bezugsbilder oder Nichtbezugsbilder definiert sind, die Größe des Puffers (5.2; 2.1) definiert ist, die Anzahl Bezugsbilder in dem Puffer (5.2; 2.1) definiert ist und die Anzahl der zum Puffern in der Decodierordnung angeordneten Bilder, die zum Wiederherstellen der Ausgabeordnung von Bildern in dem Puffer (5.2; 2.1) genügt, definiert ist, wobei die Anzahl Bilder, die neu geordnet werden sollen, derart definiert ist, dass die Summe der Anzahl als Bezugsbilder benutzter Bilder und der Anzahl der zum Puffern in der Decodierordnung angeordneten Bilder, die nicht als Bezugsbilder benutzt sind, geringer als die oder gleich der Größe des Puffers (5.2; 2.1) ist.

11. Codierer (1) nach einem der Ansprüche 5 bis 10, dadurch gekennzeichnet, dass, da die Bilder als Bezugsbilder oder Nichtbezugsbilder definiert sind, die Größe des Puffers (5.2; 2.1) definiert ist, die Anzahl Bezugsbilder in dem Puffer (5.2; 2.1) definiert ist und die Anzahl der zum Puffern in der Decodierordnung angeordneten Bilder, die zum Wiederherstellen der Ausgabeordnung von Bildern genügt, definiert ist, wobei die Größe des Puffers (5.2; 2.1) derart definiert ist, dass die Summe der Anzahl als Bezugsbilder benutzter Bilder und der Anzahl der zum Puffern in der Decodierordnung angeordneten Bilder, die nicht als Bezugsbilder benutzt sind, geringer als die oder gleich der Größe des Puffers (5.2; 2.1) ist.

12. Codierer (1), umfassend einen hypothetischen Decodierer (5) zum hypothetischen Decodieren eines codierten Bildstroms, wobei der Codierer (1) einen Bildpuffer (5.2) zum Puffern von Bezugsbildern und Bildern, die neu geordnet werden sollen, umfasst, dadurch gekennzeichnet, dass der hypothetische Decodierer (5) zum Empfangen eines Attributs von einem Codierer (1) konfiguriert ist, das eine Anzahl zum Puffern angeordneter Bilder definiert, welche zum Wiederherstellen der Ausgabeordnung von Bildern in dem Bildpuffer (5.2) in der Decodierordnung genügt, und dass der hypothetische Decodierer (5) folgendes umfasst: einen Verarbeitungsblock zum Ausführen des folgenden vor dem Einfügen eines neu decodierten Bilds in den Puffer (5.2),

- Prüfen, ob das neu decodierte Bild in den Puffer (5.2) eingefügt werden kann, ohne ein anderes Bild aus dem Puffer (5.2) zu entfernen,
- Auswählen auf der Grundlage der Prüfung, wenn das neu decodierte Bild nicht in den Puffer (5.2) eingefügt werden kann, ohne ein anderes Bild aus dem Puffer (5.2) zu entfernen, eines Bilds, das aus dem Puffer (5.2) entfernt werden soll,
- Entfernen des ausgewählten Bilds aus dem Puffer (5.2).

13. Codierer (1) nach Anspruch 12, dadurch gekennzeichnet, dass der Codierer zum Definieren

- eines anderen Attributs, das die Höchstanzahl Bezugsbilder in dem Puffer (5.2) definiert, konfiguriert ist.

14. Verfahren zum Ausbilden eines codierten Bildstroms, umfassend das Beinhaltens des Signals mit zumindest einem ersten Bezugsbild und einem zweiten Bezugsbild und zumindest eines Signalelements zum Ermöglichen, dass das erste Bezugsbild in der Decodierordnung vor dem zweiten Bezugsbild rangiert und das zweite Bezugsbild in der Ausgabeordnung vor dem ersten Bezugsbild rangiert, dadurch gekennzeichnet, dass das Verfahren den Schritt des

- Definierens einer Anzeige von einer Anzahl zum Puffern in einem Decodierer angeordneter Bilder, welche zum Wiederherstellen der Ausgabeordnung von Bildern genügt, wenn Bilder zum Puffern in Decodierordnung eingegeben sind, umfasst.

15. Verfahren nach Anspruch 14, dadurch gekennzeichnet, dass ein Signal derartig ausgebildet wird, dass, wenn das Signal von einem hypothetischen Decodierer (5) nach Anspruch 1 verarbeitet würde, der hypothetische Decodierer (5) keinen Fehler anzeigen würde.

16. Verfahren zum Puffern decodierter Bilder, wobei die Größe des Puffers (5.2) definiert ist, die Höchstanzahl Bezugsbilder in dem Puffer (5.2) definiert ist und ein Bildpuffer (5.2) zum Puffern von Prognosebezügen sowie decodierten Bildern zu ihrer Anordnung in der Ausgabeordnung benutzt wird, dadurch gekennzeichnet, dass eine Anzahl Bilder, die in der Decodierordnung gepuffert werden sollen, welche zum Wiederherstellen der Ausgabeordnung von Bildern genügt, von einem Codierer empfangen wird und die Größe des Bildpuffers auf der Grundlage der empfangenen Anzahl definiert wird.

17. Verfahren nach Anspruch 16, dadurch gekennzeichnet, dass ein Decodierer (2) zum Ausgeben von Bildern unter Benutzung von Information bezüglich der Anzahl der zum Puffern in der Decodierordnung angeordneten Bilder, die zum Wiederherstellen der Ausgabeordnung von Bildern genügt, angeordnet ist, und dass das Verfahren folgendes umfasst:

- Ausführen des folgenden vor dem Einfügen eines neu decodierten Bilds in den Puffer (5.2),
- Prüfen, ob das neu decodierte Bild in den Puffer (5.2) eingefügt werden kann, ohne ein anderes Bild aus dem Puffer (5.2) zu entfernen,
 - Auswählen auf der Grundlage der Prüfung, wenn das neu decodierte Bild nicht in den Puffer (5.2) eingefügt werden kann, ohne ein anderes Bild aus dem Puffer (5.2) zu entfernen, eines Bilds, das aus jenen Bildern in dem Puffer (5.2) entfernt werden soll, die nicht als Bezugsbilder benutzt werden und die bereits aus dem Puffer (5.2) ausgegeben wurden,
 - wenn ein Bild ausgewählt wurde, Entfernen des ausgewählten Bilds aus dem Puffer (5.2).

18. Decodierer (2), der Mittel (2.3) zum Puffern decodierter Bilder, wobei die Bilder als Bezugsbilder oder Nichtbezugsbilder definiert sind, die Größe des Puffers definiert ist und die Anzahl Bezugsbilder in dem Puffer definiert ist, einen Eingang zum Eingeben eines Signals beinhaltet, das zumindest ein erstes Bezugsbild und ein zweites Bezugsbild und zumindest ein Signalelement zum Ermöglichen, dass das erste Bezugsbild in der Decodierordnung vor dem zweiten Bezugsbild rangiert und das zweite Bezugsbild in der Ausgabeordnung vor dem ersten Bezugsbild rangiert, enthält, wobei das Signal ferner zumindest eine Anzeige einer derartigen Anzahl zum Puffern angeordneter Bilder enthält, welche zum Wiederherstellen der Ausgabeordnung von Bildern genügt, wenn Bilder zum Puffern in Decodierordnung eingegeben sind, und wobei das Mittel (2.3) zum Puffern einen Bildpuffer (2.1) zum Puffern von Prognosebezügen sowie decodierten Bildern zu ihrer Anordnung in der Ausgabeordnung beinhaltet, dadurch gekennzeichnet, dass der Decodierer (2) eine Definiervorrichtung zum Definieren der Größe des Bildpuffers auf der Grundlage der Anzeige umfasst.

19. Decodierer (2) nach Anspruch 18, dadurch gekennzeichnet, dass eine Anzahl der zum Puffern in der Decodierordnung angeordneten Bilder, die zum Wiederherstellen der Ausgabeordnung von Bildern genügt, de-

finiert ist.

20. Decodierer (2) nach Anspruch 19, dadurch gekennzeichnet, dass der Codierer (2) zum Ausgeben von Bildern unter Benutzung von Information bezüglich der Anzahl der zum Puffern in der Decodierordnung angeordneten Bilder, die zum Wiederherstellen der Ausgabeordnung von Bildern genügt, angeordnet ist, und dass der Decodierer (2) folgendes umfasst:

einen Verarbeitungsblock (2.2) zum Ausführen des folgenden vor dem Einfügen eines neu decodierten Bilds in den Puffer,

- Prüfen, ob das neu decodierte Bild in den Puffer (2.1) eingefügt werden kann, ohne ein anderes Bild aus dem Puffer (2.1) zu entfernen,
- Auswählen auf der Grundlage der Prüfung, wenn das neu decodierte Bild nicht in den Puffer (2.1) eingefügt werden kann, ohne ein anderes Bild aus dem Puffer (2.1) zu entfernen, eines Bilds, das aus jenen Bildern in dem Puffer (2.1) entfernt werden soll, die nicht als Bezugsbilder benutzt werden und die bereits aus dem Puffer (2.1) ausgegeben wurden,
- wenn ein Bild ausgewählt wurde, Entfernen des ausgewählten Bilds aus dem Puffer (2.1).

21. Decodierer (2) nach Anspruch 20, dadurch gekennzeichnet, dass, wenn das Bild, das entfernt werden soll, noch nicht angezeigt ist, es nicht entfernt wird.

22. Decodierer (2) nach einem der Ansprüche 18 bis 21, dadurch gekennzeichnet, dass die Größe des Puffers (2.1) definiert ist und die Anzahl Bilder, die als Bezugsbilder benutzt sind, definiert ist.

23. Decodierer (2) nach Anspruch 22, dadurch gekennzeichnet, dass in jedem bestimmten Moment die Anzahl der zum Puffern in der Decodierordnung angeordneten Bilder, die zum Wiederherstellen der Ausgabeordnung von Bildern genügt, derart definiert ist, dass die Summe der Anzahl als Bezugsbilder benutzter Bilder und der Anzahl Bilder zum Puffern in der Decodierordnung, die nicht als Bezugsbilder benutzt sind, geringer als die oder gleich der Größe des Puffers (2.1) ist.

24. Decodierer (2) nach einem der Ansprüche 18 bis 23, dadurch gekennzeichnet, dass, da die Bilder als Bezugsbilder oder Nichtbezugsbilder definiert sind, die Größe des Puffers (2.1) definiert ist, die Anzahl Bezugsbilder in dem Puffer (2.1) definiert ist und die Anzahl der zum Puffern in der Decodierordnung angeordneten Bilder, die zum Wiederherstellen der Ausgabeordnung von Bildern in dem Puffer (2.1) genügt, definiert ist, wobei, bevor ein neu decodiertes Bild in den Puffer eingefügt wird, ein Bild aus dem Puffer (2.1) entfernt wird, und das Bild, das entfernt werden soll, aus jenen Bildern in dem Puffer (2.1) ausgewählt wird, die nicht als Bezugsbilder benutzt sind und die bereits aus dem Puffer (2.1) ausgegeben wurden.

25. Decodierer (2) nach einem der Ansprüche 18 bis 24, dadurch gekennzeichnet, dass, da die Bilder als Bezugsbilder oder Nichtbezugsbilder definiert sind, die Größe des Puffers (2.1) definiert ist, die Anzahl Bezugsbilder in dem Puffer (2.1) definiert ist und die Anzahl der zum Puffern in der Decodierordnung angeordneten Bilder, die zum Wiederherstellen der Ausgabeordnung von Bildern in dem Puffer (2.1) genügt, definiert ist, wobei die Anzahl zum Puffern in der Decodierordnung angeordneter Bilder, die zum Wiederherstellen der Ausgabeordnung von Bildern in dem Puffer (2.1) genügt, derart definiert ist, dass die Summe der Anzahl als Bezugsbilder benutzter Bilder und der Anzahl der zum Puffern in der Decodierordnung angeordneten Bilder, die zum Wiederherstellen der Ausgabeordnung von Bildern genügt und die nicht als Bezugsbilder benutzt sind, geringer als die oder gleich der Größe des Puffers (2.1) ist.

26. Decodierer (2) nach einem der Ansprüche 18 bis 25, dadurch gekennzeichnet, dass, da die Bilder als Bezugsbilder oder Nichtbezugsbilder definiert sind, die Größe des Puffers (2.1) definiert ist, die Anzahl Bezugsbilder in dem Puffer (2.1) definiert ist und die Anzahl der zum Puffern in der Decodierordnung angeordneten Bilder, die zum Wiederherstellen der Ausgabeordnung von Bildern genügt, definiert ist, so dass die Summe der Anzahl als Bezugsbilder benutzter Bilder und der Anzahl der zum Puffern in der Decodierordnung angeordneten Bilder, die nicht als Bezugsbilder benutzt sind, geringer als die oder gleich der Größe des Puffers (2.1) ist.

27. Softwareprogramm, umfassend maschinenausführbare Schritte zum Ausbilden eines codierten Bildstroms und maschinenausführbare Schritte zum Einfügen von zumindest einem ersten Bezugsbild und einem zweiten Bezugsbild in den Bildstrom und zumindest ein Signalelement zum Ermöglichen, dass das erste Bezugsbild in der Decodierordnung vor dem zweiten Bezugsbild rangiert und das zweite Bezugsbild in der Ausgabeordnung vor dem ersten Bezugsbild rangiert, dadurch gekennzeichnet, dass das Softwareprogramm maschinenausführbare Schritte zum Einfügen einer Anzeige einer Anzahl zum Puffern angeordneter Bilder in den Bildstrom angeordnet ist, welche zum Wiederherstellen der Ausgabeordnung der decodierten Bilder genügt,

wenn die decodierten Bilder zum Puffern in Decodierordnung eingegeben sind.

28. Softwareprogramm, umfassend maschinenausführbare Schritte zum hypothetischen Decodieren eines codierten Bildstroms, wobei das Softwareprogramm maschinenausführbare Schritte zum Puffern von Bezugsbildern und Bildern, die in einem Bildpuffer (5.2; 2.1) neu geordnet werden sollen, umfasst, dadurch gekennzeichnet, dass das Softwareprogramm ferner folgendes umfasst:

maschinenausführbare Schritte zum Ausführen des folgenden vor dem Einfügen eines neu decodierten Bilds in den Puffer (5.2; 2.1),

- Empfangen eines Attributs, das eine Anzahl zum Puffern angeordneter Bilder definiert, welche zum Wiederherstellen der Ausgabeordnung von Bildern in dem Bildpuffer (5.2) genügt, wenn Bilder in der Decodierordnung in den Bildpuffer (5.2) eingegeben sind, von einem Codierer (1),
- Prüfen auf der Grundlage des empfangenen Attributs, ob das neu decodierte Bild in den Puffer (5.2; 2.1) eingefügt werden kann, ohne ein anderes Bild aus dem Puffer (5.2; 2.1) zu entfernen,
- Auswählen auf der Grundlage der Prüfung, wenn das neu decodierte Bild nicht in den Puffer (5.2; 2.1) eingefügt werden kann, ohne ein anderes Bild aus dem Puffer (5.2; 2.1) zu entfernen, eines Bilds, das aus dem Puffer (5.2; 2.1) entfernt werden soll,
- Entfernen des ausgewählten Bilds aus dem Puffer (5.2; 2.1).

29. Softwareprogramm, umfassend maschinenausführbare Schritte zum Puffern decodierter Bilder, wobei die Größe des Puffers definiert ist und die Höchstanzahl Bezugsbilder in dem Puffer definiert ist, und maschinenausführbare Schritte zum Verarbeiten eines Signals, das zumindest ein erstes Bezugsbild und ein zweites Bezugsbild enthält und zumindest ein Signalelement zum Ermöglichen, dass das erste Bezugsbild in der Decodierordnung vor dem zweiten Bezugsbild rangiert und das zweite Bezugsbild in der Ausgabeordnung vor dem ersten Bezugsbild rangiert, dadurch gekennzeichnet, dass das Signal ferner zumindest eine Anzeige einer derartigen Anzahl zum Puffern angeordneter Bilder enthält, welche zum Wiederherstellen der Ausgabeordnung von decodierten Bildern genügt, wenn die decodierten Bilder zum Puffern in Decodierordnung eingegeben sind, wobei das Softwareprogramm maschinenausführbare Schritte zum Puffern von Prognosebezügen sowie decodierten Bildern zu ihrer Anordnung in Anzeigeordnung in einem Bildpuffer (5.2; 2.1) und maschinenausführbare Schritte zum Definieren der Größe des Puffers (5.2; 2.1) auf der Grundlage der Anzeige umfasst.

30. Speichermedium zum Speichern eines Softwareprogramms, umfassend maschinenausführbare Schritte zum Ausbilden eines codierten Bildstroms und Beinhaltens des Signals mit zumindest einem ersten Bezugsbild und einem zweiten Bezugsbild und zumindest einem Signalelement zum Ermöglichen, dass das erste Bezugsbild in der Decodierordnung vor dem zweiten Bezugsbild rangiert und das zweite Bezugsbild in der Ausgabeordnung vor dem ersten Bezugsbild rangiert, dadurch gekennzeichnet, dass das Softwareprogramm maschinenausführbare Schritte zum Ausbilden einer Anzeige einer derartigen Anzahl zum Puffern in einem Decodierer angeordneter Bilder in dem Bitstrom umfasst, welche zum Wiederherstellen der Ausgabeordnung von Bildern genügt, wenn Bilder zum Puffern in Decodierordnung eingegeben sind.

31. Elektronisches Gerät, umfassend einen Codierer (1) zum Ausbilden eines codierten Bildstroms, wobei der Codierer (1) zum Einfügen von zumindest einem ersten Bezugsbild und einem zweiten Bezugsbild in den Bildstrom und zumindest einem Signalelement zum Ermöglichen, dass das erste Bezugsbild in der Decodierordnung vor dem zweiten Bezugsbild rangiert und das zweite Bezugsbild in der Ausgabeordnung vor dem ersten Bezugsbild rangiert, angeordnet ist, dadurch gekennzeichnet, dass der Codierer zum Einfügen einer Anzeige einer derartigen Anzahl zum Puffern in einem Decodierer angeordneter Bilder zu dem Bildstrom angeordnet ist, welche zum Wiederherstellen der Ausgabeordnung von Bildern genügt, wenn Bilder zum Puffern in Decodierordnung eingegeben sind.

32. Elektronisches Gerät (8), umfassend einen Decodierer (2), der Mittel (2.3) zum Puffern decodierter Bilder beinhaltet, wobei die Höchstanzahl Bezugsbilder in dem Puffer definiert ist, wobei das elektronische Gerät (8) einen Eingang zum Eingeben eines Signals umfasst, das zumindest ein erstes Bezugsbild und ein zweites Bezugsbild und zumindest ein Signalelement zum Ermöglichen, dass das erste Bezugsbild in der Decodierordnung vor dem zweiten Bezugsbild rangiert und das zweite Bezugsbild in der Ausgabeordnung vor dem ersten Bezugsbild rangiert, enthält, dadurch gekennzeichnet, dass das Signal ferner zumindest eine Anzeige einer derartigen Anzahl zum Puffern angeordneter Bilder enthält, welche zum Wiederherstellen der Ausgabeordnung von Bildern genügt, wenn Bilder zum Puffern in Decodierordnung eingegeben sind, und dass das Mittel (2.3) zum Puffern einen Bildpuffer (2.1) zum Puffern von Prognosebezügen sowie decodierten Bildern zu ihrer Anordnung in der Ausgabeordnung beinhaltet, dadurch gekennzeichnet, wobei der Decodierer (2) eine Definiervorrichtung (2.2; 2.4) zum Definieren der Größe des Bildpuffers (2.1) auf der Grundlage der Anzeige um-

fasst.

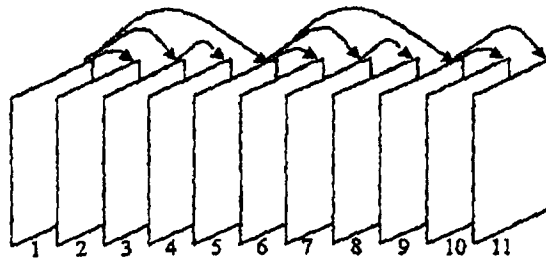
33. System, umfassend einen Codierer (1) zum Ausbilden eines Signals, das einen codierten Bildstrom enthält, einen Übertragungskanal (4) zum Übertragen des Signals an einen Decodierer (2), wobei der Decodierer (2) Mittel (2.3) zum Puffern decodierter Bilder beinhaltet, wobei das System zum Beinhalt von zumindest einem ersten Bezugsbild und einem zweiten Bezugsbild in dem Bildstrom und zumindest einem Signalelement zum Ermöglichen, dass das erste Bezugsbild in der Decodierordnung vor dem zweiten Bezugsbild rangiert und das zweite Bezugsbild in der Ausgabeordnung vor dem ersten Bezugsbild rangiert, angeordnet ist, dadurch gekennzeichnet, dass der Codierer zum Ausbilden einer Anzeige einer derartigen Anzahl zum Puffern in dem Decodierer angeordneter Bilder zu dem Bildstrom angeordnet ist, welche zum Wiederherstellen der Ausgabeordnung von Bildern genügt, wenn Bilder zum Puffern in Decodierordnung eingegeben sind.

34. Gerät (1), umfassend einen Eingang zum Eingeben von Information von zumindest einem ersten Bezugsbild und einem zweiten Bezugsbild in dem Bildstrom und zumindest einem Signalelement zum Ermöglichen, dass das erste Bezugsbild in der Decodierordnung vor dem zweiten Bezugsbild rangiert und das zweite Bezugsbild in der Ausgabeordnung vor dem ersten Bezugsbild rangiert, dadurch gekennzeichnet, dass das Gerät zum Ausbilden einer Anzeige einer Anzahl zum Puffern in der Decodierordnung angeordneter Bilder zu dem Bildstrom angeordnet ist, welche zum Wiederherstellen der Ausgabeordnung von Bildern genügt, wenn Bilder zum Puffern in Decodierordnung eingegeben sind.

Es folgen 6 Blatt Zeichnungen

Anhängende Zeichnungen

Dreischichtige rekursive temporale
Skalierbarkeit



Decodieren von zwei Schichten der obigen Dreischichtsequenz

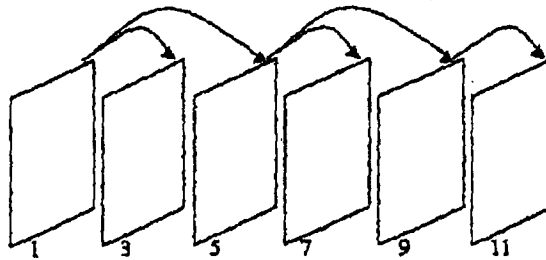
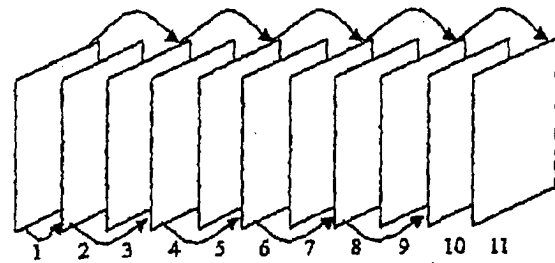
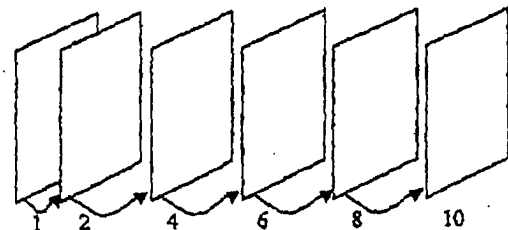


Fig. 1

Video Redundancy Decoding mit zwei Threads



Ein Thread aus der obigen Sequenz extrahiert



Ein weiterer Thread aus der obigen Sequenz extrahiert

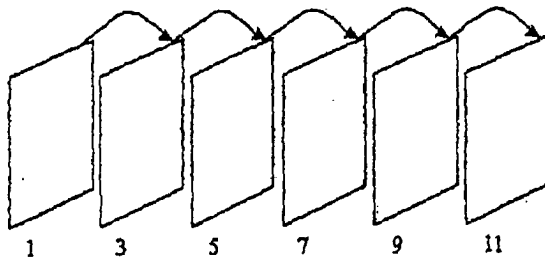


Fig. 2

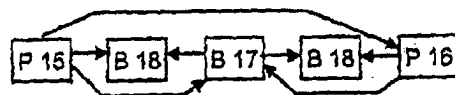


Fig. 3

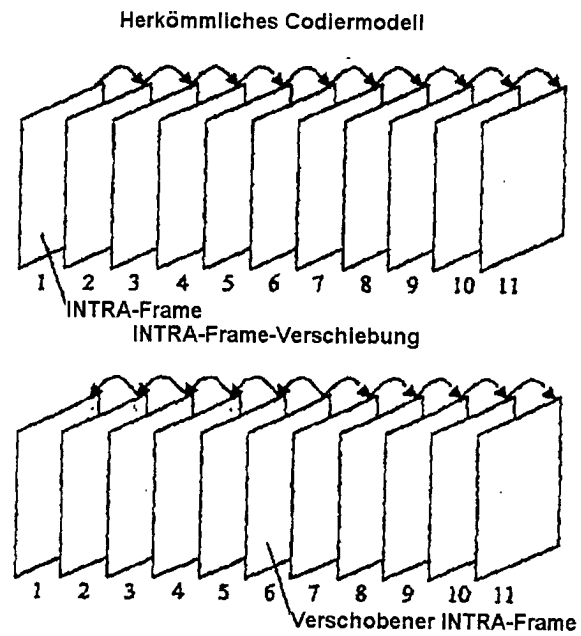


Fig. 4

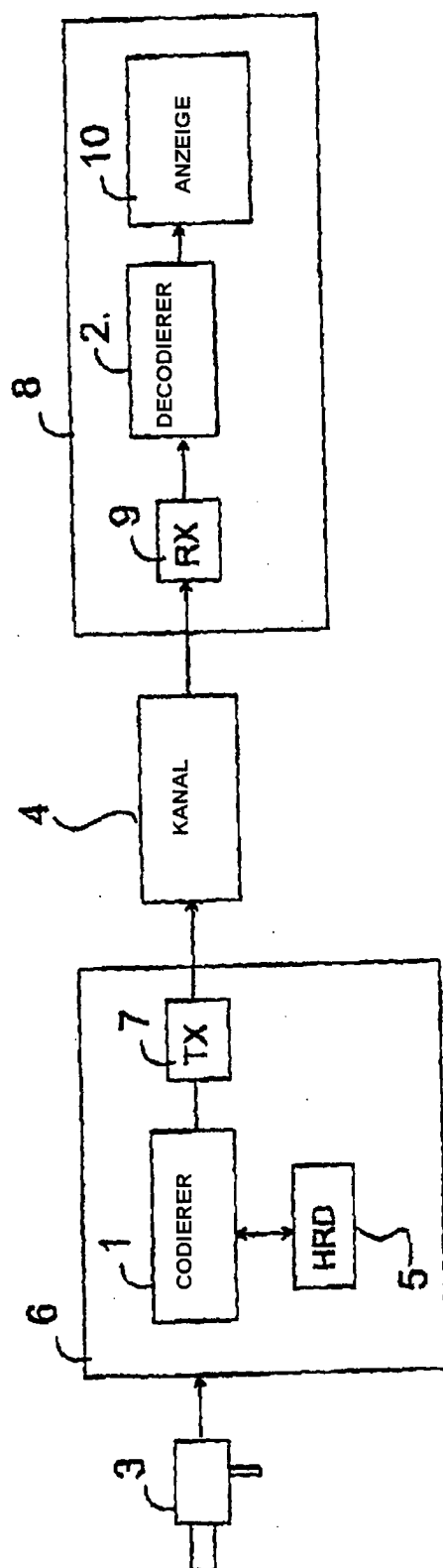


Fig. 5

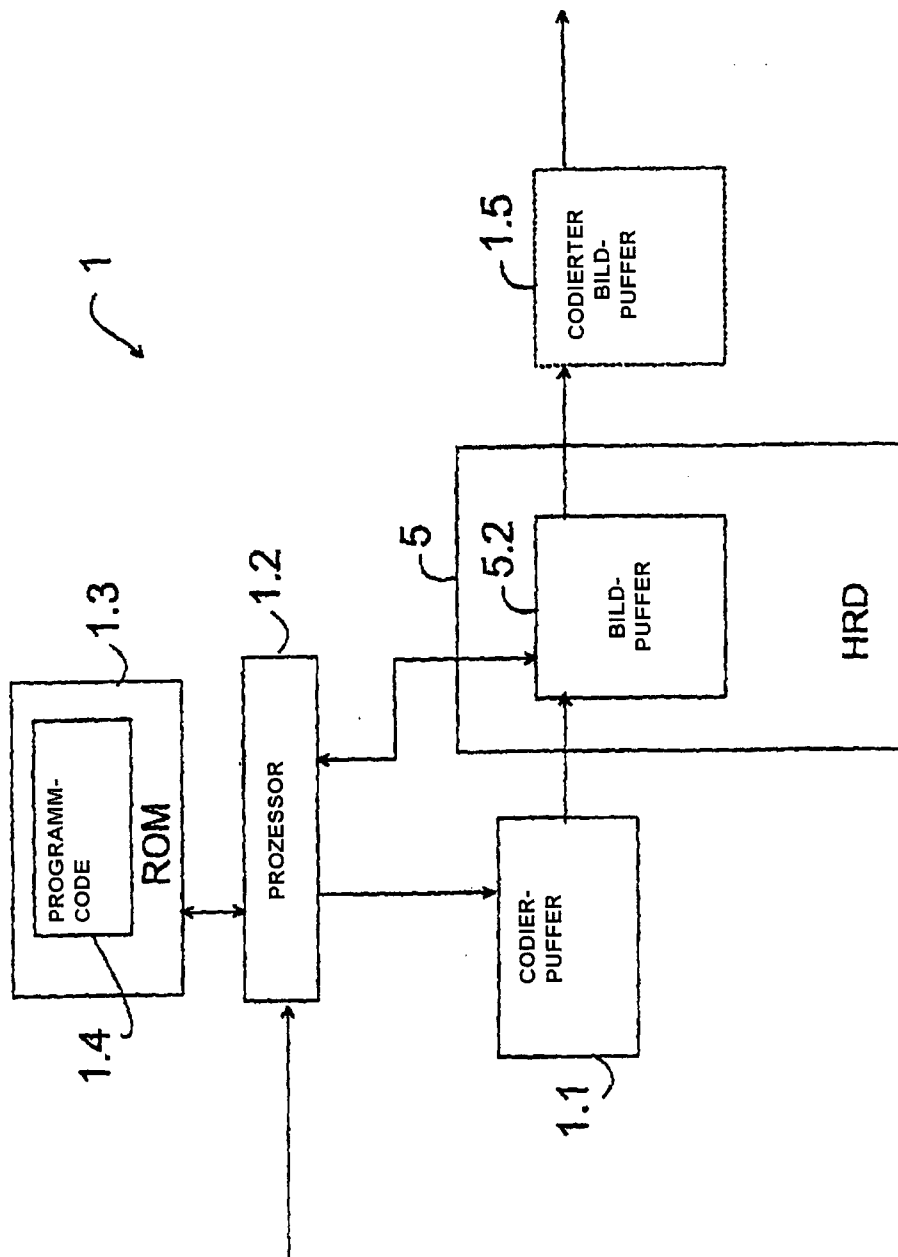


Fig. 6

2

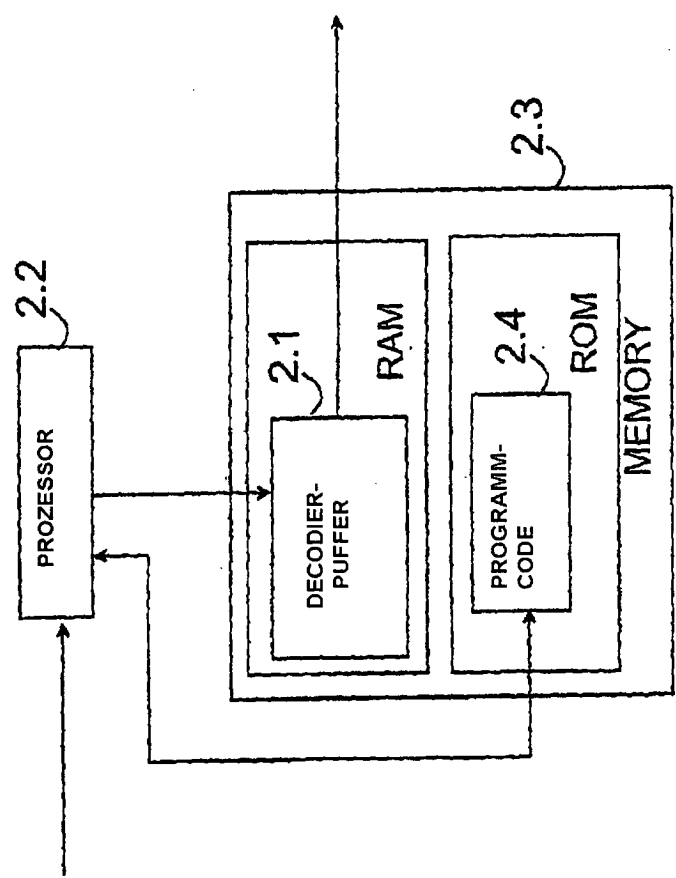


Fig. 7