(19) **Bundesrepublik Deutschland**
Deutsches Patent- und Markenamt

(10) **DE 601 14 812 T2** 2006.08.03

(12) # Übersetzung der europäischen Patentschrift

(97) **EP 1 287 704 B1**
(21) Deutsches Aktenzeichen: **601 14 812.6**
(86) PCT-Aktenzeichen: **PCT/US01/40811**
(96) Europäisches Aktenzeichen: **01 944 633.5**
(87) PCT-Veröffentlichungs-Nr.: **WO 2001/095633**
(86) PCT-Anmeldetag: **25.05.2001**
(87) Veröffentlichungstag der PCT-Anmeldung: **13.12.2001**
(97) Erstveröffentlichung durch das EPA: **05.03.2003**
(97) Veröffentlichungstag der Patenterteilung beim EPA: **09.11.2005**
(47) Veröffentlichungstag im Patentblatt: **03.08.2006**

(51) Int Cl.[8]: ***H04N 7/26*** (2006.01)

(30) Unionspriorität:
590928 **09.06.2000** **US**

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

(73) Patentinhaber:
General Instrument Corporation, Horsham, Pa., US

(72) Erfinder:
PANUSOPONE, Krit, San Diego, US; CHEN, Xuemin, San Diego, US

(74) Vertreter:
HOEGER, STELLRECHT & PARTNER Patentanwälte, 70182 Stuttgart

(54) Bezeichnung: **VIDEOFORMAT KONVERTIERUNG UND TRANSKODIERUNG VON MPEG-2 IN MPEG-4**

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

ALLGEMEINER STAND DER TECHNIK

[0001] Die vorliegende Erfindung betrifft die Komprimierung von Multimediadaten und insbesondere einen Videotranscoder, der es einem generischen MPEG-4-Decoder erlaubt, MPEG-2-Bitströme zu decodieren. Außerdem werden zeitliche und räumliche Größenumwandlung (maßstabgerechte Verkleinerung) bereitgestellt.

[0002] Der Artikel „Efficient MPEG-4/H/263 video transcoder for interoperability of heterogenous multimedia networks" von S. Dogan et al. in Electronic Letters, Mai 1999, Bd. 35, Nr. 11, S. 863 (XP-000908120) schlägt vor, einen Videotranscoder vorzusehen, um Interoperabilität zwischen den beiden Videostandards MPEG-4 und der H. 263-Bitstromsyntax zu erzielen. Es wird ein entsprechender Transcodierungsalgorithmus bereitgestellt.

[0003] Der Artikel „Efficient Processing of Compressed Video" von S. J. Wee und J. G. Apostolopoulos in „Conference Record of the 32nd Asilomar Conference on Signals, Systems & Computers", Pacific Grove, Kalifornien, 1.–4.11.1998, Asilomar Conference on Signals, Systems, and Computers, New York, New York, USA (Bd. 1, 1998, Seiten 855 bis 859, XP001032864, ISBN: 0-7803-5149-5) bespricht das Problem des Transcodierens komprimierter Videodaten aus einem Standards entsprechenden Datenstrom in einen anderen. Insbesondere zeigt der Artikel effiziente Algorithmen zum Splicen und rückwärts Abspielen von komprimierten Videoströmen.

[0004] Der Artikel „Field-to-Frame Transcoding with Spatial and Temporal Downsampling" von S. J. Wee und J. G. Apostolopoulos, Tagungsberichte 1999 International Conference on Image Processing, ICIP'99, Kobe, Japan, 24.–28.10.1999, International Conference on Image Processing, Los Alamitos, Kalifornien, USA (Bd. 4 von 4, 24.10.1999, Seiten 271 bis 275, XP000895525, ISBN: 0-7803-5468-0) bespricht einen Algorithmus zum Transcodieren von hochratigen komprimierten Bitströmen, die halbbildcodierte verschachtelte Videosignale enthalten, in niederratigere komprimierte Bitströme, die vollbildcodierte progressive Videosignale enthalten, zur Transcodierung von MPEG-2 zu H.263.

[0005] Es werden die folgenden Abkürzungen und Begriffe verwendet:

CBP – Codiertes Blockmuster
DCT – Diskrete Cosinustransformation
DTV – Digitales Fernsehen
DVD – Digitale Video-Disc
HDTV – Hochauflösendes Fernsehen
FLC – Codierung mit fester Länge
IP – Internet-Protokoll
MB – Makroblock
ME – Bewegungsschätzung
ML – Hauptebene
MP – Hauptprofil
MPS – MPEG-2-Programmstrom
MTS – MPEG-2-Transportstrom
MV – Bewegungsvektor
QP – Quantisierungsparameter
PMV – Prädiktionsbewegungsvektor
RTP – Echtzeit-Transportprotokoll (RFC 1889)
SDTV – Fernsehen mit Standardauflösung
SIF – Normzwischenformat
SVCD – Super Video Compact Disc
VCL – Codierung mit variabler Länge
VLD – Decodierung mit variabler Länge
VOP – Videoobjektebene

[0006] Der Multimedia-Codierungsstandard MPEG-4 stellt eine reichhaltige Funktionalität bereit, um verschiedene Anwendungen zu unterstützen, darunter Internet-Anwendungen wie zum Beispiel Streaming-Medien, Werbung, interaktive Spiele, virtuelle Reisen usw. Video-Streaming über das Internet (Multicast), wovon erwartet wird, dass es sich um die populärste Anwendung für das Internet handeln wird, eignet sich auch gut

für die Verwendung mit dem MPEG-4-Visual-Standard (ISOAEC 14496-2 Final Draft of International Standard (MPEG-4), „Information Technology – Generic coding of audio-visual objects, Part 2: visual", Dez. 1998).

[0007] MPEG-4 visual behandelt sowohl synthetische als auch natürliche Videosignale und berücksichtigt mehrere visuelle Objekttypen, wie zum Beispiel Video-, Face- und Mesh-Objekte. MPEG-4 visual erlaubt außerdem das Codieren eines beliebig geformten Objekts dergestalt, dass wie von einem Benutzer gewünscht mehrere Objekte in einer Szene gezeigt oder manipuliert werden können. Darüber hinaus ist MPEG-4 visual im Hinblick auf Codierungs- und Anzeigekonfigurationen sehr flexibel, indem erweiterte Merkmale enthalten sind, wie zum Beispiel mehrere Hilfs-(alpha)-Ebenen, variable Vollbildrate und geometrische Transformationen (sprites).

[0008] Der größte Teil des Videomaterials (z.B. Spielfilme, Sportereignisse, Konzerte und dergleichen) wovon erwartet wird, dass er das Ziel von Streaming-Video sein wird, wird jedoch bereits durch das MPEG-2-System komprimiert und auf Speichermedien wie etwa DVDs, Computerspeichern (z.B. Server-Festplatten) und dergleichen abgespeichert. Die MPEG-2-Systemspezifikation (ISO/IEC 13818-2 International Standard (MPEG-2), „Information Technology – Generic coding of Moving Pictures and Associated Audio: Part 2 – Video", 1995) definiert zwei System-Stream-Formate: MPEG-2 Transport Stream (MTS) und MPEG-2 Program Stream (MPS). MTS ist für die Übermittlung oder Speicherung eines oder mehrerer Programme MPEG-2-komprimierter Daten und außerdem anderer Daten in relativ fehleranfälligen Umgebungen ausgelegt. Eine typische Anwendung von MTS ist DTV. MPS ist für relativ fehlerfreie Umgebungen ausgelegt. Die populären Anwendungen sind zum Beispiel DVD und SVCD.

[0009] Versuche, dieses Problem zu behandeln, waren bisher nicht zufriedenstellend. Zum Beispiel hat das MPEG-4-Studioprofil (O. Sunohara und Y. Yagasaki, „The draft of MPEG-4 Studio Profile Amendment Working Draft 2.0", ISO/IEC JTC1/SC29/WG11 MPEG99/5135, Okt. 1999) einen Transcoder von MPEG-2 zu MPEG-4 vorgeschlagen, dieser Prozess ist jedoch nicht auf die anderen Profile von MPEG-4 Version 1 anwendbar, darunter die Natural-Visual-Profile (einfach, einfach skalierbar, Kern, Haupt, N-Bit), synthetische visuelle Profile (skalierbare Textur, einfache Face-Animation) und Synthetic/Natura) Hybrid Visual (Hybrid, einfache animierte Textur). Das Studioprofil ist nicht auf das Hauptprofil von MPEG-4 Version 1 anwendbar, da es die Syntax modifiziert, und der Decoderprozess ist nicht mit dem Rest der Profile von MPEG-4 Version 1 kompatibel.

[0010] Die MPEG-Standards designieren mehrere Mengen eingeschränkter Parameter unter Verwendung einer zweidimensionalen Einstufungsreihenfolge. Eine der Dimensionen, die als die „Profil"-Serie bezeichnet wird, spezifiziert die unterstützten Codierungsmerkmale. Die als „Ebene" bezeichnete andere Dimension spezifiziert die Bildauflösungen, Bitraten usw., die behandelt werden können.

[0011] Für MPEG-2 unterstützt das Hauptprofil auf der Hauptebene bzw. MP@ML ein Farb-Subsampling-Verhältnis von 4:2:0 und I-, P- und B-Bilder. Das einfache Profil ist dem Hauptprofil ähnlich, besitzt aber keine B-Bilder. Die Hauptebene wird für Videosignale gemäß ITU-R **601** definiert, während die einfache Ebene für SIF-Video definiert ist.

[0012] Ähnlich enthält für MPEG-4 das einfache Profil progressives SIF-Video (und besitzt keine B-VOPs oder verschachteltes Video). Das Hauptprofil erlaubt B-VOPs und verschachteltes Video.

[0013] Dementsprechend wäre es wünschenswert, Interoperabilität zwischen verschiedenen Arten von Endsystemen durch Verwendung von Videotranscodierung von MPEG-2-Video zu MPEG-4-Video und/oder Videotranscodierung von MPEG-4-Video zu MPEG-2-Video zu erzielen. Die verschiedenen Arten von Endsystemen, die berücksichtigt werden sollten, beinhalten:
TIU (Transmitting Interworking Unit): empfängt MPEG-2-Video von einem nativen MTS-(oder MPS-)System und transcodiert zu MPEG-4-Video und verteilt über Paketnetzwerke unter Verwendung einer nativen, auf RTP basierenden Systemschicht (wie zum Beispiel ein Internetzwerk auf IP-Basis). Zu Beispielen gehören ein EchtzeitCoder, eine MTS-Satellitenverbindung zum Internet und ein Videoserver mit MPS-codiertem Quellenmaterial.
RIU (Receiving Interworking Unit): empfängt MPEG-4-Video in Echtzeit von einem auf RTP basierenden Netzwerk und transcodiert dann in MPEG-2-Video (wenn möglich) und leitet zu einer nativen MTS-(oder MPS-)Umgebung weiter. Zu Beispielen gehören ein Internet-gestützter Videoserver zu einer auf MTS basierenden Kabelverteilungsanlage.
TIES (Transmitting Internet End-System): sendet MPEG-2- oder MPEG-4-Video, das im Internet-Endsystem selbst erzeugt oder gespeichert oder aus Internet-gestützten Computernetzwerken empfangen wird. Zu Beispielen gehört ein Videoserver.

RIES (Receiving Internet End-System): empfängt MPEG-2- oder MPEG-4-Video über ein auf RTP basierendes Internet zum Aufbrauch in dem Internet-Endsystem oder zur Weiterleitung zu einem traditionellen Computernetzwerk. Zu Beispielen gehört ein Desktop-PC oder eine Workstation zur Betrachtung eines Trainingsvideos.

[0014] Es wäre wünschenswert, Ähnlichkeiten und Unterschiede zwischen MPEG-2- und MPEG-4-Systemen zu bestimmen und Transcoderarchitekturen bereitzustellen, die zu einer geringen Komplexität und kleinem Fehler führen.

[0015] Die Transcoderarchitekturen sollten für Systeme bereitgestellt werden, bei denen B-Vollbilder freigegeben sind (z.B. Hauptprofil), sowie eine vereinfachte Architektur für den Fall, dass B-Vollbilder nicht benutzt werden (einfaches Profil).

[0016] Es sollte Transcodierung von Format (MPEG-2 zu MPEG-4) und/oder Größe bereitgestellt werden.

[0017] Außerdem wäre es wünschenswert, eine effiziente Abbildung von der MPEG-2- zu der MPEG-4-Syntax einschließlich einer Abbildung von Kopfteilen bereitzustellen.

[0018] Das System sollte Größentranscodierung beinhalten, einschließlich räumlicher und zeitlicher Transcodierung.

[0019] Das System sollte Größenumwandlung am Eingangsbitstom oder Ausgabebitstrom eines Transcoders erlauben.

[0020] Der Größentranscoder sollte einen Bitstrom von mit MPEG-2 MP@ML codiertem verschachtelten Video gemäß ITU-R **601** in einen MPEG-4-Bitstrom des einfachen Profils, der progressives SIF-Video, das sich z.B. für eine Streaming-Video-Anwendung eignet, enthält, umwandeln.

[0021] Das System sollte einen Ausgabebitstrom bereitstellen, der in die praktische Bandbreite für eine Streaming-Video-Anwendung (z.B. weniger als 1 Mbps) passen kann.

[0022] Die vorliegende Erfindung liefert ein System mit den obigen und weiteren Vorteilen.

KURZE DARSTELLUNG DER ERFINDUNG

[0023] Die Erfindung betrifft Formattranscodierung (MPEG-2 zu MPEG-4) und Größentranscodierung (räumlich und zeitlich).

[0024] Ein vorgeschlagener Transcoder beinhaltet Größenumwandlung, obwohl diese Parameter entweder am Eingangsbitstrom oder am Ausgabebitstrom transcodiert werden können. Es ist jedoch effizienter, alle Arten von Transcodierung in die Produktversion eines Transcoders aufzunehmen, um die Komplexität zu verringern, da sich die Transcoder untereinander Verarbeitungselemente (wie zum Beispiel einen Bitstromleser) teilen.

[0025] Die Erfindung behandelt die wichtigsten Anforderungen an einen Transcoder, z.B. Komplexität des Systems und durch den Prozess erzeugte Verluste.

[0026] Bei einer Ausführungsform verringert eine vorgeschlagene Transcoderarchitektur des Typs von vorne nach hinten die Komplexität, weil es nicht notwendig ist, Bewegungskompensation durchzuführen.

[0027] Bei einer konkreten Ausführungsform kann der Transcoder eine variable 5-Bit-QP-Repräsentation verwenden und macht AC/DC-Prädiktion und den nichtlinearen DC-Skalierer unnötig.

[0028] Die Erfindung ist alternativ dazu für die Ratensteuerung und Umbemessung nützlich.

[0029] Ein konkretes Verfahren zur Transcodierung eines vorkomprimierten Eingangsbitstroms, der in einem ersten Videocodierungsformat bereitgestellt wird, beinhaltet die folgenden Schritte: Gewinnen von Kopfinformationen aus dem Eingangsbitstrom; Bereitstellen von entsprechenden Kopfinformationen in einem zweiten, unterschiedlichen Videocodierungsformat; teilweises Dekomprimieren des Eingangsbitstroms, um teilweise dekomprimierte Daten bereitzustellen; und Neukomprimieren der teilweise dekomprimierten Daten gemäß den

Kopfinformationen in dem zweiten Format, um den Ausgabebitstrom bereitzustellen.

[0030] Ein Verfahren zur Durchführung einer maßstabgerechten 2:1-Verkleinerung an Videodaten beinhaltet die folgenden Schritte: Bilden mindestens einer Eingangsmatrix von N × N (z.B. N = 16) Koeffizienten der diskreten Cosinustransformation (DCT) aus den Videodaten durch Kombinieren von vier N/2 × N/2-Halbbildmodus-DCT-Blöcken; Durchführen eines vertikalen Downsampling und Entschachteln in die Eingangsmatrix, um zwei N/2 × N/2-Vollbildmodus-DCT-Blöcke zu erhalten; Bilden einer N × N/2-Eingangsmatrix aus den beiden Vollbildmodus-DCT-Blöcken; und Durchführen eines horizontalen Downsampling in die N × N/2-Matrix, um einen N/2 × N/2-Vollbildmodus-DCT-Block zu erhalten.

[0031] Vorzugsweise verwenden das vertikale und horizontale Downsampling jeweilige dünn besiedelte Downsampling-Matrizen. Insbesondere kann eine Vertikal-Downsampling-Matrix von 0,5 $[I_8 I_8]$ verwendet werden, wobei I_8 eine 8 × 8-Identitätsmatrix ist. Dies ist im Wesentlichen vertikale Pixel-Mittelung. Es kann eine Horizontal-Downsampling-Matrix verwendet werden, die aus ungeraden („O") und geraden („E") Matrizen zusammengesetzt ist.

[0032] Außerdem werden entsprechende Vorrichtungen vorgestellt.

KURZE BESCHREIBUNG DER ZEICHNUNGEN

[0033] Fig. 1 zeigt einen MPEG-2-Videodecoder.

[0034] Fig. 2 zeigt einen MPEG-4-Videodecoder ohne Skalierbarkeitsmerkmale.

[0035] Fig. 3 zeigt einen Transcoder des Typs von vorne nach hinten mit niedriger Komplexität (mit gesperrten B-Vollbildern) gemäß der Erfindung.

[0036] Fig. 4 zeigt eine Transcoderarchitektur, die Driftfehler (mit freigegebenen B-Vollbildern) minimiert, gemäß der Erfindung.

[0037] Fig. 5 zeigt einen Größentranscoder gemäß der Erfindung.

[0038] Fig. 6 zeigt das Downsampling von vier Halbbildmodus-DCT-Blöcken zu einem Vollbildmodus-DCT-Block gemäß der vorliegenden Erfindung.

AUSFÜHRLICHE BESCHREIBUNG DER ERFINDUNG

[0039] Die Erfindung betrifft Formattranscodierung (MPEG-2 zu MPEG-4) und Größentranscodierung (räumlich und zeitlich).

[0040] Die Erfindung stellt Bitratentranscodierung bereit, um einen vorkomprimierten Bitstrom in einen anderen komprimierten Bitstrom mit einer unterschiedlichen Bitrate umzuwandeln. Bitratentranscodierung ist z.B. für Streaming-Video-Anwendungen wichtig, weil die Netzwerkbandbreite nicht konstant ist und manchmal ein Videoserver die Bitrate reduzieren muss, um mit dem Netzwerkverkehrsbedarf fertig zu werden. Ein auf Kaskaden basierender Transcoder, der MVs aus dem Eingangsbitstrom wiederverwendet und daher Bewegungsschätzung (ME) unnötig macht, gehört zu den effizientesten der Bitratentranscoder. Der auf Kaskaden basierende Transcoder decodiert den Eingangsbitstrom, um den MV zu erhalten und das Bezugsvollbild zu bilden. Dann codiert er diese Informationen mit einem Ratensteuermechanismus, um einen Ausgabebitstrom mit der gewünschten Bitrate zu erzeugen.

[0041] Die Transcodierung der räumlichen Auflösung wird mit der Koexistenz von HDTV und SDTV in der nahen Zukunft zu einem großen Problem. Außerdem ist sie sehr nützlich für die Streaming-Video-Anwendung, da es wahrscheinlich ist, dass die Internetbandbreite für Video mit Broadcast-Qualität nicht groß genug sein wird. Das Downsampling des Bitstroms mit Broadcast-Qualität zu einem Bitstrom mit einer handhabbaren Auflösung ist daher attraktiv. Die Transcodierung der räumlichen Auflösung arbeitet gewöhnlich in der komprimierten(DCT)-Domäne, da sie die Komplexität des Systems drastisch verringert. Der Prozess des Downsampling in der komprimierten Domäne umfaßt das Verarbeiten zweier Parameter, nämlich DCT-Koeffizienten und MVs. Es wird vorgeschlagen, dass ein Downsampling-Filter und sein schneller Algorithmus das DCT-Koeffizienten-Downsampling durchführen. Es wird MV-Resampling verwendet, um den MV des downgesampelten Video zu finden. In dem realen Produkt sollte zur Vermeidung von Drift der Rest der Bewegungskompensation neu

transformiert werden, anstatt die DCT-Koeffizienten aus dem Eingangsbitstrom zu approximieren.

2. Vergleich auf hoher Ebene

[0042] In Bezug auf Struktur verwenden MPEG-2 und MPEG-4 einen ähnlichen Videokomprimierungsalgorithmus. Grundsätzlich verwenden beide Standards Bewegungsprädiktion zur Ausnutzung zeitlicher Korrelation und Quantisierung in der DCT-Domäne, um räumliche Korrelation in einem Vollbild auszunutzen. Dieser Abschnitt beschreibt die Struktur der MPEG-2- und MPEG-4-Decoder auf hoher Ebene und erwähnt dann Unterschiede zwischen den beiden Standards.

2.1 MPEG-2

[0043] Fig. 1 zeigt den vereinfachten Videodecodierungsprozess von MPEG-2. Im Decoder **100** werden codierte Videodaten einer Decodierungsfunktion **110** variabler Länge zugeführt, um die eindimensionalen Daten QFS[n] bereitzustellen, wobei n ein Koeffizientenindex im Bereich von 0–63 ist. In der Invers-Scan-Funktion **120** wird QFS[n] in ein zweidimensionales Array von Koeffizienten umgewandelt, das als QF[v][u] bezeichnet wird, wobei die Array-Indizes u und v beide im Bereich von 0 bis 7 liegen. Eine Invers-Quantisierungsfunktion **130** wendet die entsprechende Invers-Quantisierungsarithmetik an, um die letztendlichen rekonstruierten Frequenzbereichs-DCT-Koeffizienten F[v][u] zu produzieren. Eine Invers-DCT-Funktion **140** produziert die Pixel-(räumlich)Bereichswerte f[y][x]. Eine Bewegungskompensationsfunktion **150** reagiert auf einen Vollbildspeicher **160** und die Werte f[y][x], um die decodierten Pixel (pels) d[y][x] zu produzieren, wobei y und x kartesische Koordinaten im Pixelbereich sind.

[0044] MPEG-2 arbeitet auf einer Makroblockebene für Bewegungskompensation, einer Blockebene für die DCT-Transformation und der Koeffizientenebene für die Lauflängen- und verlustlose Codierung. MPEG-2 ermöglicht darüber hinaus drei Arten von Bildern, nämlich I-, P- und B-Bilder. Zulässige Bewegungsprädiktionsmoden (vorwärts, rückwärts, bidirektional) werden für die P- und B-Bilder spezifiziert. MPEG-2 verwendet verschachtelte Codierungswerkzeuge, um verschachtelte Quellen effizienter handzuhaben.

2.2 MPEG-4

[0045] Fig. 2 zeigt den MPEG-4-Videodecodierungsprozess ohne Skalierbarkeitsmerkmale.

[0046] In dem Decoder **200** werden Daten aus einem Kanal aus einem Demux **210** ausgegeben. Ein codierter Bitstrom von Formdaten wird einem Schalter **215** zusammen mit dem MPEG-4-Term video object_layer_shape (der z.B. angibt, ob das aktuelle Bild rechteckig, nur binär oder graustufig ist) zugeführt. Wenn video_object_layer_shape gleich „00" ist, ist keine Binärformdecodierung erforderlich. Andernfalls wird Binärformdecodierung ausgeführt.

[0047] Wenn Binärformdecodierung ausgeführt wird, empfängt eine Formdecodierungsfunktion **220** die zuvor rekonstruierte VOP **230** (die in einem Speicher gespeichert sein kann) und führt einer Bewegungskompensationsfunktion **240** eine formdecodierte Ausgabe zu. Die Bewegungskompensationsfunktion **240** empfängt eine Ausgabe von einer Bewegungsdecodierungsfunktion **235**, die ihrerseits einen bewegungscodierten Bitstrom aus dem Demux **210** empfängt. Die Bewegungskompensationsfunktion **240** empfängt außerdem die zuvor rekonstruierte VOP **230**, um einer VOP-Rekonstruktionsfunktion **245** eine Ausgabe zuzuführen.

[0048] Die VOP-Rekonstruktionsfunktion **245** empfängt außerdem Daten von einer Texturdecodierungsfunktion **250**, die ihrerseits einen texturcodierten Bitstrom aus dem Demux **210** empfängt, zusätzlich zu einer Ausgabe von der Formdecodierungsfunktion **220**. Die Texturdecodierungsfunktion **250** beinhaltet eine Funktion der Decodierung mit variabler Länge **255**, eine Invers-Scan-Funktion **260**, eine Invers-DC- und -AC-Prädiktionsfunktion **270**, eine Invers-Quantisierungsfunktion **280** und eine Invers-DCT-(IDCT-)Funktion **290**.

[0049] Im Vergleich zu MPEG-2 werden in MPEG-4 mehrere neue Werkzeuge verwendet, um Merkmale und Interaktivität hinzuzufügen, wie z.B. Sprite-Codierung, Formcodierung, Standbildtexturcodierung, Skalierbarkeit und Fehlernachgiebigkeit. Darüber hinaus werden Bewegungskompensations- und Texturcodierungswerkzeuge in MPEG-4, die der MPEG-2-Videocodierung ähnlich sind, modifiziert, um die Codierungseffizienz zu verbessern, z.B. Codierungswerkzeuge wie etwa Direktmodus-Bewegungskompensation, uneingeschränkte Bewegungskompensation und erweiterte Prädiktion.

[0050] Insbesondere wird Direktmodus-Bewegungskompensation für B-VOPs verwendet. Genauer gesagt

verwendet sie bidirektionale Bewegungskompensation, die durch Verwendung von I- oder P-VOP-Makroblock-MVs und Skalierung dieser abgeleitet wird, um Vorwärts- und Rückwärts-MVs für Makroblöcke in B-VOP abzuleiten. Es ist nur ein Delta-MV pro Makroblock zulässig. Der tatsächliche MV wird aus dem Delta-Vektor berechnet, und der skalierte MV aus seinem zusammenbefindlichen Makroblock.

[0051] Die uneingeschränkte Bewegungskompensation ermöglicht einen oder vier MVs pro Makroblock. Der 4-MV-Modus ist nur bei B-VOPs mit Verwendung des Direktmodus möglich. Man beachte, dass der MV für einen Chrominanz-Makroblock der Mittelwert von vier MVs aus seinem assoziierten Luminanz-Makroblock ist. Ferner ermöglicht eine uneingeschränkte Bewegungskompensation, dass ein MV aus dem Bezugsvollbild herauszeigt (die Out-Of-Bound-Textur wird aus dem Randpixel gestopft).

[0052] Erweiterte Prädiktion definiert die Prädiktionsmethode für MV- und DCT-Koeffizienten. Gemäß dem Medianwert der MVs seiner drei Nachbarn wird ein MV-Prädiktor festgelegt. Die Prädiktion von Intra-DCT-Koeffizienten folgt der Intra-AC/DC-Prädiktionsprozedur (Grahamsche Regel).

3. Transcoderarchitektur

[0053] Fig. 3 zeigt einen Transcoder des Typs von vorne nach hinten mit niedriger Komplexität gemäß der Erfindung mit gesperrten B-Vollbildern.

[0054] Ähnlichkeiten zwischen den Strukturen von MPEG-2 und MPEG-4 ermöglichen einen Transcoder mit niedriger Komplexität (front-to-back). Statt einen MPEG-2-Bitstrom völlig in die räumliche (Pixel-)Domänenebene zu decodieren, verwendet der Front-To-Back-Transcoder **300** DCT-Koeffizienten und MVs, um einen MPEG-4-Bitstrom zu erzeugen, ohne tatsächlich einen Bewegungsschätzprozess auszuführen. Ein Kompromiss besteht darin, dass diese Architektur ein Driften in dem rekonstruierten Vollbild verursachen kann und keine Bitratensteuerung erlaubt. Das Driftproblem ist jedoch gering, da der größte Teil des Unterschieds zwischen den MPEG-2- und MPEG-4-Decodern in dem Teil der verlustlosen Codierung liegt.

[0055] Der Transcoder **300** umfasst eine Kaskade aus einem MPEG-2-Bitstromleser (Decoder) (**310**–**330**) und einem MPEG-4-Kopf- und Texturcoder (Coder) (**340**–**370**), zusammen mit einer Kopfdecodierungsfunktion **304**, einer Verweistabelle **308** und einem Kommunikationsweg **312**. Der Transcoder **300** liest einen Eingangs-MPEG-2-Bitstrom, führt eine Variable-Länge-Decodierung (VLD) in einer Funktion **310** an DCT-Koeffizienten und dem MV-Rest aus und folgt dann der MPEG-2-Logik, um DCT-Koeffizienten und/oder MVs jedes Blocks in dem Vollbild zu finden.

[0056] Die Kopfdecodierfunktion **304** decodiert die MPEG-2-Kopfteile und führt diese einer Verweistabelle (oder einer analogen Funktion) **308** zu, die mit den nachfolgend aufgeführten Tabellen die entsprechenden MPEG-4-Kopfteile erhält.

[0057] Mit den Informationen der Kopfteile, den DCT-Koeffizienten und/oder MV codiert der Transcoder **300** diese Informationen in das MPEG-4-Format. Man beachte, dass das Bezugsvollbild in dieser Architektur nicht notwendig ist.

[0058] Der Transcoder **300** liest den MPEG-4-Kopfteil aus dem Eingangsbitstrom und schreibt den entsprechenden MPEG-4-Kopfteil an seine Stelle in einen Ausgabebitstrom.

[0059] Nach Verarbeitung in der VLD **310** werden die Daten einer Invers-Scan-Funktion **320** und einer Invers-Quantisierungsfunktion **330** zugeführt. Unter Verwendung der über den Weg **312** zugeführten MPEG-4-Kopfinformationen werden als nächstes die decodierten DCT-Koeffizientendaten in einem MPEG-4-Kopf- und Texturcoder verarbeitet, der eine Quantisierungsfunktion **340** und eine AC/DC-Prädiktionsfunktion **350** zum differenziellen Codieren der quantisierten DCT-Koeffizienten enthält. Insbesondere erzeugt der AC/DC-Prädiktionsprozess einen Rest von DC- und AC-DCT-Koeffizienten in einem Intra-MB durch Subtrahieren des DCT-Koeffizienten und entweder der ersten Zeile oder ersten Spalte der AC-Koeffizienten. Der Prädiktor wird adaptiv ausgewählt. Man beachte, dass die AC/DC-Prädiktionsfunktion **350** die MPEG-4-Kopfinformationen möglicherweise nicht benötigt.

[0060] Danach ergeben eine Scan-/Lauflängencodierungsfunktion **360** und eine Funktion der Codierung mit variabler Länge **370** den MPEG-4-Bitstrom.

[0061] Fig. 4 zeigt eine Transcoderarchitektur, die Driftfehler gemäß der Erfindung minimiert, wobei B-Vollbil-

der freigegeben sind.

[0062] Gleichbezifferte Elemente entsprechen in den Figuren einander.

[0063] Um den Drift-Problemen im rekonstruierten Vollbild und dem Fehlen der Bitratensteuerung entgegenzuwirken, kann man eine komplexere Architektur verwenden, wie zum Beispiel den Transcoder **400**, der eine Erweiterung des Transcoders **300** von Fig. 3 ist. Diese Architektur berechnet tatsächlich den DCT-Koeffizienten der Textur-/Restdaten, und daher ist Bewegungskompensation erforderlich. Da der Coder dieses Transcoders einen Decodierungsprozess beinhaltet, kann der Driftfehler minimiert werden.

[0064] Darüber hinaus kann man mit dem Transcoder **400** Bitströme mit B-Vollbildern transcodieren, da MPEG-4 für B-Vollbilder keinen Intra-Modus erlaubt. Der Transcoder **400** behandelt einen Block im Intra-Modus in einem B-Vollbild (bei MPEG-2) als einen Block mit Null-MV im Inter-Modus (bei MPEG-4). Er kann entweder ein Null-Rest-MV (PMV) oder Null-MV (der einen von Null verschiedenen MV-Code ergeben kann) sein, da der MV im Vergleich zum PMW prädiktiv codiert wird.

[0065] Insbesondere beinhaltet der Transcoder **400** eine Funktion der Decodierung mit variabler Länge **405**, die einem MV-Decoder **425** MV-Restdaten zuführt, und die der Invers-Scan-Funktion **320** DCT-Koeffizientendaten zuführt. Die DCT-Daten werden durch die Invers-Quantisierungsfunktion **330** und eine Invers-DCT-Funktion **420** verarbeitet, um Pixeldomänendaten zu erhalten. Intra-codierte Pixeldaten werden über einen Weg **422** einem Puffer zugeführt, während inter-codierte Pixeldaten über einen Weg **424** einem Addierer **435** zugeführt werden.

[0066] Die Pixel-(Differenz-)Daten auf dem Weg **424** werden zu Bezugspixeldaten aus einer Bewegungskompensationsfunktion **430** (die auf den MV-Decoder **425** anspricht) addiert, um dem Puffer **450** über einen Weg **448** inter-codierte Daten zuzuführen.

[0067] Zur Neucodierung, z.B. im MPEG-4-Format, gibt der Puffer **450** entweder die Intra-Pixeldaten direkt an eine DCT-Funktion **455** aus oder gibt die Inter-Pixeldaten an einen Subtrahierer **445** aus, in dem eine Differenz in Bezug auf eine Ausgabe einer Bewegungskompensationsfunktion **440** (die auf den MV-Decoder **425** anspricht) der DCT-Funktion **455** zugeführt wird.

[0068] Die DCT-Koeffizienten werden von der DCT-Funktion **455** der Quantisierungsfunktion **340** zugeführt, und die quantisierten DCT-Daten werden dann der AC/DC-(DCT-Koeffizienten-)Prädiktionsfunktion **350** zugeführt, in der AC- und DC-Reste des aktuellen MB erzeugt werden. Diese Reste von DCT-Koeffizienten werden entropiecodiert. Die Ausgangsdaten werden der Scan-/Lauflängencodierungsfunktion **360** zugeführt, und deren Ausgabe wird der Funktion **370** der Codierung mit variabler Länge zugeführt, um den MPEG-4-gerechten Bitstrom zu erhalten.

[0069] Die quantisierten DCT-Koeffizienten werden außerdem von der Quantisierungsfunktion **340** an eine Invers-Quantisierungsfunktion **495** ausgegeben, deren Ausgabe einer Invers-DCT-Funktion **490** zugeführt wird, deren Ausgabe in einem Addierer **485** mit der Ausgabe der Bewegungskompensationsfunktion **440** summiert wird. Die Ausgabe des Addierers **485** wird einem Puffer **480** und danach der Bewegungskompensationsfunktion **440** zugeführt.

[0070] Die Kopfdecodierungsfunktion **304** und die Verweistabelle **308** und der Weg **312** arbeiten wie in Verbindung mit Fig. 3 besprochen, um die Neucodierung in das MPEG-4-Format in den Funktionen **340**–**370** zu steuern.

4. Implementierung des Formattranscoders

[0071] Dieser Abschnitt erläutert die Implementierung der Formattranscodierung, z.B. wie in den oben besprochenen Fig. 3 und Fig. 4 und in der später zu besprechenden Fig. 5 implementiert. Geringfügige Implementierungsdetails (z.B. systembezogene Einzelheiten, wie die Verwendung von Zeitstempeln und dergleichen), die nicht spezifisch besprochen werden, sollten Fachleuten offensichtlich sein.

[0072] Bei einer konkreten Implementierung kann mann mit den Transcodern der vorliegenden Erfindung einen MPEG-2-Bitstrom des Hauptprofils und der Hauptebene (MP@ML) in einen MPEG-4-Bitstrom des Hauptprofils umwandeln. Es wird angenommen, dass der MPEG-2-Bitstrom in Vollbildstruktur mit B-Bild-Codierung (keine Dual-Prime-Prädiktion) codiert wird. Im Allgemeinen sollte derselbe Codierungsmodus behalten wer-

den, der bei der MPEG-2-Codierung verwendet wird. Dieser Modus ist wahrscheinlich bei MPEG-4 optimal und vermeidet daher die Komplexität des Modusentscheidungsprozesses. Das Tranzparenzmuster bei MPEG-4 ist immer 1 (ein rechteckiges Objekt mit derselben Größe der VOP in einer VOP). Das heißt, MPEG-4 erlaubt ein beliebig geformtes Objekt, das durch ein von Null verschiedenes Transparenzmuster definiert wird. Dieses Merkmal existiert in MPEG-2 nicht, so dass man sicher annehmen kann, dass alle Transparenzmuster des Transcodierungsobjekts Eins sind.

4.1 Der MPEG-2-Bitstromleser

[0073] Ein Transcoder gemäß der Erfindung erhält den Bitstromkopf, DCT-Koeffizienten und MVs aus dem MPEG-2-Bitstrom. Diese Informationen werden in dem Bitstrom miteinander gemischt. Sowohl MPEG-2- als auch MPEG-4-Bitströme verwenden eine hierarchische Struktur, die aus mehreren Schichten besteht. Jede Schicht beginnt mit dem Kopfteil, gefolgt durch ein Vielfaches ihrer Subschicht. Bei dieser Implementierung (siehe Tabelle 1) besitzt die MPEG-2-Schicht eine direkte Übersetzung in die MPEG-4-Schicht, mit der Ausnahme der Slice-Schicht in MPEG-2, die in MPEG-4 nicht benutzt wird. DC-Koeffizienten und vorhergesagte MVs in MPEG-4 werden in den Blöcken, die das Slice starten, zurückgesetzt.

[0074] Bestimmte MPEG-4-Kopfteile sind jedoch von MPEG-2-Kopfteilen verschieden und umgekehrt. Zum Glück reichen die Beschränkungen in MPEG-2- und MPEG-2-Kopfinformationen aus, um einen MPEG-4-Kopfteil zu spezifizieren. Die Tabellen 2 bis 6 listen MPEG-4-Kopfteile und ihre Beziehung mit einem MPEG-2-Kopfteil bzw. Einschränkung auf jeder Schicht auf.

Tabelle 1. Beziehung zwischen MPEG-2- und MPEG-4-Schichten

MPEG-2	MPEG-4
Videosequenz	Videoobjektsequenz (VOS)/
	Videoobjekt (VO)
Sequenz-skalierbare Erweiterung	Videoobjektschicht (VOL)
Gruppe von Bild (GOP)	Gruppe-von-Video-Objekt-Ebene (GOV)
Bild	Videoobjektebene (VOP)
Makroblock	Makroblock

Tabelle 2. MPEG-4-Kopfteil und seine Ableitung (VOS und VO)

Kopfteil	Code	Kommentar
Visuelles_objekt_sequenz_start_code	00001B0	Einleiten einer visuellen Sitzung
Profil_und_ebenen_anzeige	00110100	Hauptprofil/ Ebene 4
Visuelles_objekt_sequenz_end_code	00001B1	Eine visuelle Sitzung beenden
Visuelles_objekt_start_code	00001B5	Ein visuelles Objekt einleiten
Ist_visuelles_objekt_kennung	0	Es muss keine Versions-identifikation mit Priorität spezifiziert werden
Visuelles_objekt_typ	0001	Video-ID
Video_objekt_start_code	0000010X-0000011X	Ein neues Video-Objekt markieren
Video_signal_typ	Abgeleitet aus MPEG-2	Entspricht Sequenz_display_erweiterung_id von MPEG-2
Video_format	Genauso wie MPEG-2	Entspricht Sequenz_display_erweiterung_id von MPEG-2
Video_bereich	Abgeleitet aus MPEG-2	Entspricht Sequenz_display_erweiterung_id von MPEG-2
Farb_beschreibung	Genauso wie MPEG-2	Entspricht Sequenz_display_

		erweiterung_id von MPEG-2
Farb_primäre	Genauso wie MPEG-2	Entspricht Farb_ beschreibung von MPEG-2
Transfer_eigenschaften	Genauso wie MPEG-2	Entspricht Farb_ beschreibung von MPEG-2
Matrix_koeffizienten	Genauso wie MPEG-2	Entspricht Farb_ beschreibung von MPEG-2

Tabelle 3. MPEG-4-Kopfteil und seine Ableitung (VOL)

Kopfteil	Code	Kommentar
Video_objekt_schicht_start_code	0000012X	Eine neue Videoobjekt-schicht markieren
Direkt_zugänglich_vol	0	Erlaubt nicht-intracodierte VOP
Video_objekt_typ_identifikation	00000100	Hauptobjekttyp
Ist_objekt_typ_kennung	0	Keine Versionsiden-tifikation mit Priorität muss spezifiziert werden
Seiten_verhältnis_info	Genauso wie MPEG-2	Entspricht Seiten_ verhältnis_ informationen von MPEG-2
Paar_breite	Genauso wie MPEG-2	Entspricht Vertikaler_ größe von MPEG-2
Paar_höhe	Genauso wie MPEG-2	Entspricht Horizontaler_ größe von MPEG-2
Vol_steuer_parameter	Genauso wie MPEG-2	Entspricht Erweiterungs_ start_code_

		kennung (Sequenz-erweiterung) von MPEG-2
Chroma_format	Genauso wie MPEG-2	Entspricht Chroma_format von MPEG-2
Niedrige_verzögerung	Genauso wie MPEG-2	Entspricht Niedrige_ verzögerung von MPEG-2
Vbv_parameter	Neuberechnet	Folgt der MPEG-4-VBV-spez.
Video_objekt_schicht_form	00	Rechteckig
Vop_zeit_inkrement_auflösung	Neuberechnet	Siehe Tabelle 7
Feste_vop_rate	1	Gibt an, dass alle VOPs mit fester Rate codiert werden
Festes_vop_zeit_inkrement	Neuberechnet	Siehe Tabelle 7
Video_objekt_schicht_breite	Genauso wie MPEG-2	Entspricht Display_vertikale_ größe
Video_objekt_schicht_höhe	Genauso wie MPEG-2	Entspricht Display_ horizontale_größe
Verschachtelt	Genauso wie MPEG-2	Entspricht Progressive_ Sequenz
Obmc_sperrung	1	OBMC sperren
Sprite_freigabe	0	Zeigt Fehlen von Sprite an
Nicht_8_bit	Abgeleitet aus MPEG-2	Entspricht Intra_dc_präzision von MPEG-2
Quant_typ	1	MPEG-Quantisierung
Komplexität_schätzung_sperrung	1	Komplexitäts-Schätzungs-Kopfteil sperren
Resync_markierung_sperrung	1	Zeigt fehlen von

		Resync_ markierung an
Daten_partitioniert	0	Sperrung der Datenparti-tionierung
Reversibel_vlc	0	Sperrung von reversiblen vlc
Skalierbarkeit	0	Gibt an, dass die aktuelle Schicht als Basisschicht verwendet wird

Tabelle 4. MPEG-4-Kopfteil und seine Ableitung (VOP)

Kopfteil	Code	Kommentar
Vop_start_code	000001B6	Einen Start einer Videoobjektebene markieren
Vop_codierungs_typ	Genauso wie MPEG-2	Entspricht Bild_ codierungs_typ von MPEG-2
Modulo_zeit_basis	Regeneriert	Folgt MPEG-4-spez.
Vop_zeit_inkrement	Regeneriert	Folgt MPEG-4-spez.
Vop_codiert	1	Zeigt an, dass nachfolgende Daten für die VOP existieren
Vop_rundungs_typ	0	Setze Wert von Rundungs_steuerung auf „0“
Ändern_umwandlungs_verhältnis_sperrung	1	Annahme, dass Umw_verhältnis für alle Makroblöcke „1“ ist
Vop_konstant_alpha	0	Nicht Vop_konstant_ alpha_Wert in den Bitstrom aufnehmen
Intra_dc_vlc_thr	0	Für gesamte VOP Intra-DC-vlc verwenden
Oberes_halbbild_zuerst	Genauso wie	Entspricht Oberes_

	MPEG-2	halbbild_zuerst von MPEG-2
Abwechselndes_vertikal_scan_flag	Genauso wie MPEG-2	Entspricht Abwechselndes_scan von MPEG-2
Vop_quant	Abgeleitet aus MPEG-2	Entspricht Quantisierer_ maßstabs_code von MPEG-2
Vop_fcode_vorwärts	Genauso wie MPEG-2	Siehe Abschnitt 4.3
Vop_fcode_rückwärts	Genauso wie MPEG-2	Siehe Abschnitt 4.3

Tabelle 5. MPEG-4-Kopfteil und seine Ableitung (Makroblock und MV)

Kopfteil	Code	Kommentar
Nicht_codiert	Abgeleitet aus MPEG-2	Entspricht Makroblock_adressen_ inkrement von MPEG-2
Mcbpc	Abgeleitet aus MPEG-2	Entspricht Makroblock_typ von MPEG-2
Ac_präd_flag	0	Intra-AC-Prädiktion sperren
Cbpy	Abgeleitet aus MPEG-2	Siehe Abschnitt 4.2
Dquant	Abgeleitet aus MPEG-2	Siehe Abschnitt 4.2
Modb	Abgeleitet aus MPEG-2	Entspricht Makroblock_typ
Mb_typ	Abgeleitet aus MPEG-2	Entspricht Makroblock_typ
Cbpb	Abgeleitet aus MPEG-2	Siehe Abschnitt 4.2
Dbquant	Abgeleitet aus MPEG-2	Siehe Abschnitt 4.2
Horizontal_mv_daten	Abgeleitet aus MPEG-2	Entspricht Bewegungs_code[r][s][0] von MPEG-2

Vertikal_mv_daten	Abgeleitet aus MPEG-2	Entspricht Bewegungs_code[r][s][1] von MPEG-2
Horizontal_mv_rest	Abgeleitet aus MPEG-2	Entspricht Bewegungs_rest[r][s][0] von MPEG-2
Vertikal_mv_rest	Abgeleitet aus MPEG-2	Entspricht Bewegungs_rest[r][s][1] von MPEG-2

Tabelle 6. MPEG-4-Kopfteil und seine Ableitung (Block und verschachtelte Informationen)

Kopfteil	Code	Kommentar
Dct_dc_größe_luminanz	Genauso wie MPEG-2	Entspricht dct_dc_ größe_luminanz von MPEG-2
Dct_dc_differenzial	Genauso wie MPEG-2	Entspricht dct_dc_differenzial von MPEG-2
Dct_dc_größe_chrominanz	Genauso wie MPEG-2	Entspricht dct_dc_ größe_chrominanz von MPEG-2
Dct_koeffizient	Abgeleitet aus MPEG-2	Siehe Abschnitt 4.2
Dct_typ	Genauso wie MPEG-2	Entspricht DCT_typ von MPEG-2
Halbbild_prädiktion	Genauso wie MPEG-2	Entspricht Vollbild_ bewegungs_typ von MPEG-2
Vorwärts_oberes_halbbild_bezug	Genauso wie MPEG-2	Entspricht Bewegungs_vertikal_ halbbild_auswahl[0][0] von MPEG-2
Vorwärts_unteres_halbbild_bezug	Genauso wie MPEG-2	Entspricht Bewegungs_vertikal_ halbbild_auswahl[1][0] von MPEG-2
Rückwärts_oberes_halbbild_bezug	Genauso wie MPEG-2	Entspricht Bewegungs_vertikal_ halbbild_auswahl[0][1] von MPEG-2
Rückwärts_unteres_halbbild_bezug	Genauso wie MPEG-2	Entspricht Bewegungs_vertikal_ halbbild_auswahl[1][1] von MPEG-2

Tabelle 7. Abbildung des Vollbild_raten_code in MPEG-2 auf vop_zeit_inkrement_auflösung und Festes_vop_zeit_inkrement in MPEG-4.

Vollbild_raten_code	Vop_zeit_inkrement_ auflösung	Festes_vop_zeit_ inkrement
0001	24.000	1001
0010	24	1
0011	25	1
0100	30.000	1001
0101	30	1
0110	50	1
0111	60.000	1001
1000	60	1

[0075] MV-Daten werden in der Makroblockschicht gespeichert. Für jeden Makroblock sind bis zu vier MVs möglich. Darüber hinaus kann ein MV entweder den Halbbild- oder den Vollbildtyp aufweisen und kann entweder Vollpixel- oder Halbpixelauflösung aufweisen. Der MPEG-2-MV-Decodierungsprozess wird verwendet, um Bewegungs_code (VLC) und Bewegungs_rest (FLC) und daher Delta zu bestimmen. Kombiniert mit dem prädiktiven MV ergibt delta den Halbbild-/Vollbild-MV. Der MV für übersprungene Makroblöcke wird auf Null gesetzt.

[0076] DCT-Daten werden in der Blockschicht gespeichert. Sie werden zuerst aus dem Bitstrom (VLC) decodiert, unter Verwendung entweder des Zickzack- oder Wechsel-Scan-Musters invers gescannt und dann invers quantisiert. Der Intra-DC-Koeffizient wird aus dct_dc-differenzial und dem Prädiktor bestimmt (der Prädiktor wird gemäß der MPEG-2-spez. zurückgesetzt). dCT-Koeffizienten in einem übersprungenen Makroblock werden auf Null gesetzt.

4.2 Texturcodierung

[0077] DCT-Koeffizienten (für Inter-Vollbilder) werden von einem Transcoder gemäß der Erfindung wiederverwendet. Es sollten die folgenden Richtlinien verwendet werden:

1. q_maßstab_typ = 1 (linearer Maßstab) wird bei MPEG-2-Quantisierung verwendet.
2. Die MPEG-Quantisierungsmethode sollte bei der MPEG-4-Quantisierung ausschließlich verwendet werden (nicht H.263), um eine Nichtübereinstimmung zwischen einem rekonstruierten MPEG-2- und MPEG-4-Vollbild (Drift) zu verringern.
3. Ein Differenzialwert von MPEG-2-QP bestimmt dquant bei MPEG-4. dquant wird immer dann auf ±2 gesetzt, wenn der Differenzialwert größer als ±2 ist. dquant ist ein 2-Bit-Code, der eine Änderung in dem Quantisierer (quant) für I- und P-VOPs spezifiziert.
4. Die Quantisierungsmatrix sollte im Anschluss an die Änderung der Matrix in dem MPEG-2-Bitstrom verändert werden.
5. Der Transcoder besitzt die Flexibilität, eine alternative Vertikal-Scan-Methode (für verschachtelte Sequenz) auf der VOL-Ebene freizugeben.
6. Intra-AC/DC-Prädiktion (wobei Skalierung erfolgt, wenn der QP des aktuellen Blocks nicht mit dem des vorhergesagten Blocks übereinstimmt) sollte auf einer Makroblockebene ausgeschaltet werden, um Komplexität und Nichtübereinstimmung bei der AC-Quantisierung zu verringern.
7. Höhere Effizienz lässt sich durch Verwendung von Intra_dc_vlc_thr erreichen, um die richtige VLC-Tabelle (AC/DC) für die Codierung von Intra-DC-Koeffizienten auszuwählen, z.B. als Funktion des Quantisierungsparameters (außer wenn Intra_dc_vlc_thr entweder 0 oder 7 ist – diese Schwellen erzwingen die Verwendung der Intra-DC- oder -AC-Tabelle ungeachtet des QP.
8. Ein übersprungener Makroblock wird als Makroblock des Typs nicht_codiert codiert (alle DCT-Koeffizienten sind 0).

[0078] 9. Cbpy und cbpc (CBP) werden gemäß Code_block_muster **420** (CBP_**420**) festgesetzt. Man beachte, dass eine leichte Diskrepanz zwischen CBP bei MPEG-4 und CBP_**420** bei MPEG-2 für einen Intra-Makroblock besteht. Genauer gesagt zeigt CBP_**420**, wenn es gesetzt ist, an, dass mindestens einer der DCT-Koeffizienten in diesem Block nicht null ist. CBP enthält ähnliche Informationen, mit der Ausnahme, dass es nicht einem DC-Koeffizienten in einem Intra-Makroblock entspricht (auch abhängig von intra_dc_vlc_thr). Daher ist

es möglich, dass CBP nicht null ist wenn CBP_**420** in einem Intra-Makroblock null ist (dieser Fall kann in einer I-VOP und P-VOP, nicht aber einer B-VOP vorkommen).

[0079] Es gibt drei Quellen von Verlusten bei der Texturcodierung, nämlich QP-Codierung, DC-Prädiktion und nichtlinearer Skalierer für DC-Quantisierung. MPEG-4 verwendet Differenzialcodierung zur Codierung eines QP. MPEG-2 erlaubt alle möglichen 32 QP-Werte auf Kosten von 5 Bit. Der Differenzialwert kann jedoch ±2 (in QP-Wert-Einheiten) aufbrauchen und daher ist ein Differenzialwert von mehr als ±2 Verlust. Dieser Verlust kann durch Begrenzung der QP-Fluktuation zwischen dem Makroblock in dem MPEG-2-Ratensteueralgorithmus minimiert werden. Alle Intra-Makroblöcke führen adaptive DC-Prädiktion durch, die dem vorherigen Makroblock eine unterschiedliche Prädiktion entnehmen kann (MPEG-2-DC-Prädiktion), wodurch ein unterschiedlicher DC-Rest für die Quantisierung verursacht wird. DC-Koeffizienten aller Intra-Makroblöcke bei MPEG-4 werden auch wegen des nichtlinearen Skalierers auf von MPEG-2 unterschiedliche Weise quantisiert. Deshalb sind quantisierte DC-Koeffizienten für MPEG-2- und MPEG-4-Codierung wahrscheinlich für einen Intra-Makroblock unterschiedlich.

4.3 MV-Codierung

[0080] Der Transcoder codiert MVs zu einem MPEG-4-Format. Bei der Transcodierung eines MV von MPEG-2 zu MPEG-4 tritt jedoch kein Fehler auf, weil MV-Codierung ein verlustloser Prozess ist. Dem MPEG-4-Coder werden die folgenden Beschränkungen auferlegt:

1. Der Modus der uneingeschränkten Bewegungskompensation ist gesperrt, das heißt, kein MV zeigt aus der Grenze des Vollbildes heraus.
2. Es wird der erweiterte Prädiktionsmodus verwendet. In einem MPEG-4-Bitstrom wird ein unterschiedlicher Prädiktor (ein Medianwert) verwendet, ein MV für einen 8 × 8-Pell-Block aber nicht. Das heißt, erweiterter Prädiktionsmodus erlaubt 8 × 8-MV- und nichtlinearen (Median-Filter-)Prädiktor. In dem erfindungsgemäßen Formattranscoder wird nur ein nichtlinearer Prädiktor verwendet (es wird weiter ein 16 × 16-MV behalten).
3. Der Direktmodus wird in einem MPEG-4-Bitstrom nicht erlaubt, das heißt, dass es nur vier MV-Typen für eine B-VOP gibt, d.h. 16 × 16-Vorwärts- und -Rückwärtsvektoren und 16 × 8-Vorwärts- und -Rückwärts-Halbbildvektoren.
4. Halbbildbewegungskompensation wird immer dann angewandt, wenn ein 16 × 8-Halbbildvektor benutzt wird (Modus aufrechterhalten).
5. Ein übersprungener Makroblock wird als ein Makroblock des Typs nicht_codiert codiert (Bewegungskompensation mit null-MV).
6. Bei MPEG-4 ist Einzel-f_code erlaubt. Deshalb wird der größere f_code in MPEG-2 zwischen den beiden Richtungen (vertikal, horizontal) in f_code in MPEG-4 auf der Basis der folgenden Beziehung umgewandelt: f_code (MPEG-4) = f_code (MPEG-2)-1.
7. Es wird kein Stopfprozess verwendet, da die Textur für das gesamte Bezugsvollbild bekannt ist.
8. Halbbildbewegungskompensation wird immer dann verwendet, wenn Dual-Prime-Arithmetik aktiviert ist. Vektorparitäten (Halbbild des Bezugs- und Halbbild des vorhersagenden Vollbildes) werden erhalten. Halbbild-MVs werden gemäß Vektor [0][0][1:0] erzeugt, der in dem MPEG-2-Bitstrom codiert ist. Wenn Prädiktion derselben Parität verwendet wird (z.B. oberes Halbbild zu oberes Halbbild oder unteres Halbbild zu unteres Halbbild), sind beide Halbbild-MVs Vektor [0][0][1:0]. Wenn Prädiktion der ungeraden Parität verwendet wird (z.B. oberes Halbbild zu unteres Halbbild oder unteres Halbbild zu oberes Halbbild) verwendet der MV des oberen Halbbildes Vektor [2][0][1:0] und der MV des unteren Halbbildes verwendet Vektor [3][0][1:0]. Vektor [r][0][0:1] für r = 2,3 kann folgendermaßen berechnet werden:
   (a)

   Vektor[r][0][0] = (vektor[0][0][0] ×
   m[parity_ref)[parity_pred]//2) + dmvektor[0].

   (b)

   Vektor[r][0][0] = (vektor[0][0][1] ×
   m[parity_ref][parity_pred]//2) +
   e[parity_ref][parity_pred] + dmvektor[1].

[0081] Man beachte, dass (m[parity_ref][parity_pred] und e[parity_ref][parity_pred] in Tabelle 7–11 bzw. 7–12 in der MPEG-2-Spezifikation (ISO/IEC 13818-2) definiert sind.

[0082] Darüber hinaus bedeutet „r" die Ordnung des MV, z.B. erster, zweiter usw. r = 0 bedeutet die erste Menge von MV und r = 1 bedeutet die zweite Menge von MV. Die Dual-Prime-Prädiktion verwendet r = 2 und r = 3, um zwei zusätzliche Mengen von MVs zu identifizieren.

[0083] „//" bedeutet ganzzahlige Division mit Rundung auf die nächste ganze Zahl.

4.4 Codierung von Intra-MB in B-VOP

[0084] Beim Codieren eines Intra-MB in einem B-Vollbild eines MPEG-2-Bitstroms (z.B. wie in Fig. 4 gezeigt) ist eine zusätzliche Umwandlung notwendig. MPEG-4 ersetzt den Intra-Modus für B-VOP mit Direktmodus und daher muss ein Intra-MB in B-Vollbild in der MPEG-4-Syntax unterschiedlich codiert werden. Es gibt zwei praktische Lösungen für dieses Problem.

[0085] Die erste Lösung verwendet die Architektur, die dem Front-to-Back-Transcoder von Fig. 3 ähnlich ist (kein Puffer für das gesamte Bezugsvollbild). MC wird im Vergleich zu dem vorherigen MB (oder vorherigem MB ohne Kompensation des Texturrestes mit den Kosten zusätzlichen Speichers mit der Größe eines MB) in derselben VOP unter der Annahme durchgeführt, dass dieser MB seinem Bezugs-MB (seiner unkomprimierten Version) nahe genug kommt. Der MV für den Intra-MB ist gleich dem MV des vorherigen MB, verschoben um seine MB-Distanz.

[0086] Die zweite Lösung verwendet die Architektur, die der in Fig. 4 gezeigten ähnlich ist. Sie behält das Bezugsvollbild für alle I- und P-VOPs. Man beachte, dass bei dieser Lösung MC an allen P-VOPs durchgeführt werden muss. Der MV für den Intra-MB ist derselbe wie der vorhergesagte MV (Median seiner drei Nachbarn) und MC wird im Vergleich zu dem Bezugs-MB durchgeführt, den der abgeleitete MV zeigt.

5. Maßstabgerechte Video-Verkleinerung in der komprimierten Domäne

[0087] Im Allgemeinen bedeutet maßstabgerechte Video-Verkleinerung und Größentranscodierung dasselbe. Downsampling bedeutet Unterabtastung mit einem Anti-Alias-Filter (Tiefpass), aber Unterabtastung und Downsampling werden hier synonym verwendet.

[0088] Größentranscodierung wird rechnerisch intensiv, wenn ihre Eingabe und Ausgabe in der komprimierten Domäne liegen. Ein Prozess der maßstabgerechten Video-Verkleinerung, der seine Operationen in der komprimierten Domäne beschränkt (und effektiv Decodierungs- und Codierungsprozesse vermeidet) ergibt eine wesentlich verringerte Komplexität. Es entstehen jedoch zwei neue Probleme bei der maßstabgerechten Verkleinerung in der komprimierten Domäne, nämlich. Downsampling von DCT-Koeffizienten und MV-Daten.

[0089] In letzter Zeit wurden Algorithmen für die maßstabgerechte Video-Verkleinerung in der komprimierten Domäne besprochen, sie behandeln jedoch nicht die vollständige Transcodierung zwischen MPEG-2 und MPEG-4, wozu Entschachtelung von Halbbild zu Vollbild gehört. Die vorliegende Erfindung behandelt dieses Problem.

[0090] Die Unterabschnitte 5.1 und 5.2 geben Lösungen für zwei neue Probleme in dem Downsampling-Prozess an. Die Implementierung eines vorgeschlagenen Größentranscoders gemäß der Erfindung wird im Abschnitt 6 und in Fig. 5 und Fig. 6 beschrieben.

5.1 Subsampling eines DCT-Blocks

[0091] Bei der auf Vollbildern basierenden maßstabgerechten Video-Verkleinerung ist es notwendig, vier 8 × 8-DCT-Blöcke zu einem neuen 8 × 8-DCT-Block zusammenzuführen (spezifische Einzelheiten bei Betrachtung eines Halbbildblocks werden später beschrieben). Darüber hinaus sollte der Ausgangsblock eine Tiefpassversion der Eingangsblöcke sein. Dieser Prozess wird in der räumlichen Domäne ausgeführt, indem die Eingangsmatrix mit einer Subsampling-Matrix (vorzugsweise mit einem Tiefpassfilter) multipliziert wird. Multiplikation mit einer Subsampling-Matrix in der räumlichen Domäne ist wegen der distributiven Eigenschaft der Orthogonal-Transformation einer Multiplikation mit DCT-Koeffizienten einer Subsampling-Matrix in der DCT-Domäne äquivalent. Die Anzahl der Operationen (Berechnungen) in dem Downsampling-Prozess in der DCT-Domäne für bestimmte Downsampling-Filter kann jedoch sogar so hoch wie die Gesamtzahl der Operationen seines Gegenstücks in der räumlichen Domäne sein. Die Lösung dieses Problems besteht darin, eine Downsampling-Matrix zu verwenden, die dünn besiedelt ist (z.B. eine Matrix, die relativ wenige von null verschiedene Werte aufweist, z.B. ungefähr 30% oder weniger).

[0092] Eine dünn besiedelte Downsampling-Matrix kann auf der orthogonalen Eigenschaft zwischen dem DCT-Basisvektor und der Symmetriestruktur des DCT-Basisvektors basieren. Ein Ansatz, der in R. Dugad und N. Ahuja, „A Fast Scheme For Downsampling And Upsampling In The DCT Domain", International Conference on Image Processing (ICIP) 99, worauf hiermit ausdrücklich Bezug genommen wird, besprochen wird, nimmt die unteren 4 × 4 DCT-Koeffizienten aus vier Verarbeitungsblöcken, wendet auf jeden DCT-Subblock eine 4 × 4-IDCT an, bildet einen neuen 8 × 8-Pixel-Block und wendet eine 8 × 8-DCT an, um einen Ausgangsblock zu erhalten. Die Downsampling-Matrix kann im Voraus berechnet werden, da der Downsampling-Prozess festliegt. Durch Aufteilen der 8 × 8-DCT-Matrix in eine linke und eine rechte Hälfte ist etwa die Hälfte der Downsampling-Matrixwerte aufgrund der Orthogonalität zwischen der Spalte der 4 × 4-IDCT-Matrix und der Zeile sowohl der linken als auch der rechten 8 × 4-DCT-Matrizen null. Diese Operation (eine Dimension) kann mathematisch folgendermaßen geschrieben werden:

$$B = Tb = T\begin{bmatrix} b_1 \\ \cdots \\ b_2 \end{bmatrix} = \begin{bmatrix} T_L & \vdots & T_R \end{bmatrix}\begin{bmatrix} T_4^t B_1 \\ \cdots \\ T_4^t B_2 \end{bmatrix} = T_L T_4^t B_1 + T_R T_4^t B_2$$

dabei ist b ein räumlicher 8 × 1-Eingangsvektor, B ist sein entsprechender 8 × 1-DCT-Vektor, b_1 und b_2 sind unterabgetastete 4 × 1-Vektoren, B_1 und B_2 sind untere 4 × 1-DCT-Vektoren, T ist die 8 × 8-DCT-Transformationsmatrix, T_4 ist die 4 × 4-DCT-Transformationsmatrix, T_L und T_R sind die linke und rechte Hälfte von T. Der Index „t" bedeutet eine Matrixtransponierung. Der Algorithmus von Dugad verwendet auch die Symmetrieeigenschaft des DCT-Basisvektors, um die Komplexität des Downsampling-Prozesses zu verringern. $T_L T_4^t$ und $T_R T_4^t$ sind im Hinblick auf Betrag identisch ($T_L T_4^t(i, j) = (-1)^{i+j} T_R T_4^t((i, j), 0 \le i \le 7, 0 \le j \le 3$), da ungerade Zeilen von T antisymmetrisch und gerade Zeilen von Tsymmetrisch sind. „i" ist ein Matrix-Zeilenindex und „j" ist ein Matrix-Spaltenindex. Daher können sowohl $T_L T_4^t$ als auch $T_R T_4^t$ auf der Basis derselben Komponenten, d.h. eines symmetrischen Teils E (Index mit i + j gerade) und eines antisymmetrischen Teils O (Index mit i + j ungerade) berechnet werden ($T_L T_4^t = E + O$ und $T_R T_4^t = E - O$). Diese Anordnung verringert effektiv die Anzahl der Multiplikationen um einen Faktor von 2, wenn der Downsampling-Prozess folgendermaßen durchgeführt wird:

$$B = T_L T_4^t B_1 + T_R T_4^t B_2 = (E + O)B_1 + (E - O)B_2 = E(B_1 + B_2) + O(B_1 - B_2)$$

[0093] Die Implementierung der Methode von Dugad zur Umwandlung von vier Halbbildblöcken in einen Vollbildblock ist nicht so einfach. Eine Erweiterung des Downsampling jedes Prozesses in diesem Szenario (eine Dimension) kann folgendermaßen geschrieben werden:

$$B = T(S_L T_4^t B_T + S_B T_4^t B_B)$$

dabei sind B_T und B_B die unteren 4 × 1-Halbbildvektoren, S_T und S_B sind DCT-Werte einer 8 × 4-Entschachtelungsmatrix entsprechend ihrem oberen (S_T) bzw. unteren (S_B) Halbbildblock. Für Elemente von S_T gilt $S_T(i, j) = 1$ im Fall ($j = 2i, 0 \le i \le 3$) und $S_T(i, j) = 0$ andernfalls. Für Elemente von S_B gilt $S_B(i, j) = 1$ im Fall ($j = 2i + 1, 0 \le i \le 3$) und $S_B(i, j) = 0$ andernfalls.

[0094] Dies ist eine Modifikation des Algorithmus von Dugad zum Downsampling und Entschachteln gemäß der vorliegenden Erfindung.

[0095] Die Operationen des Prozesses der maßstabgerechten Verkleinerung und des Entschachtelns sind komplexer, da S und T nicht zueinander orthogonal sind und daher die Downsampling-Matrix nicht dünn besiedelt ist. C. Y im und M. A. Isnardi, „An Efficient Method For DCT-Domain Image Resizing With Mixed Field/Frame-Mode Macroblocks", IEEE Trans. Circ. and Syst. For Video Technol., Bd. 9, S. 696–700, Aug. 1999, worauf hiermit ausdrücklich Bezug genommen wird, schlagen ein effizientes Verfahren zum Downsampling eines Halbbildblocks vor. Ein Tiefpassfilter wird dergestalt in die Entschachtelungsmatrix integriert, dass die Downsampling-Matrix ($S = 0{,}5[I_8 I_8]$) dünn besiedelt ist.

[0096] I_8 bedeutet eine 8 × 8-Identitätsmatrix und $[I_8 I_8]$ bedeutet eine 16 × 8-Matrix, die eine Verkettung der beiden Identitätsmatrizen umfasst. Die Identitätsmatrix besitzt natürlich auf der Diagonalen nur Einsen und überall sonst nur Nullen.

[0097] Das Verfahren beginnt mit vier 8 × 8-IDCT-Halbbildblöcken, wendet dann die Downsampling-Matrix S an und führt eine 8 × 8-DCT durch, um den Ausgangsblock zu erhalten. Man beachte, dass bei diesem Verfahren an Stelle einer 4 × 4-IDCT eine 8 × 8-IDCT verwendet wird. diese Operation kann mathematisch (in ei-

ner Dimension) folgendermaßen gezeigt werden:

$$D = TS\begin{bmatrix} T^t & 0 \\ 0 & T^t \end{bmatrix} = \frac{1}{2}T\begin{bmatrix} I_8 & \vdots & I_8 \end{bmatrix}\begin{bmatrix} T^t & 0 \\ 0 & T^t \end{bmatrix} = \frac{1}{2}\begin{bmatrix} I_8 & \vdots & I_8 \end{bmatrix}$$

5.2 Unterabtastung von MV-Daten

[0098] ME ist der Engpass des gesamten Videocodierungsprozesses. Es ist daher wünschenswert, einen MV des umbemessenen MB zu schätzen, indem man MVs von vier ursprünglichen MBs benutzt, ohne tatsächlich ME durchzuführen (vorausgesetzt, dass alle MBs im Inter-Modus codiert sind). Man beachte, dass, wenn ein MPEG-2-Bitstrom angenommen wird, die Unterabtastung von MV-Daten MVs von vier MBs nimmt, da jeder MB einen Eingang besitzt (nur ein MPEG-4-Bitstrom kann einen MV für jeden Block aufweisen). Die einfachste Lösung besteht darin, vier MVs miteinander zu mitteln, um den neuen MV zu erhalten, man erhält aber eine schlechte Schätzung, wenn diese vier MVs verschieden sind. B. Shen, I. K. Sethi und B. Vasudev, „Adaptive Motion-Vector Resampling For Compressed Video Downscaling", IEEE Trans. Circ. and Syst. For Video Technol., Bd. 9, S. 929–936, Sep. 1999, zeigen, dass man ein besseres Ergebnis erhalten kann, indem dem schlechtesten vorhergesagten MV mehr Gewicht gegeben wird. Eine Übereinstimmungsgenauigkeit 4 jedes MV wird durch die Anzahl der von Null verschiedenen AC-Koeffizienten in diesem MB angegeben. Durch Verwendung der Technik von Shen et al. kann der neue MV für den maßstabgerecht verkleinerten MB folgendermaßen berechnet werden:

$$MV' = \frac{1}{2}\frac{\sum_{i=1}^{4} MV_i A_i}{\sum_{i=1}^{4} A_i}$$

[0099] M. R. Hashemi, L. Winger und S. Panchanathan, „Compressed Domain Motion Vector Resampling For Downscaling Of MPEG Video", ICIP 99, schlagen eine nichtlineare Methode zur Schätzung des MV des umbemessenen MB vor. Ähnlich wie bei dem Algorithmus in Shen et al. verwendet die Technik von Hashemi räumliche Aktivität der Verarbeitungs-MBs zur Schätzung des neuen MV. Eine als MAC (Maximum Average Correlation) bezeichnete heuristische Messung wird bei dem Verfahren von Hashemi zur Identifizierung eines der vier ursprünglichen MVs als der Ausgangs-MV verwendet. Durch Verwendung von MAC kann der neue MV für den maßstabgerecht verkleinerten MP folgendermaßen berechnet werden:

$$MV = \max \sum_{i=1}^{4} A_i \rho^{d_i}$$

dabei ist p die räumliche Korrelation und wird auf 0,85 gesetzt und d_i ist die euklidische Distanz zwischen dem i-ten Eingangs-MV (MV_i) und dem Ausgangs-MV.

6. Implementierung des Größentranscoders

[0100] Fig. 5 zeigt einen Größentranscoder gemäß der Erfindung. B-Vollbilder können im Eingangsbitstrom vorhanden sein, werden aber vom Transcoder verworfen und erscheinen deshalb nicht im Ausgabebitstrom.

[0101] In dem Transcoder **500** werden eine MV-Skalierungsfunktion **510**, eine DCT-Skalierungsfunktion **520** und eine räumliche Skalierungsfunktion **540** hinzugefügt. Die Schalter **530** und **535** werden so koordiniert, dass in einer ersten Stellung eine Ausgabe der DCT-Funktion **455** zu der Quantisierungsfunktion **340** geroutet wird, und der Schalter **535** wird geschlossen, damit eine Ausgabe der räumlichen Skalierungsfunktion **540** in den Addierer **445** eingegeben werden kann. In einer zweiten Stellung der Schalter **530** und **535** wird eine Ausgabe der DCT-Skalierungsfunktion **520** in die Quantisierungsfunktion **340** geroutet, und der Schalter **535** ist offen.

[0102] Der Transcoder **500** wandelt einen MPEG-2-Bitstrom in einen MPEG-4-Bitstrom um, der Video mit kleinerer Größe entspricht, z.B. von ITU-R **601** (720 × 400) zu SIF (352 × 240).

[0103] Um eine Bandbreitenanforderung für den MPEG-4-Bitstrom zu erfüllen, führt der Transcoder **500** eine Unterabtastung des Video sowohl in der horizontalen als auch der vertikalen Richtung (in der räumlichen Skalierungsfunktion **540**) um zwei durch und überspringt alle B-Vollbilder (in den zeitlichen Skalierungsfunktionen **545** und **546**), wodurch entsprechend die zeitliche Auflösung verringert wird. Man beachte, dass die zeitliche Skalierungsfunktion **546** als Alternative nach der DCT-Skalierungsfunktion **520** vorgesehen werden könnte. Das Überspringen von B-Vollbildern vor der Durchführung der maßstabgerechten Verkleinerung verringert die Komplexität.

[0104] Darüber hinaus sollte ein Tiefpassfilter (das in der räumlichen Skalierungsfunktion **540** vorgesehen werden kann) vor der Unterabtastung die Bildqualität verbessern.

[0105] Die Erfindung kann mit geringfügigen Modifikationen erweitert werden, so dass sie andere Downsampling-Faktoren und B-VOPs beinhaltet. Genauer gesagt werden Änderungen an der maßstabgerechten MV-Verkleinerung und der Modusentscheidung vorgenommen. Die maßstabgerechte MV-Verkleinerung für B-VOP ist eine direkte Erweiterung des zur Aufnahme des Rückwärts-MV Besprochenen. Die Modusentscheidung für B-VOP kann auf ähnliche Weise wie bei P-VOP gehandhabt werden (z.B. durch Umwandeln des unidirektionalen MV in einen bidirektionalen MV, wie bei der Umwandlung von Intra-MB in Inter-MB bei einer P-VOP).

[0106] Nachfolgend werden sechs Probleme besprochen, die der Größentranscoder **500** behandelt. Es wird außerdem angenommen, dass das Eingangsvideo 704 × 480 Pixel Auflösung aufweist und mit einem Coder des Typs MP@ML MPEG-2 codiert ist und die gewünschte Ausgabe ein MPEG-4-Bitstrom des einfachen Profils ist, der progressives SIF-Video (mit einer Vollbildratenreduktion um N) enthält. Die Erfindung kann jedoch auch auf andere Eingangs- und Ausgangsformate erweitert werden.

6.1 Maßstabgerechte MV-Verkleinerung für progressives Video (Luma)

[0107] Dieses Problem erscheint, wenn alle vier MBs als Inter codiert sind und Vollbildprädiktion verwenden. Jeder MV in diesen MBs wird in jeder Richtung (horizontal und vertikal) maßstabgerecht um zwei verkleinert, um den MV von vier Blöcken in MPEG-4 zu bestimmen (MPEG-4 erlaubt einen MV pro 8 × 8-Block). Die skalierten MVs werden dann (unter Verwendung eines Medianfilters) unter Verwendung der normalen MPEG-4-Prozedur prädiktiv codiert.

[0108] Man beachte, dass jeder (vier Blöcke umfassende) MB sowohl in MPEG-2 als auch in MPEG-4 im selben Modus codiert sein muss. Bei maßstabgerechter Video-Verkleinerung entspricht der Ausgangs-MB (vier Blöcke) vier Eingangs-MBs.

6.2 Maßstabgerechte MV-Verkleinerung für verschachteltes Video (Luma)

[0109] Dieses Problem existiert, wenn alle vier MBs als Inter codiert sind und Halbbildprädiktion verwenden. Man muss zwei Halbbild-MVs in jedem MB kombinieren, um einen Vollbild-MV des umbemessenen Blocks zu erhalten. Anstatt den neuen MV auf der Basis der räumlichen Aktivität festzusetzen, sucht der vorgeschlagene Transcoder den neuen MV auf der Basis der MVs seiner Nachbarn aus. Die MVs aller acht umgebenden MBs werden verwendet, um einen Prädiktor zu finden (Halbbild-MVs werden im Fall eines MB mit Halbbildprädiktion gemittelt). Der Medianwert aus diesen acht MVs wird zu einem Prädiktor und der Halbbild-MV des aktuellen MB, der im Hinblick auf euklidische Distanz näher ist, wird in der horizontalen Richtung um zwei skaliert, um zu dem neuen MV zu werden.

6.3 MV-Downsampling (Chroma)

[0110] Dieses Problem tritt auf, wenn alle vier MBs als Inter codiert sind und entweder Vollbild- oder Halbbildprädiktion verwenden (MPEG-4 behandelt beide Prädiktionsmodi für einen Chroma-Block auf dieselbe Weise). Der Prozess folgt der MPEG-4-Methode, um einen Chroma-MV aus einem Luma-MV zu erhalten, d.h. ein Chroma-MV ist die maßstabgerecht verkleinerte Version des Mittelwerts seiner vier entsprechenden 8 × 8-Luma-MVs.

6.4 DCT-Downsampling (Luma-progressiv, Chroma)

[0111] Dieses Problem tritt auf, wenn alle vier Luma-MBs als Intra oder Inter codiert sind und Vollbild-MB-Struktur verwenden und ihre acht Chroma-Blöcke (vier für Cr und vier für Cb) entweder Vollbild- oder Halbbildstruktur verwenden). Die Methode von Dugad wird benutzt, um die Luma- und Chroma-DCT-Blöcke in

jeder Richtung maßstabgerecht um einen Faktor von zwei zu verkleinern.

6.5 Verschachteltes DCT-Downsampling (Luma)

[0112] Zu diesem Problem kommt es auf eine von zwei Weisen. Erstens verwendet sein assoziierter MB Halbbildprädiktion und zweitens verwendet sein assoziierter MB Vollbildprädiktion. In jedem Fall will man vier 8 × 8-Halbbild-DCT-Blöcke (zwei für das obere Halbbild und zwei für das untere Halbbild) maßstabgerecht zu einem 8 × 8-Vollbild-DCT-Block verkleinern. Die Lösung für den ersten Fall besteht darin, denselben Halbbild-DCT-Block zu verwenden, wie der, der für MC gewählt wurde. Bei dem zweiten Fall erfolgt Entschachtelung und es wird eine Kombination der oben besprochenen Methoden von Dugad und Yim vorgeschlagen.

[0113] Auf der Basis des Yim-Algorithmus verkleinert genauer gesagt der Transcoder zunächst vier Halbbildblöcke maßstabsgerecht in der vertikalen Richtung (und führt gleichzeitig Entschachtelung durch), um zwei Vollbildblöcke zu erhalten. Unter Verwendung des Dugad-Algorithmus verkleinert der Transcoder dann diese beiden Vollbildblöcke maßstabgerecht in der horizontalen Richtung, um den Ausgangsblock zu erhalten.

[0114] Dies ist in Fig. 6 dargestellt, wobei bei **600** vier 8 × 8-Koeffizienten-Halbbildmodus-DCT-Blöcke, bei **610** zwei 8 × 8-Vollbildmodus-DCT-Blöcke und bei **620** ein 8 × 8-Vollbildmdus-DCT-Block gezeigt sind.

[0115] Die Prozedur für das maßstabgerechte DCT-Verkleinern gemäß der Erfindung kann folgendermaßen zusammengefasst werden:

1. Durch Kombinieren von vier Halbbildblöcken miteinander wie bei 600 gezeigt, die 16 × 16-Koeffizienten-Eingangsmatrix bilden.
2. Für vertikale maßstabgerechte Verkleinerung und Filterung gemäß dem Algorithmus von Yim auf jede Zeile der Eingangsmatrix ein Tiefpassfilter (LP) D anwenden. Die LP-Eingangsmatrix beträgt nun 16 × 8 Pixel, wie bei **610** gezeigt.
3. Aus der LP-Matrix ($[B_1 \vdots B_2]$) 8 × 8-Matrizen B_1 und B_2 bilden.
4. Gemäß dem Algorithmus von Dugad an jeder Spalte von B_1 und B_2 eine Operation der maßstabgerechten horizontalen Verkleinerung durchführen, um die Ausgangsmatrix (8 × 8) (**620**) folgendermaßen zu erhalten:

$$B = B_1(T_L T_4^t)^t + B_2(T_R T_4^t)^t = (B_1 + B_2)E + (B_1 - B_2)O$$

wobei E und O wie oben besprochen gerade und ungerade Zeilen bedeuten.

[0116] Insbesondere kann man (den Skalierungsfaktor ignorierend) eine Horizontal-Downsampling-Matrix verwenden, die folgendermaßen aus ungeraden („O") und geraden („E") Matrizen besteht:

```
E = [e(0)  0     0     0,
     0     e(1)  0     e(2),
     0     0     0     0,
     0     e(3)  0     e(4),
     0     0     e(5)  0,
     0     e(6)  0     e(7),
     0     0     0     0,
     0     e(8)  0     e(9)].

O = [0     0     0     0,
     o(0)  0     o(1)  0,
     0     o(2)  0     0,
     o(3)  0     o(4)  0,
     0     0     0     0,
     o(5)  0     o(6)  0,
     0     0     0     o(7),
     o(8)  0     o(9)  0].
```

[0117] Die Koeffizienten wie folgt können verwendet werden:

e(0)= 4
e(1) = 0,831469612
e(2) = 0,045774654
e(3) = 1,582130167
e(4) = –0,195090322
e(5) = 2
e(6) = –0,704885901
e(7) = 0,980785280
e(8) = 0,906127446
e(9) = 1,731445835

o(0) = 2,56915448
o(1) = –0,149315668
o(2) = 2
o(3) = –0,899976223
o(4) = 1,026559934
o(5) = 0,601344887
o(6) = 1,536355513
o(7) = 2
o(8) = –0,509795579
o(9) = –0,750660555

[0118] Im Wesentlichen wird das Produkt einer DCT-Matrix, die dünn besiedelt ist, als die Downsampling-Matrix benutzt.

[0119] Die Technik kann im Allgemeinen für eine 2:1-Abbemessung eines N × N-Blocks erweitert werden, der vier N/2 × N/2-Koeffizienten-Halbbildmodus-Blöcke umfasst. Es können auch andere Abbemessungsverhältnisse ermöglicht werden.

6.6 Spezialfälle

[0120] Spezialfälle treten auf, wenn nicht alle vier MBs im selben Modus codiert sind (nicht in einen der fünf vorherigen Fälle fallend). Es wird immer angenommen, dass ein Intra- oder übersprungener MB unter den anderen Inter-MBs mit Null-MV Inter-Modus sind. Halbbild-MVs werden auf der Basis von Abschnitt 6.2 zusammengeführt, um Vollbild-MV zu erhalten und man wendet dann die Techniken des Abschnitts 6.1 an. MC wird empfohlen, um die Textur des Intra-Blocks zu bestimmen, der von dem Transcoder als ein Inter-Block mit Null-MV behandelt wird.

7. Schlussfolgerung

[0121] Es sollte nun erkennbar sein, dass die vorliegende Erfindung eine Transcoderarchitektur bereitstellt, die die kleinstmögliche Komplexität mit einem kleinen Fehler bereitstellt. Dieser Fehler wird in dem MPEG-4-Texturcodierungsprozess erzeugt (QP-Codierung, DC-Prädiktion, nichtlinearer DC-Skalierer). Diese

Prozesse sollten in dem zukünftigen Profil von MPEG-4 entfernt werden, um ein nahezu verlustloses Transcodierungssystem zu erzeugen.

[0122] Die Erfindung gibt außerdem vollständige Einzelheiten eines Größentranscoders an, um einen Bitstrom verschachtelter Videocodierung gemäß ITU-R **601** mit MPEG-2 MP@ML in einen MPEG-4-Bitstrom des einfachen Profils umzuwandeln, der für eine Streaming-Videoanwendung geeignetes progressives SIF-Video enthält.

[0123] Zur räumlichen maßstabgerechten Verkleinerung von Halbbildmodus-DCT-Blöcken wird vorgeschlagen, Vertikal- und Horizontal-Techniken der maßstabgerechten Verkleinerung auf neuartige Weise dergestalt zu kombinieren, dass sowohl in der vertikalen als auch in der horizontalen Richtung dünn besiedelte Downsampling-Matrizen verwendet werden, wodurch Berechnungen des Transcoders verringert werden.

[0124] Für maßstabgerechte MV-Verkleinerung wird darüber hinaus vorgeschlagen, einen Medianwert aus seinen acht benachbarten MV zu benutzen. Dieser Vorschlag funktioniert besser als Algorithmen im Abschnitt 5.2, da der vorliegende vorhergesagte MV mit dem globalen MV zusammengeht. Außerdem funktioniert er gut mit einem verschachtelten MB, der anstelle von vier MVs pro MB nur zwei MVs aufweist.

[0125] Obwohl die Erfindung in Verbindung mit verschiedenen bevorzugten Ausführungsformen beschrieben wurde, versteht sich, dass verschiedene Modifikationen und Anpassungen daran vorgenommen werden können, ohne von dem in den Ansprüchen dargelegten Schutzumfang der Erfindung abzuweichen.

Patentansprüche

1. Ein Verfahren zum Transcodieren eines vorkomprimierten Eingangsbitstroms, der in einem ersten Videocodierungsformat bereitgestellt wird, in einen komprimierten Ausgabebitstrom, der in einem zweiten Videocodierungsformat bereitgestellt wird, wobei das erste Videocodierungsformat MPEG-2 Main Profile/Main Level beinhaltet und das zweite Videocodierungsformat einen MPEG-4-Bitstrom mit progressivem Video in Normzwischenformat (SIF = Standard Intermediate Format) beinhaltet, wobei das Verfahren die folgenden Schritte beinhaltet:
Gewinnen von Kopfinformationen aus dem Eingangsbitstrom;
Bereitstellen von entsprechenden Kopfinformationen in dem zweiten, unterschiedlichen Videocodierungsformat;
teilweises Dekomprimieren des Eingangsbitstroms, um teilweise dekomprimierte Daten bereitzustellen; und
Neukomprimieren der teilweise dekomprimierten Daten gemäß den Kopfinformationen in dem zweiten Format, um den Ausgabebitstrom bereitzustellen.

2. Verfahren gemäß Anspruch 1, wobei:
die teilweise dekomprimierten Daten Bewegungsvektoren und Koeffizienten der diskreten Cosinus-Transformation (DCT) beinhalten; und
das zweite Format mindestens eines von einer Neuer-Modus-Entscheidung, AC/DC-Vorhersage (AC/DC-Prediction) und Bewegungskompensation beinhaltet.

3. Verfahren gemäß einem der Ansprüche 1 und 2, wobei:
mindestens eine Verweistabelle verwendet wird, um die entsprechenden Kopfinformationen in dem zweiten Videocodierungsformat bereitzustellen.

4. Verfahren gemäß einem der Ansprüche 1 bis 3, wobei:
an den teilweise dekomprimierten Daten durch Downsampling von DCT-Koeffizienten und Bewegungsvektordaten davon eine maßstabgerechte Verkleinerung durchgeführt wird.

5. Verfahren gemäß einem der Ansprüche 1 bis 4, wobei:
an mindestens einer Gruppe von vier Halbbildmodus-DCT-Blöcken der teilweise dekomprimierten Daten eine maßstabgerechte 2:1-Verkleinerung durchgeführt wird, indem daran ein vertikales Downsampling und Deinterlacing durchgeführt wird, um eine entsprechende Gruppe von zwei Vollbildmodus-DCT-Blöcken zu erhalten, und indem an den zwei Vollbildmodus-DCT-Blöcken ein horizontales Downsampling durchgeführt wird, um einen Vollbildmodus-DCT-Block zu erhalten.

6. Verfahren gemäß Anspruch 5, wobei:
das vertikale Downsampling auch eine Tiefpassfilterung der vier Halbbildmodus-DCT-Blöcke erreicht.

7. Verfahren gemäß Anspruch 5 oder 6, wobei:
das vertikale und das horizontale Downsampling jeweilige dünn besiedelte Matrizen verwenden.

8. Verfahren gemäß einem der Ansprüche 1 bis 7, wobei:
in dem Neukomprimierungsschritt ein Code (DQUANT), der eine Veränderung bei einem Quantisierer spezifiziert, gemäß einem Differenzialwert eines Quantisierungsparameters der teilweise dekomprimierten Daten festgesetzt wird.

9. Verfahren gemäß einem der Ansprüche 1 bis 8, wobei:
zum Neukomprimieren intracodierter Makroblöcke ein codiertes Blockmuster (CBP = Coded Block Pattern) gemäß einem entsprechenden Wert der teilweise dekomprimierten Daten festgesetzt wird.

10. Verfahren gemäß einem der Ansprüche 1 bis 9, wobei: zum Neukomprimieren nicht intracodierter Makroblöcke übersprungene Makroblöcke in den teilweise dekomprimierten Daten als not_coded-Makroblöcke (nicht_codierte Makroblöcke) codiert werden, bei denen alle Koeffizienten der diskreten Cosinus-Transformation (DCT) einen Nullwert aufweisen.

11. Verfahren gemäß einem der Ansprüche 1 bis 10, wobei:
in dem Neukomprimierungsschritt vorhergesagte Bewegungsvektoren in den teilweise dekomprimierten Daten gemäß dem zweiten Format neu festgesetzt werden.

12. Verfahren gemäß einem der Ansprüche 1 bis 11, wobei:
in dem Neukomprimierungsschritt Dual-Prime-Modus-Makroblöcke der teilweise dekomprimierten Daten in halbbildcodierte Makroblöcke umgewandelt werden.

13. Verfahren gemäß einem der Ansprüche 1 bis 12, wobei:
das zweite Videocodierungsformat einen MPEG-4-Simple-Profile-Bitstrom mit progressivem Video in Normzwischenformat (SIF) beinhaltet.

14. Eine Vorrichtung zum Transcodieren eines vorkomprimierten Eingangsbitstroms, der in einem ersten Videocodierungsformat bereitgestellt wird, in einen komprimierten Ausgabebitstrom, der in einem zweiten Videocodierungsformat bereitgestellt wird, wobei das erste Videocodierungsformat MPEG-2 Main Profile/Main Level beinhaltet und das zweite Videocodierungsformat einen MPEG-4-Bitstrom mit progressivem Video in Normzwischenformat (SIF) beinhaltet, wobei die Vorrichtung Folgendes beinhaltet:
Mittel (**304**) zum Gewinnen von Kopfinformationen aus dem Eingangsbitstrom;
Mittel (**308**) zum Bereitstellen von entsprechenden Kopfinformationen in dem zweiten, unterschiedlichen Videocodierungsformat;
Mittel (**310**, **320**, **330**) zum teilweisen Dekomprimieren des Eingangsbitstroms, um teilweise dekomprimierte Daten bereitzustellen; und
Mittel (**340**, **350**, **360**, **370**) zum Neukomprimieren der teilweise dekomprimierten Daten gemäß den Kopfinformationen in dem zweiten Format, um den Ausgabebitstrom bereitzustellen.

15. Vorrichtung gemäß Anspruch 14, wobei:
die teilweise dekomprimierten Daten Bewegungsvektoren und Koeffizienten der diskreten Cosinus-Transformation (DCT) beinhalten; und
das zweite Format mindestens eines von einer Neuer-Modus-Entscheidung, AC/DC-Vorhersage (**350**) und Bewegungskompensation (**430**) beinhaltet.

16. Vorrichtung gemäß einem der Ansprüche 14 oder 15, wobei:
mindestens eine Verweistabelle (**308**) verwendet wird, um die entsprechenden Kopfinformationen in dem zweiten Videocodierungsformat bereitzustellen.

17. Vorrichtung gemäß einem der Ansprüche 14 bis 16, wobei:
an den teilweise dekomprimierten Daten durch Downsampling von DCT-Koeffizienten und Bewegungsvektordaten davon eine maßstabgerechte Verkleinerung (**520**) durchgeführt wird.

18. Vorrichtung gemäß einem der Ansprüche 14 bis 17, wobei:
an mindestens einer Gruppe von vier Halbbildmodus-DCT-Blöcken der teilweise dekomprimierten Daten eine maßstabgerechte 2:1-Verkleinerung (**540**) durchgeführt wird, indem daran ein vertikales Downsampling und Deinterlacing durchgeführt wird, um eine entsprechende Gruppe von zwei Vollbildmodus-DCT-Blöcken zu er-

halten, und indem an den zwei Vollbildmodus-DCT-Blöcken ein horizontales Downsampling durchgeführt wird, um einen Vollbildmodus-DCT-Block zu erhalten.

19. Vorrichtung gemäß Anspruch 18, wobei:
das vertikale Downsampling auch eine Tiefpassfilterung der vier Halbbildmodus-DCT-Blöcke erreicht.

20. Vorrichtung gemäß Anspruch 18 oder 19, wobei:
das vertikale und das horizontale Downsampling jeweilige dünn besiedelte Matrizen verwenden.

21. Vorrichtung gemäß einem der Ansprüche 14 bis 20, wobei:
das Mittel (**340**) zum Neukomprimieren einen Code (DQUANT), der eine Veränderung bei einem Quantisierer spezifiziert, gemäß einem Differenzialwert eines Quantisierungsparameters der teilweise dekomprimierten Daten festsetzt.

22. Vorrichtung gemäß einem der Ansprüche 14 bis 21, wobei:
zum Neukomprimieren intracodierter Makroblöcke ein codiertes Blockmuster (CBP) gemäß einem entsprechenden Wert der teilweise dekomprimierten Daten festgesetzt wird.

23. Vorrichtung gemäß einem der Ansprüche 14 bis 22, wobei:
zum Neukomprimieren nicht intracodierter Makroblöcke übersprungene Makroblöcke in den teilweise dekomprimierten Daten als not_coded-Makroblöcke codiert werden, bei denen alle Koeffizienten der diskreten Cosinus-Transformation (DCT) einen Nullwert aufweisen.

24. Vorrichtung gemäß einem der Ansprüche 14 bis 23, wobei:
das Mittel (**430**) zum Neukomprimieren vorhergesagte Bewegungsvektoren in den teilweise dekomprimierten Daten gemäß dem zweiten Format neu festsetzt.

25. Vorrichtung gemäß einem der Ansprüche 14 bis 24, wobei:
das Mittel (**370**) zum Neukomprimieren Dual-Prime-Modus-Makroblöcke der teilweise dekomprimierten Daten in halbbildcodierte Makroblöcke umwandelt.

26. Vorrichtung gemäß einem der Ansprüche 14 bis 25, wobei:
das zweite Videocodierungsformat einen MPEG-4-Simple-Profile-Bitstrom mit progressivem Video in Normzwischenformat (SIF) beinhaltet.

Es folgen 8 Blatt Zeichnungen

Anhängende Zeichnungen

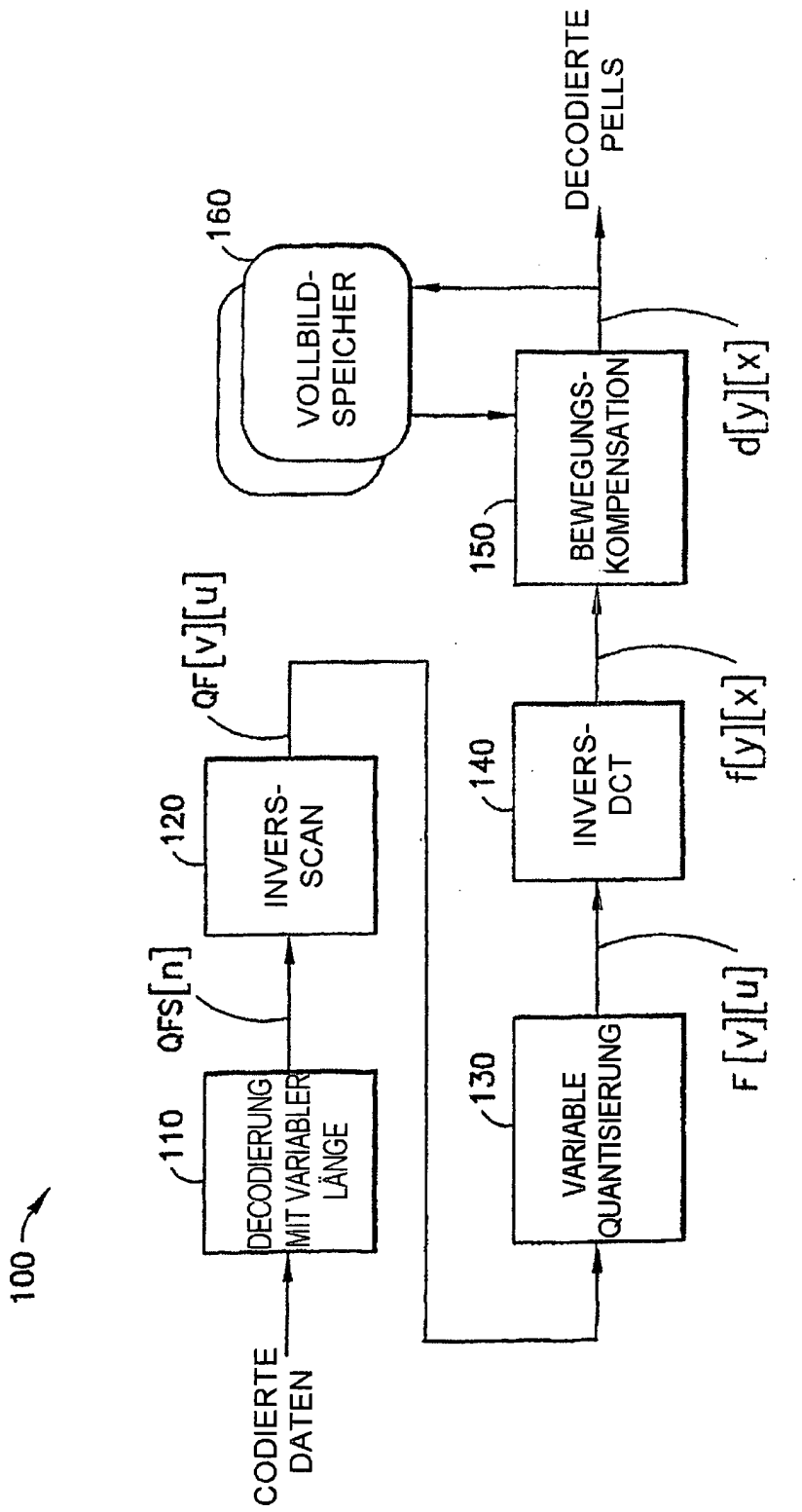

FIG. 1
STAND DER TECHNIK

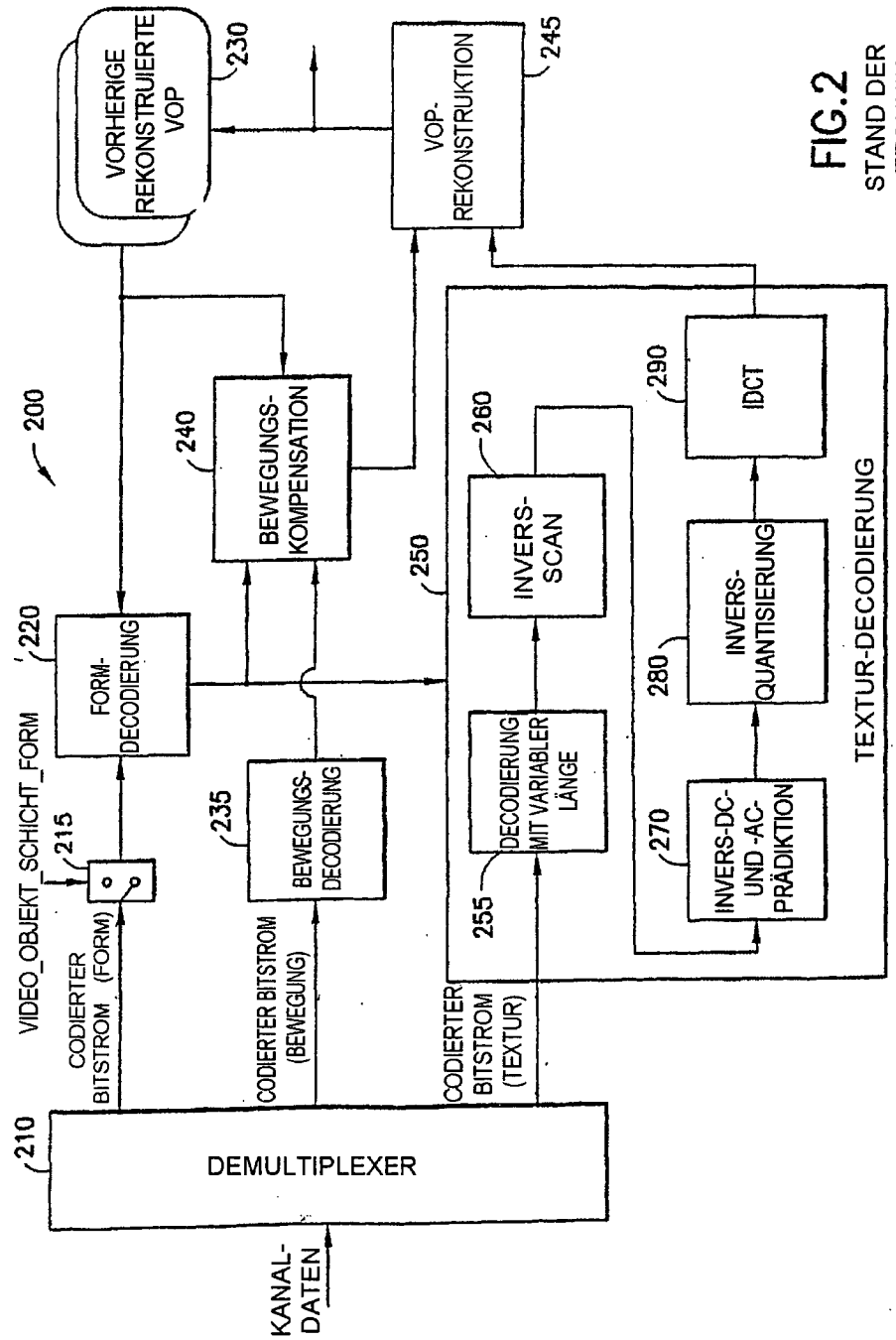

FIG.2
STAND DER TECHNIK

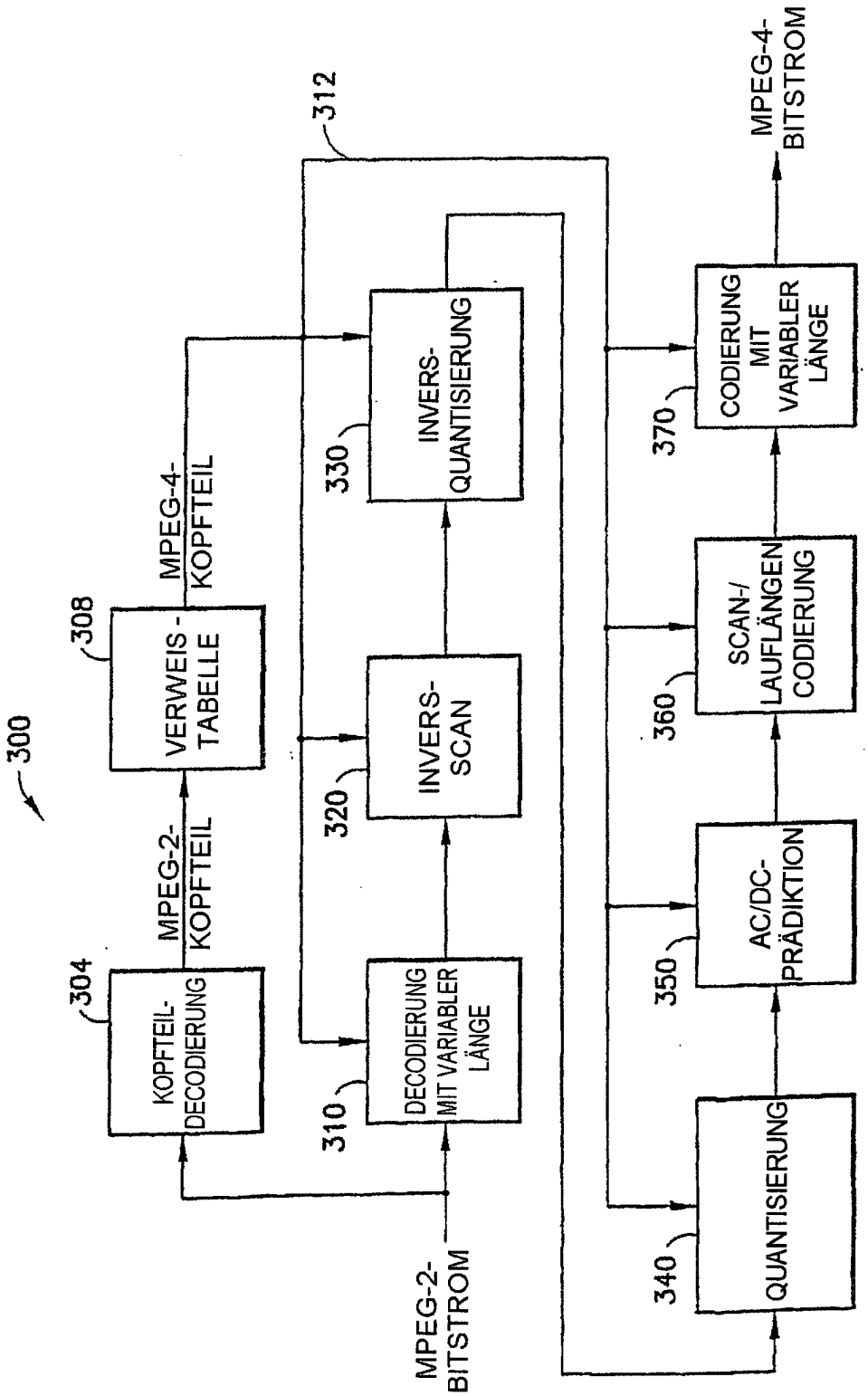
300
304
KOPFTEIL-DECODIERUNG
MPEG-2-KOPFTEIL
308
VERWEIS-TABELLE
MPEG-4-KOPFTEIL
MPEG-2-BITSTROM
310
DECODIERUNG MIT VARIABLER LÄNGE
320
INVERS-SCAN
330
INVERS-QUANTISIERUNG
312
340
QUANTISIERUNG
350
AC/DC-PRÄDIKTION
360
SCAN-/LAUFLÄNGEN CODIERUNG
370
CODIERUNG MIT VARIABLER LÄNGE
MPEG-4-BITSTROM

FIG.3

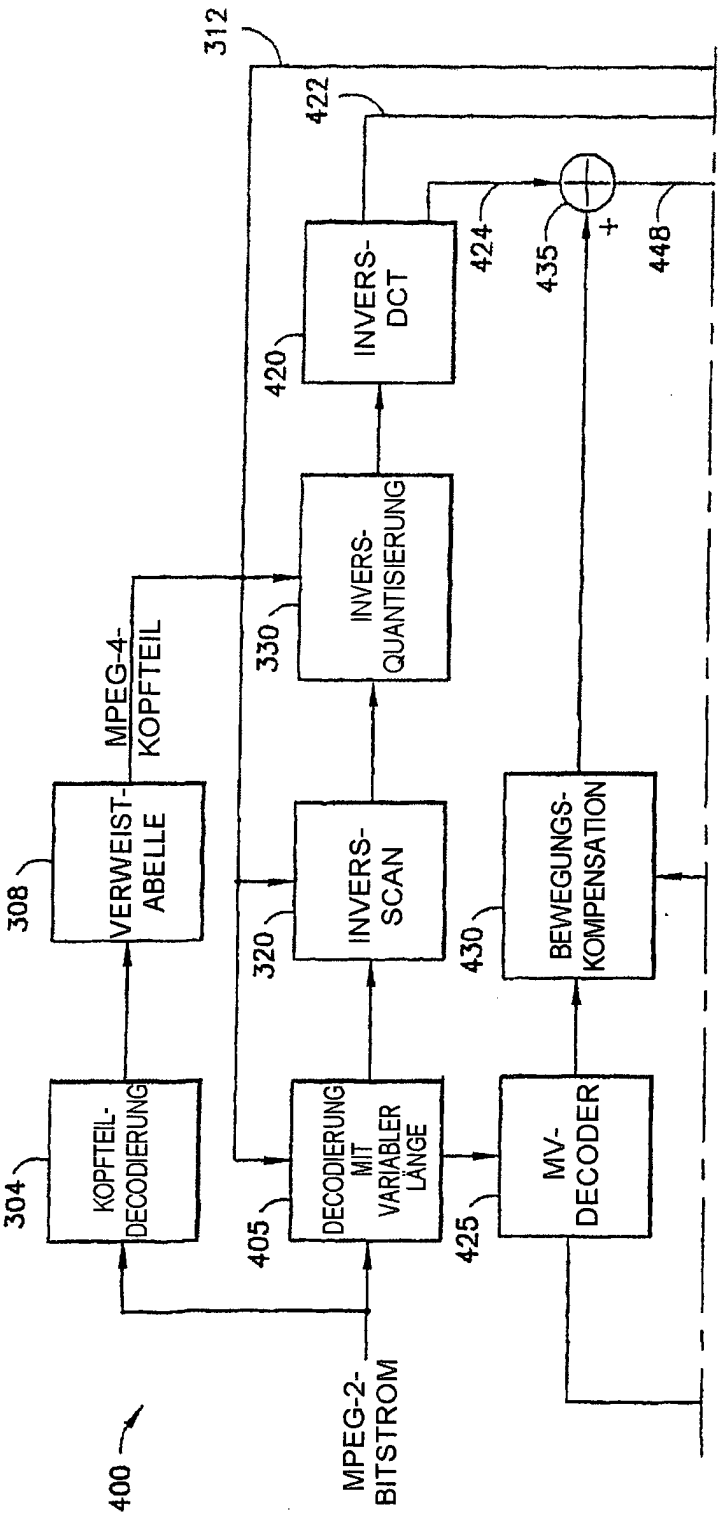

FIG.4A

FIG.4B

FIG.4

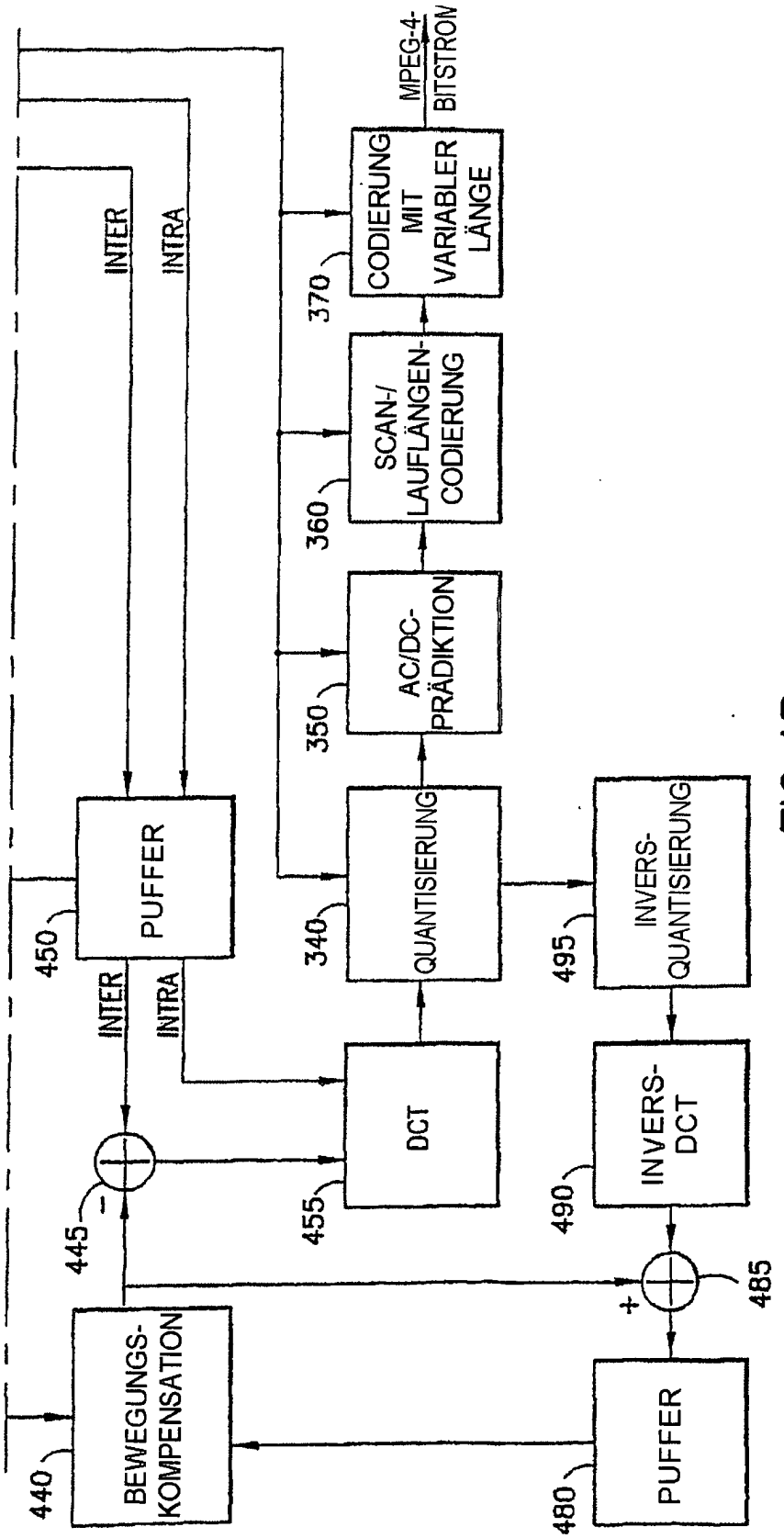
MPEG-4-BITSTROM
CODIERUNG MIT VARIABLER LÄNGE
370
SCAN-/ LAUFLÄNGEN-CODIERUNG
360
AC/DC-PRÄDIKTION
350
QUANTISIERUNG
340
INVERS-QUANTISIERUNG
495
INVERS-DCT
490
PUFFER
480
+
485
DCT
455
-
445
INTER
INTRA
PUFFER
450
INTER
INTRA
BEWEGUNGS-KOMPENSATION
440

FIG.4B

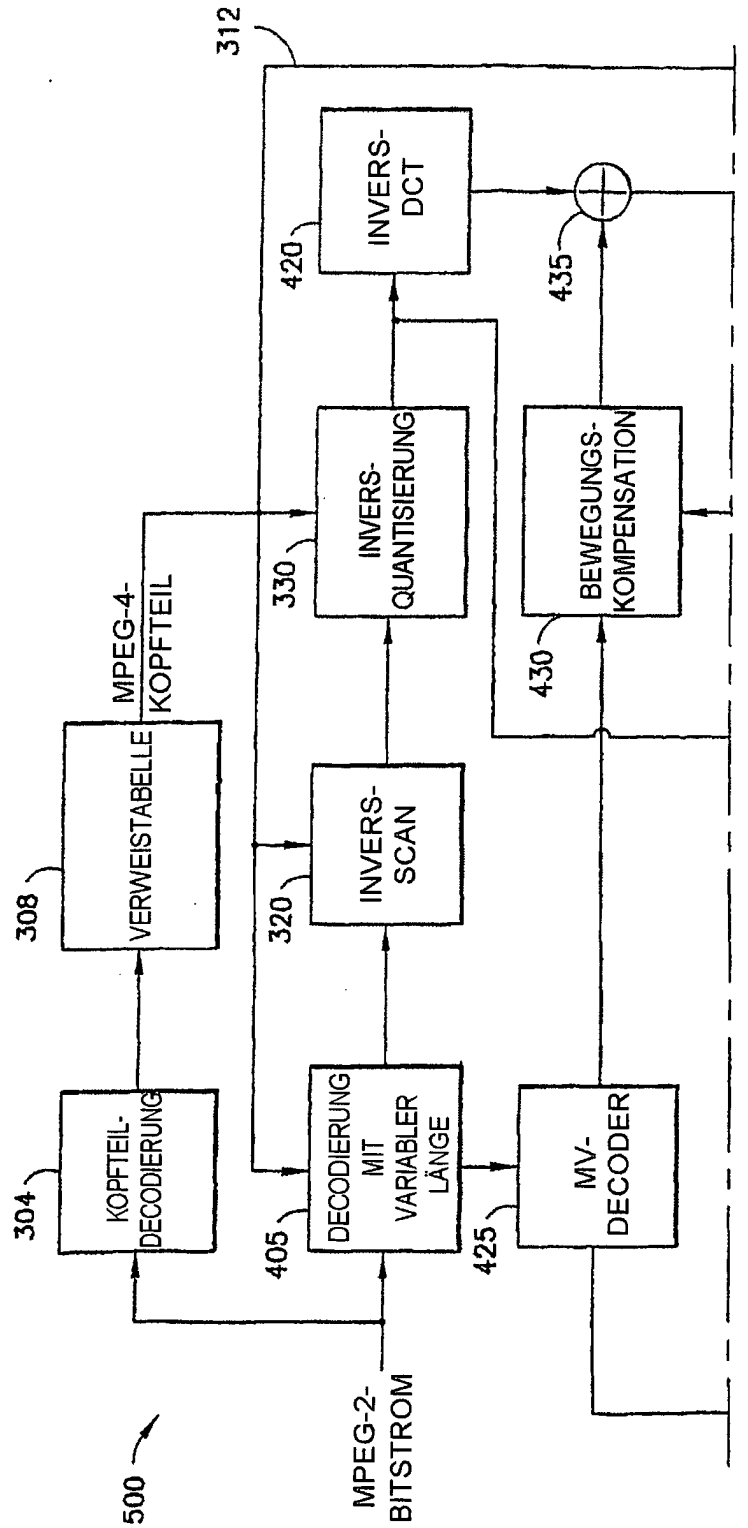

FIG.5A

FIG.5B

FIG.5

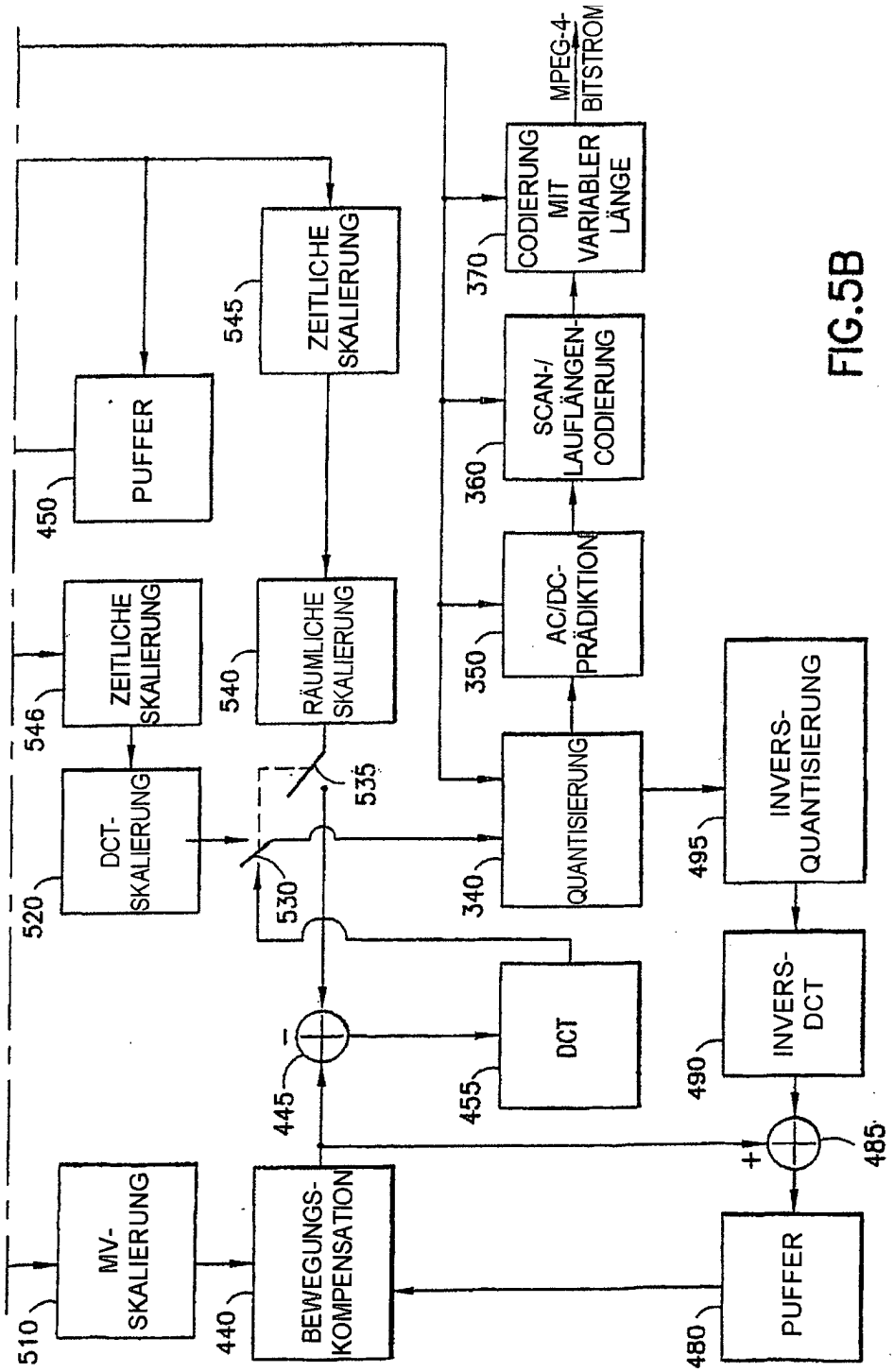
MPEG-4-BITSTROM
CODIERUNG MIT VARIABLER LÄNGE
370
SCAN-/ LAUFLÄNGEN-CODIERUNG
360
AC/DC-PRÄDIKTION
350
QUANTISIERUNG
340
INVERS-QUANTISIERUNG
495
INVERS-DCT
490
PUFFER
480
ZEITLICHE SKALIERUNG
545
PUFFER
450
RÄUMLICHE SKALIERUNG
540
ZEITLICHE SKALIERUNG
546
DCT-SKALIERUNG
520
535
530
DCT
455
445
MV-SKALIERUNG
510
BEWEGUNGS-KOMPENSATION
440
485

FIG.5B

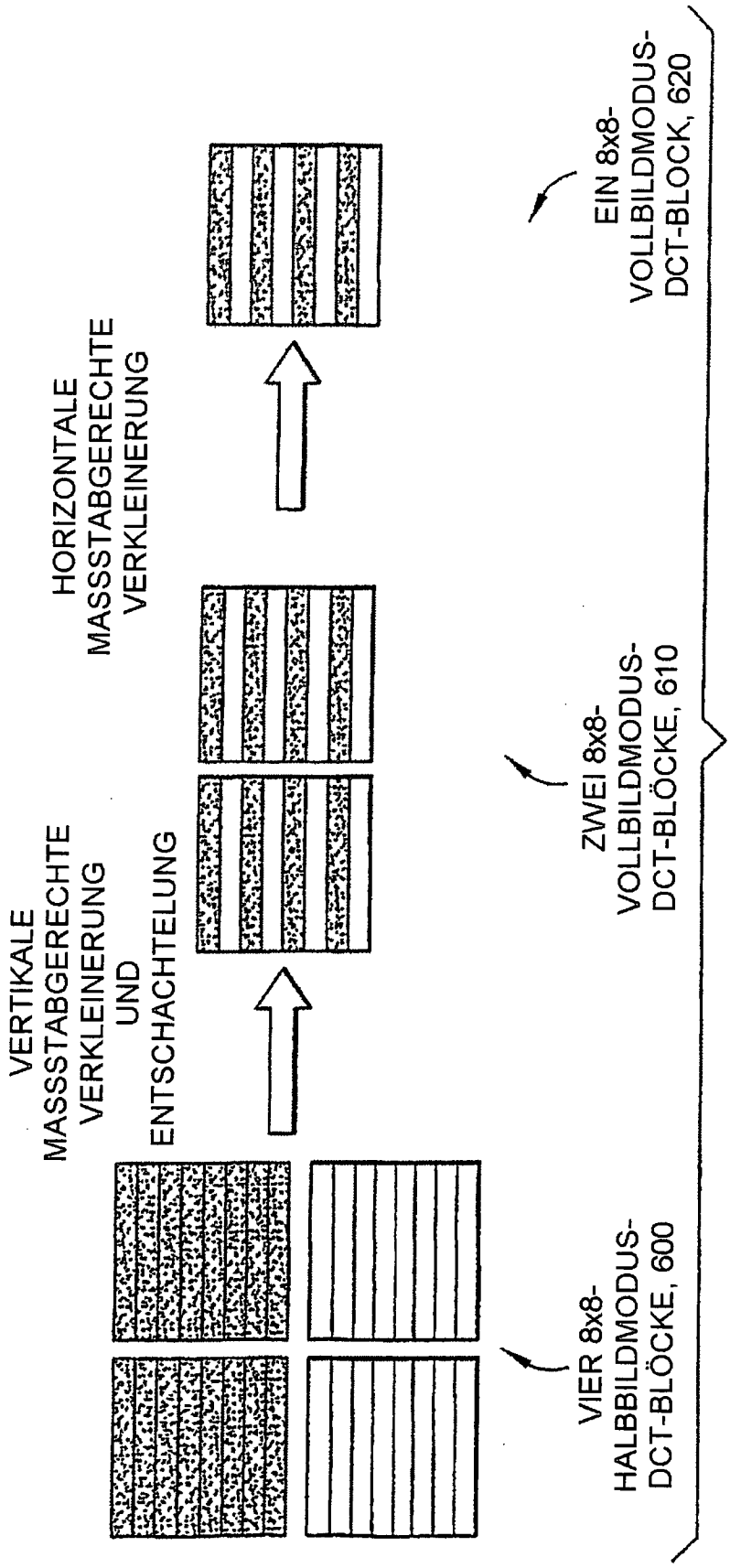
VERTIKALE MASSSTABGERECHTE VERKLEINERUNG UND ENTSCHACHTELUNG
HORIZONTALE MASSSTABGERECHTE VERKLEINERUNG
VIER 8x8-HALBBILDMODUS-DCT-BLÖCKE, 600
ZWEI 8x8-VOLLBILDMODUS-DCT-BLÖCKE, 610
EIN 8x8-VOLLBILDMODUS-DCT-BLOCK, 620

FIG.6