



(19)  
Bundesrepublik Deutschland  
Deutsches Patent- und Markenamt

(10) **DE 697 27 372 T2 2004.12.09**

(12)

## Übersetzung der europäischen Patentschrift

(97) **EP 0 895 694 B1**

(21) Deutsches Aktenzeichen: **697 27 372.5**

(86) PCT-Aktenzeichen: **PCT/US97/06933**

(96) Europäisches Aktenzeichen: **97 918 812.5**

(87) PCT-Veröffentlichungs-Nr.: **WO 97/40623**

(86) PCT-Anmeldetag: **25.04.1997**

(87) Veröffentlichungstag  
der PCT-Anmeldung: **30.10.1997**

(97) Erstveröffentlichung durch das EPA: **10.02.1999**

(97) Veröffentlichungstag  
der Patenterteilung beim EPA: **28.01.2004**

(47) Veröffentlichungstag im Patentblatt: **09.12.2004**

(51) Int Cl.<sup>7</sup>: **H04N 5/783**  
**H04N 7/173**

(30) Unionspriorität:

**639284                      25.04.1996              US**

(73) Patentinhaber:

**Sun Microsystems, Inc., Palo Alto, Calif., US;**  
**OpenTV, Inc., Mountain View, Calif., US**

(74) Vertreter:

**Dr. Weber, Dipl.-Phys. Seiffert, Dr. Lieke, 65183**  
**Wiesbaden**

(84) Benannte Vertragsstaaten:

**DE, FR, GB, IT**

(72) Erfinder:

**ZDEPSKI, Joel, Mountain View, US; KALLURI,**  
**Rama, Los Altos, US; PAGE, Howard, Burlingame,**  
**US; KAUBISCH, Wolf-Hasso, Palo Alto, US**

(54) Bezeichnung: **SYSTEM UND VERFAHREN ZUR ERZEUGUNG VON TRICKWIEDERGABE-VIDEODATENSTRÖ-  
MEN AUS EINEM KOMPRIMIERTEN NORMALWIEDERGABE-VIDEODATENSTROM**

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**

## Technisches Gebiet der Erfindung

**[0001]** Die vorliegende Erfindung betrifft im allgemeinen Video-On-Demand-Systeme und die Videokomprimierung, insbesondere, jedoch nicht ausschließlich, ein System und ein Verfahren für die Erzeugung komprimierter schneller Vorlauf- und schneller Rücklaufvideobitströme aus einem komprimierten normalen Abspielvideobitstrom.

## Beschreibung des relevanten Standes der Technik

**[0002]** Video-On-Demand-Systeme ermöglichen es einer Mehrzahl von Benutzern oder Zuschauern, selektiv Filme oder andere Audio-/Videosequenzen, die auf einem oder mehreren Videoservern oder Medienservern abgelegt sind, anzusehen. Die Videoserver sind über Datenübertragungskkanäle, wie z. B. ein Sendernetzwerk, mit der Mehrzahl von Benutzern verbunden. Beispielsweise können die Videoserver über ein Sendekabelsystem oder ein Satellitensendesystem mit einer Mehrzahl von Benutzern oder Teilnehmern verbunden sein. Die Videoserver speichern eine Mehrzahl von Filmen oder andere Audio-/Videosequenzen und jeder Benutzer kann einen oder mehrere Filme von dem Videoserver für das Ansehen auswählen. Jeder Benutzer hat einen Fernseher oder eine andere Ansichtseinrichtung, sowie eine verknüpfte Decodierlogik für das Auswählen und Ansehen der gewünschten Filme. Wenn ein Benutzer einen Film auswählt, wird der ausgewählte Film über einen der Datenübertragungskkanäle zu dem Fernseher des jeweiligen Benutzers übertragen.

**[0003]** Digitales bewegtes Video erfordert eine große Speichermenge und eine große Datenübertragungsbandbreite. Video-On-Demand-Systeme verwenden somit verschiedene Typen von Videokomprimierungsalgorithmen, um die Menge des notwendigen Speichers und der Datenübertragungsbandbreite zu reduzieren. Allgemein gibt es unterschiedliche Videokomprimierungsverfahren für stehende graphische Bilder und für bewegte Bilder. Videokomprimierungsverfahren für stillstehende graphische Bilder oder einzelne Videoeinzelsbilder werden als Intraframe- bzw. Intraeinzelsbildkomprimierungsverfahren bezeichnet und Komprimierungsverfahren für bewegte Bilder werden als Interframe- bzw. Intereinzelsbildkomprimierungsverfahren bezeichnet.

**[0004]** Beispiele der Videodatenkomprimierung für stillstehende graphische Bilder sind die RLE (Laufängencodierung) und die JPEG- (Joint Photographic Experts Group) Komprimierung. Das RLE-Komprimierungsverfahren beruht auf der Prüfung von duplizierten Pixeln in einer einzelnen Zeile der Bitmap und dem Speichern der Anzahl von nachfolgenden duplizierten Pixeln anstelle der Daten für das Pixel selbst.

Die JPEG-Komprimierung ist eine Gruppe von in Beziehung stehender Normen, die entweder verlustfreie (keine Bildqualitätsverschlechterung) oder verlustbehaftete (eine ernste Verschlechterung ist nicht wahrnehmbar) Komprimierungstypen bereitstellen. Obgleich die JPEG-Komprimierung ursprünglich für die Komprimierung von stillstehenden Bildern statt von Videobildern erdacht wurde, wird die JPEG-Komprimierung in einigen Anwendungen mit bewegtem Video verwendet.

**[0005]** Im Gegensatz zu den Komprimierungsalgorithmen für stillstehende Bilder sind die meisten Videokomprimierungsalgorithmen dafür ausgelegt, bewegte Bilder zu komprimieren. Videokomprimierungsalgorithmen für bewegtes Video verwenden ein Konzept, das als Interframekomprimierung bezeichnet wird, das das Speichern nur der Unterschiede zwischen aufeinanderfolgenden Einzelbildern in dem Datenfile beinhaltet. Die Interframekomprimierung speichert das gesamte Bild eines Schlüsselbildes oder eines Referenzbildes im allgemeinen in einem mittelgradig komprimierten Format. Nachfolgende Einzelbilder werden mit dem Schlüsselbild verglichen und nur die Unterschiede zwischen dem Schlüsselbild und den nachfolgenden Einzelbildern werden gespeichert. Periodisch, z. B. wenn neue Szenen angezeigt werden, werden neue Schlüsselbilder gespeichert und die nachfolgenden Vergleiche beginnen von diesem neuen Referenzpunkt. Es sei bemerkt, daß das Interframekomprimierungsverhältnis konstant gehalten werden kann, während die Videoqualität variiert wird. Alternativ dazu können die Interframekomprimierungsverhältnisse inhaltsabhängig sein, d. h., wenn der Videoclip, der komprimiert wird, viele abrupte Szenenübergänge von einem Bild zu einem anderen hat, ist die Komprimierung weniger effizient. Beispiele der Videokomprimierung, die eine Interframekomprimierungstechnik verwenden, sind unter anderem MPEG, DVI und Indeo.

## MPEG-Hintergrund

**[0006]** Ein Komprimierungsstandard, der als MPEG- (Moving Pictures Experts Group) Komprimierung bezeichnet wird, ist eine Gruppe von Verfahren für die Komprimierung und die Dekomprimierung von bewegten Videobildern, die die oben beschriebene Interframekomprimierungstechnik verwenden. Die MPEG-Komprimierung verwendet sowohl die Bewegungskompensations- als auch die diskrete Kosinustransformations- (DCT) Prozesse und kann zu Komprimierungsverhältnissen von mehr als 200 : 1 führen.

**[0007]** Der MPEG-Standard erfordert, daß Ton gleichzeitig mit den Videodaten aufgezeichnet wird und daß die Video- und Audiodaten in einer einzelnen Datei verzahnt (interleaved) werden, um zu versuchen, Video und Audio während der Wiedergabe

synchronisiert zu halten. Die Audiodaten werden typischerweise ebenso komprimiert und der MPEG-Standard spezifiziert ein Audiokomprimierungsverfahren, wie z. B. MPEG Layer II, ebenso durch den Philips-Handelsnamen "Musicam" bekannt.

**[0008]** In den meisten Videosequenzen bleibt der Hintergrund relativ stabil, während Aktivität im Vordergrund stattfindet. Der Hintergrund kann sich bewegen, jedoch sind große Abschnitte von aufeinanderfolgenden Einzelbildern in einer Videosequenz redundant. Bei der Erzeugung eines MPEG-Stroms erzeugt ein MPEG-Codierer I- oder Intraeinzelnbilder und P- oder vorhergesagte Einzelbilder sowie auch B-Einzelnbilder. Die I-Einzelnbilder enthalten die Videodaten für das gesamte Einzelbild des Videos und werden typischerweise alle 10 bis 15 Einzelbilder platziert. Die P-Einzelnbilder beinhalten nur Veränderungen relativ zu vorherigen I- oder P-Einzelnbildern. Sowohl I- als auch P-Einzelnbilder werden als Referenzen für nachfolgende Einzelbilder verwendet. Im allgemeinen sind für das (die) Einzelbild(er), das (die) einem I- oder P-Einzelnbild folgt bzw. folgen, d. h. Einzelbildern, die einem Referenzeinzelbild folgen, nur kleine Abschnitte dieser Einzelbilder von den entsprechenden Abschnitten des jeweiligen Referenzeinzelbildes verschieden. Somit werden für diese Einzelbilder nur die Unterschiede festgehalten, komprimiert und gespeichert.

**[0009]** Nachdem die I-Einzelnbilder erzeugt wurden, teilt der MPEG-Codierer jedes I-Einzelbild in ein Gitter von  $16 \times 16$  Pixel-Quadraten, die Makroblöcke genannt werden, auf. Das entsprechende I-Einzelbild wird in Makroblöcke unterteilt, um die Bewegungskompensierung durchzuführen. Jedes der nachfolgenden Bilder nach dem I-Einzelbild wird ebenso in dieselben Makroblöcke unterteilt. Der Codierer sucht dann nach einer exakten oder nahezu exakten Übereinstimmung zwischen dem Referenzbildmakroblock und den Makroblöcken in nachfolgenden Bildern. Wenn eine Übereinstimmung gefunden wird, überträgt der Codierer einen Vektorbewegungscode oder einen Bewegungsvektor. Der Vektorbewegungscode oder Bewegungsvektor beinhaltet nur Information über den Unterschied zwischen dem I-Einzelbild und dem entsprechenden vorhergehenden Bild. Die Blöcke in vorhergehenden Bildern, die keine Veränderung relativ zu dem Block in dem Referenzbild oder dem I-Einzelbild haben, werden ignoriert. Somit wird die Datenmenge, die tatsächlich für diese Einzelbilder gespeichert wird, erheblich reduziert.

**[0010]** Nachdem Bewegungsvektoren erzeugt wurden, verfolgt dann der Kopierer die Veränderung unter Verwendung der örtlichen Redundanz. Nach dem Auffinden der Veränderungen an dem Ort der Makroblöcke reduziert somit der MPEG-Algorithmus die Daten noch weiter durch Schreiben des Unterschiedes

des zwischen entsprechenden Makroblöcken. Dies wird verwirklicht durch einen mathematischen Prozeß, der als die diskrete Kosinustransformation oder als DCT bezeichnet wird. Dieser Prozeß unterteilt die Makroblöcke in vier Unterblöcke, wodurch Veränderungen in der Farbe und der Helligkeit herausgefunden werden. Die menschliche Wahrnehmungsfähigkeit ist empfindlicher gegenüber Helligkeitsveränderungen als gegenüber Farbveränderungen. Der MPEG-Algorithmus widmet somit mehr Anstrengung der Reduzierung des Farbraumes als dem der Helligkeit.

**[0011]** Ein MPEG-Strom beinhaltet drei Typen von Bildern, bezeichnet als das Intra- (I-) Einzelbild, das vorhergesagte (P-) Einzelbild und das bidirektional interpolierte (B-) Einzelbild. Intraeinzelnbilder bzw. Intraframes stellen Einfügepunkte in die Datei für den wahlfreien Zugriff zur Verfügung und werden im allgemeinen nur mittelgradig komprimiert. Vorhergesagte Einzelbilder werden unter Bezug auf ein vergangenes Einzelbild, d. h. ein früheres Intraeinzelnbild oder ein früheres vorhergesagtes Einzelbild codiert. Im allgemeinen erfahren vorhergesagte Einzelbilder eine ziemlich große Komprimierung und werden als Referenzen für zukünftige vorhergesagte Einzelbilder verwendet. Bidirektionale Bilder beinhalten die größte Komprimierung und erfordern sowohl eine zurückliegende als auch eine zukünftige Referenz, um codiert zu werden. Bidirektionale Einzelbilder werden niemals für Bezugnahmen für andere Einzelbilder verwendet.

**[0012]** Jedes Bild oder Einzelbild beinhaltet ebenso eine Bildkopfzeile, die das Einzelbild identifiziert und Information für dieses Einzelbild enthält. Der MPEG-Standard beinhaltet ebenso Sequenzkopfzeilen, die den Start einer Videosequenz identifizieren. Sequenzkopfzeilen werden nur einmal vor dem Beginn einer Videosequenz benötigt. Der MPEG-2-Standard erlaubt es jedoch, daß eine Sequenzkopfzeile vor jedem I-Einzelbild oder P-Einzelbild übertragen wird. Die Sequenzkopfzeile beinhaltet Informationen, die für die Videosequenz relevant sind einschließlich unter anderem der Einzelbildgeschwindigkeit und der Bildgröße.

**[0013]** MPEG-Bitströme, die in digitalen Fernsehwendungen verwendet werden, beinhalten im allgemeinen eine Sequenzkopfzeile vor jedem I-Einzelbild und P-Einzelbild. Dies ist notwendig, um das Kanalsurfen zwischen unterschiedlichen Videokanälen, was eine wichtige Benutzeranforderung ist, zu erleichtern. Im allgemeinen kann, wenn ein Benutzer zu einem neuen Kanal wechselt, das Video für den neuen Kanal nicht angezeigt werden, bis die nächste Sequenzkopfzeile in dem Bitstrom erscheint. Dies liegt daran, daß die Sequenzkopfzeile wichtige Information über die Videosequenz enthält, die von dem Decoder benötigt wird, bevor die Sequenz angezeigt

werden kann. Wenn keine Sequenzkopfzeile vor jedem I-Einzelbild und/oder P-Einzelbild enthalten wäre, dann könnte, wenn der Benutzer zu einem neuen Kanal schaltet, das Video für den neuen Kanal möglicherweise nicht sofort angezeigt werden, d. h. das Video könnte bis zur nächsten Sequenzkopfzeile nicht angezeigt werden.

**[0014]** Ein MPEG-codierter Strom beinhaltet ebenso Wichtungsmatrizen, die für die Decodierung der I-Einzelbilder in dem MPEG-Bitstrom verwendet werden. Jede Wichtungsmatrix weist eine Koeffizientenmatrix auf, die an unterschiedliche Parameter der diskreten Kosinustransformation (DCT), die bei der Codierung des Einzelbildes verwendet wird, angewendet werden. Neue Wichtungsmatrixwerte werden zu Beginn jeder Videosequenz eingefügt und diese Werte werden für die jeweiligen Einzelbilder verwendet, bis eine nachfolgende neue Wichtungsmatrix in dem MPEG-Strom erscheint. Die Wichtungsmatrizen sind typischerweise in den Sequenzkopfzeilen oder den Bildkopfzeilen aufgenommen. Die Wichtungsmatrizen können jedoch ebenso in P- oder B-Einzelbildern eingefügt sein.

**[0015]** Die ISO/IEC-MPEG-Spezifizierung ist in der ISO/IEC 13818 definiert, auf die für weitere Hintergrundinformation zurückgegriffen werden kann.

#### Trickplay-Ströme

**[0016]** In einem interaktiven Video-On-Demand-System ist es für den Benutzer sehr wünschenswert, selektiv mittels schnellem Vorlauf und/oder schnellem Rücklauf durch den betrachteten Film zu laufen. Einige Video-On-Demand-Systeme beinhalten somit für jeden Film schnelle Vorlauf- und schnelle Rücklaufströme, die als Trickplay-Ströme bezeichnet werden. Beispielsweise beschreibt die DE 195 13 772 A1 einen digitalen Videorecorder mit einer Trickplay-Möglichkeit. Ein digitaler Dateninput mit einem MPEG-2-Bitstrom wird decodiert, um Daten aus einem intracodierten Block zu extrahieren. Die decodierten Daten werden dann numerisch verarbeitet, um zu erlauben, daß alle Sequenzen in dem intracodierten Block verwendet werden, um die Daten hoher Priorität für die Hochgeschwindigkeitsanzeige zu rekonstruieren. Ein Syntaxanalyseschaltkreis wird bereitgestellt für das Analysieren des eingegebenen MPEG-2-Bitstroms und für das Erfassen eines Intraeinzelbildes, das in einen neuen Codierschaltkreis eingegeben wird, der einen Decoder und nachfolgend angeordneten Codierer aufweist.

**[0017]** Die EP 0 613 297 A2 beschreibt einen anderen digitalen Videorecorder mit Trickplay-Merkmalen. Ein Schaltkreis wird bereitgestellt, der die Videosignale hoher Auflösung verarbeitet, um ein Videosignal mit relativ niedriger Auflösung zu erzeugen, das auf dem Band mit dem Videosignal hoher Auflösung auf-

gezeichnet werden kann. Das Videosignal mit niedriger Auflösung wird unter Verwendung von Intraframe-Techniken codiert und in Segmenten mit fester Länge aufgezeichnet, die an vorbestimmten Orten auf den Bandspuren erscheinen. Segmente mit niedriger Auflösung, die Bilder in unterschiedlichen Einzelbildern repräsentieren, können wiederhergestellt werden und kombiniert werden, um ein Bild zu erzeugen, das für die Anzeige in Trickplay-Modi geeignet ist unter Verwendung der Tiefpassfilterung und des Dezimierungsschaltkreises und des Intracodierungsschaltkreises.

**[0018]** Die WO 95/28706 A1 beschreibt ein weiteres Trickplay-Verfahren und -System für MPEP-Signale. Das System weist einen Decoder und einen Trickplay-Prozessor mit einem Videorecorder auf. Die Trickplaymerkmale werden durch Tiefpassfilterung der vertikalen Auflösung auf 1/3 der Eingangsdaten erzielt.

**[0019]** Wenn der Benutzer einen schnellen Vorlauf oder schnellen Rücklauf durch einen Film wünscht, wählt der Benutzer die schnelle Vorlauf- oder schnelle Rücklaufoption aus. Der entsprechende schnelle Vorlauf- oder schnelle Rücklauf-Trickplay-Strom wird dann an dem geeigneten Punkt, wo der Benutzer den Film betrachtete, übertragen, wodurch ein schneller Vorlauf oder schneller Rücklauf durch den betrachteten Film simuliert wird.

**[0020]** Interaktive Video-On-Demand-Systeme, die Trickplay-Ströme beinhalten, erfordern Verfahren für das Erzeugen der Trickplay-Ströme aus einem normalen Abspielbitstrom. Ein gegenwärtiges Verfahren für das Erzeugen von schnellen Vorlauf- und schnellen Rücklauf-Bitströmen aus einem normalen Abspielbitstrom beinhaltet die Verwendung von Nachschlagtabellen in mehreren Strömen. Die Nachschlagtabelle beinhaltet eine Mehrzahl von Indizes, die auf jeweilige I-Einzelbilder Bezug nehmen, und der Videosever versucht, von Index zu Index mit der Übertragung zu springen und spielt nur das I-Einzelbild bei jedem Sprung ab. Mit anderen Worten indiziert der Videosever in eine Nachschlagtabelle, um nur die I-Einzelbilder für schnelle Vorlauf- und schnelle Rücklauf-Trickplay-Ströme abzuspielen. Ein Problem bei diesem Verfahren ist, daß eine beachtliche Last auf dem Videosever beim Durchführen eines Nachschlagvorgangs in einer Tabelle und beim Springen von Index zu Index mit der Übertragung, während der schnelle Vorlauf oder der schnelle Rücklauf angefordert wird, gelegt wird. Weiterhin hat dieses Verfahren verknüpfte Bitratenexpansionsprobleme.

**[0021]** Ein anderes Verfahren, das bekannt ist, um schnelle Vorlauf- und schnelle Rücklauf-Trickplay-Bitströme zu erzeugen, ist es, einen Videostrom zu erzeugen, der nicht die AC-Koeffizienten des DCT

beinhaltet, sondern nur die DC-Koeffizienten beinhaltet. Dies erzeugt einen blockhaften Trickplay-Strom und ist somit weniger wünschenswert als andere Trickplay-Strom-Erzeugungsverfahren.

**[0022]** Daher ist ein verbessertes System und Verfahren wünschenswert für die effizienten Erzeugung von Trickplay-Videoströmen, d. h. schnelle Vorlauf- und schnelle Rücklaufvideoströme aus einem komprimierten normalen Abspielbitstrom.

#### Zusammenfassung der Erfindung

**[0023]** Die vorliegende Erfindung weist ein System und ein Verfahren auf für das Erzeugen von Trickplay-Videoströmen, wie z. B. schnelle Vorlauf- und schnelle Rücklaufvideoströme, aus einem komprimierten normalen Abspielbitstrom.

**[0024]** Entsprechend einem ersten Aspekt der Erfindung wird ein Verfahren zur Verfügung gestellt für das Erzeugen von Trickplay-Strömen aus einem komprimierten normalen Abspielstrom, das aufweist: Empfangen eines komprimierten normalen Abspielbitstroms, wobei der komprimierte normale Abspielbitstrom eine Mehrzahl von intracodierten Einzelbildern und eine Mehrzahl von intercodierten Einzelbildern beinhaltet, Extrahieren der intracodierten Einzelbilder aus dem komprimierten normalen Abspielbitstrom, wobei das Extrahieren das Speichern der intracodierten Einzelbilder in einem Ablagespeicher beinhaltet, Zusammensetzen der intracodierten Einzelbilder nach dem Extrahieren, um einen zusammengesetzten Bitstrom zu bilden, Decodieren des zusammengesetzten Bitstroms, um eine Mehrzahl von nicht-komprimierten Einzelbildern zu erzeugen, und Codieren der Mehrzahl von nicht-komprimierten Einzelbildern nach der Decodierung, um einen komprimierten Trickplay-Bitstrom zu erzeugen, wobei der komprimierte Trickplay-Bitstrom nur eine Untergruppe von Einzelbildern des normalen Abspielstroms beinhaltet.

**[0025]** Gemäß einem zweiten Aspekt der Erfindung wird ein System zur Verfügung gestellt für das Erzeugen von Trickplay-Strömen aus einem komprimierten normalen Abspielbitstrom, das aufweist: ein Speichermedium, das einen komprimierten normalen Abspielbitstrom speichert, wobei der komprimierte normale Abspielbitstrom eine Mehrzahl von intracodierten Einzelbildern und eine Mehrzahl von intercodierten Einzelbildern aufweist, einen Filter für das Extrahieren der intracodierten Einzelbilder aus dem komprimierten normalen Abspielbitstrom, einen Ablagespeicher für das Speichern der extrahierten intracodierten Einzelbilder, einen Verifizierer/Fixierer für das Zusammensetzen

der gespeicherten intracodierten Einzelbilder, um einen zusammengesetzten Bitstrom zu bilden, einen Decoder, der derart betreibbar ist, daß er den zusammengesetzten Bitstrom decodiert, um eine Mehrzahl von nicht-komprimierten Einzelbildern zu erzeugen, und einen Codierer, der derart betreibbar ist, daß er die Mehrzahl von nicht-komprimierten Einzelbildern codiert, um einen komprimierten Trickplay-Bitstrom zu erzeugen, wobei der komprimierte Trickplay-Bitstrom nur eine Untergruppe von Einzelbildern des normalen Abspielbitstroms beinhaltet.

**[0026]** In einer Ausführungsform der vorliegenden Erfindung ist es möglich, komprimierte Trickplay-Videoströme effizient zu erzeugen, die reduzierte Speicher- und reduzierte Datenübertragungsbandbreitenanforderungen benötigen. Die Echtzeitverarbeitung von Videodaten, wie z. B. Indexnachschiebervorgänge, ist nicht erforderlich.

**[0027]** In einer Ausführungsform empfängt das System zuerst einen komprimierten normalen Abspielbitstrom, der entweder auf einem lokalen Medium gespeichert ist oder von einem entfernten Ort empfangen wird. Das System filtert dann den Bitstrom durch Extrahieren und Speichern von nur Teilen des Bitstroms. Das System extrahiert vorzugsweise I-Einzelbilder und Sequenzkopfzeilen einschließlich aller Wichtungsmatrizen aus dem MPEG-Bitstrom und speichert diese Information in einem oder mehreren neuen Files. Das Filtern entfernt oder löscht somit Abschnitte des MPEG-Datenstroms einschließlich vorhergesagter (P-) Einzelbilder und bidirektionaler (B-) Einzelbilder.

**[0028]** Das System setzt dann die gefilterten Daten zusammen oder vereinigt diese in Vorwärts- oder Rückwärtsordnung, um einen einzelnen zusammengesetzten Bitstrom zu erzeugen. Das System stellt ebenso sicher, daß die Wichtungsmatrizen geeignet den jeweiligen I-Einzelbildern entsprechen. Für einen schnellen Vorlaufs-Trickplay-Strom weist der zusammengesetzte Bitstrom die Sequenzkopfzeilen, die I-Einzelbilder und entsprechende Wichtungsmatrizen in der geeigneten Zeit oder Sequenzordnung auf, so wie sie in dem ursprünglichen MPEG-Strom erscheinen. Für einen schnellen Rücklauf-Trickplay-Bitstrom kehrt das System die Anordnung der Kopfzeilen-/I-Frame-Gruppierungen oder -Tupel um, um einen Rücklaufstrom zu erzeugen. Dies erzeugt einen zusammengesetzten Bitstrom, der eine Mehrzahl der Sequenzkopfzeilen und I-Einzelbilder einschließlich der verknüpften Wichtungsmatrizen aufweist.

**[0029]** Der zusammengesetzte Bitstrom wird dann MPEG-2-codiert, um einen neuen Videostrom zu erzeugen. Die neue Videosequenz weist nur eines von X Bildern oder Einzelbildern des ursprünglichen nicht-komprimierten normalen Abspielbitstroms auf,

wobei  $1/X$  die Frequenz der I-Einzelbilder in dem ursprünglichen komprimierten normalen Abspielbitstrom ist. Der Ausgangsbildstrom wird dann erneut codiert mit den MPEG-Parametern, die für den Trickplay-Strom gewünscht werden, wodurch ein Trickplay-Strom erzeugt wird, der ein gültiger MPEG-codierter Strom ist. Wenn dieser neue MPEG-codierte Trickplay-Strom decodiert wird, wird das Ergebnis eine schnelle Vorlauf- oder schnelle Rücklauf-Videossequenz sein, die nur eines von jeweils  $X$  Einzelbildern des ursprünglichen nicht-komprimierten normalen Abspielbitstroms enthält.

**[0030]** Daher ist es mit der oben beschriebenen Ausführungsform möglich, Trickplay-Ströme aus einem komprimierten normalen Abspielbitstrom effizienter zu erzeugen. Der resultierende Trickplay-Strom ist ein gültiger MPEG-codierter Strom und hat somit reduzierte Speicher- und Datenübertragungsbandbreitenanforderungen und dieser Trickplay-Strom kann mit bekanntem Verhalten auf jedem MPEG-Decodierer decodiert werden.

#### Kurze Beschreibung der Figuren

**[0031]** Ein besseres Verständnis der vorliegenden Erfindung kann erreicht werden, wenn die folgende detaillierte Beschreibung bevorzugten Ausführungsform in Verbindung mit den folgenden Zeichnungen betrachtet wird, in denen:

**[0032]** Fig. 1 ein Computersystem darstellt, das Video-Trickplay-Ströme gemäß einer Ausführungsform der vorliegenden Erfindung erzeugt,

**[0033]** Fig. 1A ein Blockdiagramm ist, das das Computersystem von Fig. 1 darstellt,

**[0034]** Fig. 2 ein Flußdiagramm ist, das den Betrieb bzw. Ablauf einer Ausführungsform der vorliegenden Erfindung darstellt,

**[0035]** Fig. 3 ein Flußdiagramm ist, das den Ablauf des Filters von Fig. 2 darstellt,

**[0036]** Fig. 4 ein Flußdiagramm ist, das den Betrieb des Verifizierers/Fixierers von Fig. 2 darstellt,

**[0037]** Fig. 5 ein Flußdiagramm ist, das den Betrieb der vorliegenden Erfindung gemäß einer alternativen Ausführungsform darstellt, und

**[0038]** Fig. 6A–6C Flußdiagramme einer bevorzugten Ausführungsform für das Erzeugen eines Trickplay-Rücklaufstroms gemäß einer Ausführungsform der vorliegenden Erfindung sind.

#### Detaillierte Beschreibung der Ausführungsformen

**[0039]** In Fig. 1 ist ein System für das Erzeugen von

Trickplay-Videoströmen aus einem komprimierten normalen Abspielbitstrom gezeigt. Das System erzeugt vorzugsweise Trickplay-Ströme für die Verwendung in Video-On-Demand-Systemen. Das System kann jedoch verwendet werden, um Trickplay-Ströme für die Verwendung in irgendeiner von verschiedenen Anwendungstypen, so wie es gewünscht ist, zu erzeugen.

**[0040]** Wie gezeigt, weist in einer Ausführungsform das Trickplay-Erzeugungssystem ein Allzweckcomputersystem **60** auf. Das Computersystem **60** empfängt einen komprimierten normalen Abspielbitstrom und erzeugt einen oder mehrere Trickplay-Ströme. In der vorliegenden Beschreibung bezieht sich der Begriff "Trickplay-Ströme" auf schnelle Vorlauf- und/oder schnelle Rücklauf-Videoströme, vorzugsweise komprimierte Ströme, die aus einem normalen Abspielbitstrom erzeugt werden, und die vorzugsweise aus einem komprimierten normalen Abspielbitstrom erzeugt werden.

**[0041]** Das Computersystem **60** beinhaltet vorzugsweise verschiedene Standardkomponenten einschließlich eines oder mehrerer Prozessoren oder mehrerer Busse, eines Festplattenlaufwerks und Speicher. In Fig. 1A ist ein Blockdiagramm gezeigt, das die Komponenten veranschaulicht, die in dem Computersystem von Fig. 1 enthalten sind. Es sei bemerkt, daß Fig. 1A lediglich anschaulich ist und andere Computerarchitekturen verwendet werden können, so wie es gewünscht wird. Das Computersystem beinhaltet, wie gezeigt, zumindest einen Prozessor **80**, der über die Chip-Set-Logik **82** mit einem Systemspeicher **84** verbunden ist. Der Chipsatz **82** beinhaltet vorzugsweise eine PCI- (Peripheral Component Interconnect) Brücke für das Bilden einer Schnittstelle zu dem PCI-Bus **86**. Der MPEG-Decodierer **74** und der MPEG-Codierer **76** sind als mit dem PCI-Bus **86** verbunden gezeigt. In einer alternativen Ausführungsform wird die MPEG-Decodierung und -Codierung in Software durchgeführt. Verschiedene andere Komponenten können in dem Computersystem vorhanden sein, wie z. B. ein Video **88** und eine Festplatte **99**.

**[0042]** In Fig. 1 beinhaltet in der bevorzugten Ausführungsform das Computersystem **60** einen oder mehrere digitale Speicher oder Medienspeichereinrichtungen oder ist damit verbunden. Beispielsweise ist in der Ausführungsform von Fig. 1 das Computersystem **60** mit einer Medienspeichereinheit **62** über ein Kabel **64** verbunden. Die Medienspeichereinheit **62** beinhaltet ein oder mehrere CD-ROM-Laufwerke und/oder ein oder mehrere digitale Videodisk- (DVD-) Speichereinheiten für das Speichern von digitalem Video. Das Computersystem kann ebenso ein oder mehrere interne CD-ROM-Laufwerke beinhalten oder kann mit einem oder mehreren getrennten digitalen Videodisk(DVD-) Speichereinheiten verbunden

sein. Das Computersystem **60** kann ebenso mit anderen Typen von digitalen oder analogen Speichereinrichtungen verbunden sein, so wie dies gewünscht wird.

**[0043]** Der komprimierte normale Abspielbitstrom kann in einem Speichermedium, wie z. B. einer CD-ROM oder einer digitalen Videodisk (DVD), enthalten sein. In dieser Ausführungsform wird das Speichermedium, das den komprimierten normalen Abspielbitstrom beinhaltet, in die entsprechende Speichereinrichtung eingesetzt, die das Computersystem **60** aufweist oder mit dem sie verbunden ist und das Computersystem **60** liest den komprimierten normalen Abspielbitstrom aus dem Speichermedium. Der komprimierte normale Abspielbitstrom kann beispielsweise auf einer CD-ROM enthalten sein und die CD-ROM wird in das CD-ROM-Laufwerk der Medienspeichereinheit **62** oder des Computersystems **60** eingelegt, um es dem Computersystem **60** zu ermöglichen, auf den komprimierten normalen Abspielbitstrom zuzugreifen. Ebenso kann der komprimierte normale Abspielbitstrom auf einer DVD enthalten sein und der Bitstrom wird von dem Computersystem **60** von der DVD gelesen.

**[0044]** Alternativ kann der komprimierte normale Abspielbitstrom von einer externen Quelle empfangen werden, wie z. B. einer entfernten Speichereinrichtung oder einem entfernten Computersystem. In dieser Ausführungsform beinhaltet das Computersystem vorzugsweise eine Eingabeeinrichtung, wie z. B. eine ATM- (Asynchronous Transfer Mode) Adapterkarte und einen ISDN- (Integrated Services Digital Network) Terminaladapter oder einen anderen digitalen Datenempfänger für das Empfangen des komprimierten normalen Abspielbitstroms. Der komprimierte normale Abspielbitstrom kann ebenso in analogem Format gespeichert sein oder empfangen werden und in digitale Daten umgewandelt werden entweder außerhalb des Computersystems **60** oder innerhalb des Computersystems **60**.

**[0045]** Wie oben erwähnt wurde, erzeugt das Computersystem **60** Trickplay-Videoströme aus einem komprimierten normalen Abspielbitstrom. Wie weiter unten erörtert wird, führt das Computersystem **60** Filter- und Verifizier-/Fixierfunktionen sowie MPEG-2-Decodier- und -Codierfunktionen durch. In der bevorzugten Ausführungsform werden die Filter- und Verifizier-/Fixierfunktionen von dem Computersystem **60** mittels Software durchgeführt, wobei die Software durch die Disketten **72** dargestellt wird. In einer alternativen Ausführungsform beinhaltet das Computersystem **60** spezielle Hardware, die Filterungs- und/oder Verifizier-/Fixierfunktionen durchführt.

**[0046]** In der Ausführungsform von **Fig. 1** beinhaltet das Computersystem **60** vorzugsweise als Hardware

eine MPEG- (MPEG-2-) Decoderkarte **74** und als Hardware eine MPEG- (MPEG-2-) Codierkarte **76**. Der MPEG-Decodierer **74** und der MPEG-Codierer **76** weisen Adapterkarten auf, die mit einem Bus in dem Computersystem verbunden sind, sind jedoch aus anschaulichen Gründen außerhalb des Computersystems **60** gezeigt. Alternativ befindet sich der MPEG-Decodierer und/oder der MPEG-Codierer außerhalb des Computersystems **60**. In einer alternativen Ausführungsform führt das Computersystem **60** die MPEG-Dekomprimierung und/oder die MPEG-Komprimierung mittels Software durch, wobei die Software durch die Disketten **72** dargestellt wird. In dieser Ausführungsform beinhaltet das Computersystem **60** keinen MPEG-Decoder oder MPEG-Codierer als Hardware.

**[0047]** Es sei bemerkt, daß das System für das Erzeugen von Trickplay-Videoströmen zwei oder mehrere miteinander verbundene Computer aufweisen kann, sofern dies gewünscht wird. Das System für das Erzeugen von Trickplay-Videoströmen kann ebenso spezielle Hardware aufweisen, die entweder allein oder in Verbindung mit einem allzweckprogrammierbaren Computer verwendet wird. Es sei bemerkt, daß irgendeiner von verschiedenen Systemtypen verwendet werden kann für das Erzeugen von Trickplay-Videoströmen gemäß verschiedener Ausführungsformen der vorliegenden Erfindung, so wie dies gewünscht wird.

#### Flußdiagramm

**[0048]** In **Fig. 2** ist ein Diagramm, das den Betrieb einer Ausführungsform der vorliegenden Erfindung darstellt, gezeigt. Wie gezeigt, beinhalten die Funktionen eine Filteroperation **102**, eine Verifizier-/Fixieroperation **104**, eine MPEG-2-Decodieroperation und eine MPEG-2-Codieroperation. Wie oben erörtert wurde, kann jede dieser Operationen entweder in Hardware oder in Software durchgeführt werden, wie dies gewünscht wird.

**[0049]** Wie gezeigt, empfängt das System einen normalen Abspielbitstrom. Der normale Abspielbitstrom ist ein Bitstrom aus Videodaten, der verwendet wird, um eine Videosequenz darzustellen, wie z. B. ein Fernsehsegment oder einen Film auf einem Schirm, wie z. B. einem Fernseher oder einem Computersystem. In der bevorzugten Ausführungsform ist der normale Abspielbitstrom ein komprimierter Bitstrom. Der komprimierte normale Abspielbitstrom ist vorzugsweise ein MPEG-komprimierter Bitstrom, vorzugsweise ein MPEG-1- oder MPEG-2-komprimierter Bitstrom. Andere Typen der Komprimierung können verwendet werden, sofern dies gewünscht wird.

**[0050]** Wie gezeigt, beinhaltet das System einen Filter **102**, der als der "I-Bild-Sequenzkopfzeilenfilter"

bezeichnet wird, der den komprimierten Bitstrom filtert. Der Filter **102** speichert nur Abschnitte des Bitstroms, d. h. er entfernt oder löscht Abschnitte des MPEG-Datenstroms. Genauer gesagt extrahiert der Filter **102** die I-Frames und die Sequenzkopfzeilen sowie auch alle Gewichtungsmatrizen aus dem MPEG-Bitstrom und speichert diese Information in einer neuen Datei. Somit entfernt der Filter **102** alle MPEG-Videodaten außer den I-Frames, den Sequenzkopfzeilen und den Gewichtungsmatrizen. Der Filter **102** löscht somit Teile des MPEG-Datenstroms einschließlich der vorhergesagten (P-) Einzelbilder und der bidirektionalen (B-) Einzelbilder.

**[0051]** Wie oben erörtert wurde, beinhaltet ein MPEG-codierter Bitstrom eine Mehrzahl von I-Frames, die intracodierte Bilder sind, und eine Mehrzahl von B- und P-Frames, die intercodierte Einzelbilder sind. Die I-Frames enthalten jeweils Videodaten für ein gesamtes Videoeinzelbild und werden periodisch in der Abfolge platziert. Die P- und B-Frames beinhalten Veränderungsinformation relativ zu vorherigen oder nachfolgenden Einzelbildern. Jedes Bild oder Einzelbild beinhaltet ebenso eine Bildkopfzeile, die das Einzelbild identifiziert und Informationen für dieses Einzelbild beinhaltet. Ein MPEG-codierter Bitstrom beinhaltet weiterhin eine oder mehrere Sequenzkopfzeilen, die bestimmte Information beinhalten betreffend die Videosequenz einschließlich unter anderem der Einzelbildgeschwindigkeit und der Bildgröße.

**[0052]** Ein MPEG-codierter Strom beinhaltet ebenso Gewichtungsmatrizen, die für die Rekonstruktion der Pixelwerte aus den DCT-Koeffizienten in dem MPEG-Bitstrom verwendet werden. Jede Gewichtungsmatrix weist eine Koeffizientenmatrix auf, die an verschiedene Parameter der diskreten Kosinustransformation (DCT) angewendet werden, die bei der Codierung des Einzelbildes verwendet wird. Die Matrizen in dem Decodierer werden zu Beginn jeder Videosequenz reinitialisiert und diese Werte werden für die entsprechenden Einzelbilder verwendet, bis eine nachfolgende neue Gewichtungsmatrix in dem MPEG-Strom erscheint. Es sei bemerkt, daß ein MPEG-codierter Strom sowohl Interframe- als auch Intraframematrizen beinhaltet. Das Trickplay-Erzeugungssystem und -verfahren verwendet nur die Intraframematrizen bei der Erzeugung der Trickplay-Ströme.

**[0053]** Die Gewichtungsmatrizen sind typischerweise in der Bildkopfzeile für jedes I-Frame oder in einer Sequenzkopfzeile vor dem jeweiligen I-Frame enthalten. In manchen Fällen kann jedoch die Wichtungsmatrix für ein bestimmtes I-Frame nicht in der I-Frame-Bildkopfzeile oder in einer Sequenzkopfzeile enthalten sein, sondern kann stattdessen in einem vorherigen P- oder B-Frame enthalten sein. Mit anderen Worten können in manchen Fällen die neuen Werte

einer Wichtungsmatrix für ein bestimmtes I-Frame in einem vorherigen P- oder B-Frame enthalten sein. Dies tritt auf, wo das P- oder B-Frame einen oder mehrere Makroblöcke beinhaltet, die mit einer I-Frame-Syntax codiert sind. Der Filter **102** untersucht somit die P- und B-Frames nach Wichtungsmatrizen und speichert diese Matrizen für die Verwendung durch den Verifizier-/Fixierblock **104**.

**[0054]** Wie gezeigt, stellt der Filter **102** einen gespeicherten Ausgang, der Abschnitt des MPEG-Stroms aufweist, einem Verifizierer/Fixierer **104** zur Verfügung. Der Verifizierer/Fixierer **104** sammelt die Daten, die von dem Filter **102** erzeugt wurden, oder setzt sie in einem einzelnen Bitstrom zusammen. Der Verifizierer/Fixierer **104** sammelt oder verkettet die gespeicherten Daten in die geeignete Ordnung, um einen zusammengesetzten Bitstrom zu erzeugen. Der Verifizierer/Fixierer **104** verwendet Informationen, die von dem Filter **102** bereitgestellt werden, um sicherzustellen, daß die Sequenzkopfzeilen den geeigneten I-Einzelbildern entsprechen.

**[0055]** Der Verifizierer/Fixierer **104** stellt ebenso sicher, daß die Gewichtungsmatrizen, die in dem Strom gefunden werden, wie z. B. in einem P- oder B-Einzelbild in dem verknüpften Strom aufgenommen werden und korrekt den jeweiligen I-Frames entsprechen. Mit anderen Worten stellt der Verifizierer/Fixierer **104** ebenso sicher, daß die Gewichtungsmatrizen oder Quantisierungsmatrix-Veränderungen korrekt in dem neuen zusammengesetzten Bitstrom aufgenommen werden. In der bevorzugten Ausführungsform erzeugt der Verifizierer/Fixierer **104** neue Sequenzkopfzeilen für Wichtungsmatrizen, die in P- und B-Einzelbildern gefunden wurden, und verkettet diese neuen Sequenzkopfzeilen mit den korrekten I-Frames.

**[0056]** Für einen schnellen Vorlauf-Trickplay-Strom weist der gesammelte Bitstrom die Sequenzkopfzeilen, die I-Frames und entsprechende Wichtungsmatrizen in der richtigen Zeit- oder Abfolgeordnung auf, wie in dem ursprünglichen MPEG-Strom erschienen. Für eine schnelle Rücklauf-Trickplay-Bitstromsequenz kehrt der Verifizierer/Fixierer **104** die Ordnung der Sequenzkopfzeile/I-Frame-Gruppierungen oder -Tupel um, um eine rückwärts ablaufende Sequenz zu erzeugen. Somit ordnet der Verifizierer/Fixierer **104** ebenso die Sequenzkopfzeile/I-Frame-Tupel in einer umgekehrten Ordnung neu, um sicherzustellen, daß die Matrizen den jeweiligen I-Frames entsprechen.

**[0057]** Der Ausgang des Verifizierers/Fixierers **104** ist somit ein zusammengesetzter Bitstrom, der eine Mehrzahl von Sequenzkopfzeilen und I-Frames aufweist. Der Verifizierer/Fixierer **104** erzeugt somit einen resultierenden Bitstrom, der ein zulässiger MPEG-codierter Bitstrom ist.

**[0058]** Dieser zusammengesetzte Ausgangsbitstrom wird einem Decoderblock **106**, vorzugsweise einem MPEG-2-Decoderblock **106**, zur Verfügung gestellt. Der MPEG-2-Decoderblock **106** decodiert den zusammengesetzten Bitstrom, d. h. jedes der jeweiligen I-Frames, um eine neue Videosequenz zu erzeugen. Die neue Videosequenz ist eine nicht-komprimierte Sequenz und weist nur eines von jeweils X Bildern des ursprünglichen nicht-komprimierten normalen Abspielstroms auf. Wenn somit der ursprüngliche normale Abspielbitstrom I-Einzelbilder mit einer Frequenz von 1 je X Einzelbilder beinhaltet, weist die neue Videosequenz nur eines von jeweils X Bildern oder Einzelbildern des ursprünglichen nicht-komprimierten normalen Abspielbitstroms auf. Wenn beispielsweise der ursprüngliche MPEG-2-komprimierte Bitstrom, der am Eingang des Filters empfangen wird, ein I-Einzelbild alle sieben Einzelbilder beinhaltet, erzeugt der MPEG-2-Decoderblock **106** einen Bitstrom, der nicht-komprimierte Videodaten aufweist, jedoch nur eines von jeweils sieben Einzelbildern des ursprünglichen nicht-komprimierten Bitstroms beinhaltet.

**[0059]** Der Ausgangsbitstrom wird dann zu einem Codierblock **108** geliefert, wo der Strom erneut mit den jeweiligen MPEG-Parametern, die für den Trickplay-Strom gewünscht werden, codiert wird. Die MPEG-Parameter beinhalten unter anderem die Bitrate und die Bildgröße. In der bevorzugten Ausführungsform codiert der Codierblock **108** den Strom mit einer kleineren Bildgröße und einer niedrigeren Datenrate als den normalen Abspielstrom für eine reduzierte Datenspeicherung und eine reduzierte Übertragungsbandbreite.

**[0060]** Der MPEG-2-Codierer **108** erzeugt einen Trickplay-Strom, der ein gültiger MPEG-codierter Strom ist, der jedoch nur jedes X-te Einzelbild des ursprünglichen Stroms beinhaltet. Somit beinhaltet der Trickplay-Ausgangsstrom von dem Codierer **108** I-Frames, P-Frames und B-Frames. Wenn dieser neue MPEG-codierte Trickplay-Strom nachfolgend an einen Benutzer übertragen wird und von einem MPEG-Decoder decodiert wird, weist der resultierende Bitstrom einen Strom auf, der nur eines von jeweils X Einzelbildern des ursprünglichen nicht-komprimierten normalen Abspielbitstroms beinhaltet.

**Fig. 3 - Filterflußdiagramm**

**[0061]** In **Fig. 3** stellt ein Flußdiagramm die Funktion des Filters **102** gemäß einer Ausführungsform dar. In Schritt **202** untersucht der Filter **102** einen Block der MPEG-Daten. Hier wird angenommen, daß der MPEG-Datenblock entweder eine Sequenzkopfzeile oder eine Bildkopfzeile ist. Falls die untersuchten MPEG-Daten eine Sequenzkopfzeile sind, speichert der Filter **102** in Schritt **212** die Sequenzkopfzeile. Der Filter **102** kehrt dann zu Schritt **202** zurück und

beginnt mit der Untersuchung des nächsten Blockes der MPEG-Daten.

**[0062]** Falls das Einzelbild oder die Daten, die in Schritt **202** untersucht werden, eine Bildkopfzeile oder ein Bild ist, untersucht in Schritt **214** der Filter **102** die Bildkopfzeile und das nachfolgende Einzelbild, das zu der Bildkopfzeile korrespondiert. Falls das Einzelbild oder die Daten, die untersucht werden, ein I-Frame ist, speichert der Filter **102** in Schritt **222** die Bildkopfzeile und das I-Frame. Der Filter **102** speichert ebenso vorzugsweise Korrespondenzdaten in Schritt **224**, die die jeweilige Sequenzkopfzeile oder die Bildkopfzeile anzeigen, die zu dem I-Frame korrespondieren, das gespeichert wird. Nach dem Speichern des I-Frames und der Korrespondenzdaten kehrt der Filter **102** zu Schritt **202** zurück, um den nächsten Block der MPEG-Daten zu untersuchen.

**[0063]** Falls das Einzelbild, das untersucht wird, ein P- oder B-Frame ist, bestimmt der Filter **102** in Schritt **232**, ob das jeweilige P- oder B-Frame eine Wichtungsmatrix beinhaltet. Falls nicht, werden dann die P- oder B-Frame-Daten nicht gespeichert und der Filter **102** kehrt zu Schritt **202** zurück, um den nächsten Block der MPEG-Daten zu untersuchen. Falls das P- oder B-Frame, das untersucht wird, eine Wichtungsmatrix beinhaltet, dann speichert in Schritt **234** der Filter **102** die Wichtungsmatrix. In Schritt **236** verknüpft der Filter **102** die Wichtungsmatrix mit dem jeweiligen I-Frame, d. h. dem nachfolgenden I-Frame. In Schritt **236** speichert der Filter **102** ebenso vorzugsweise Korrespondenzdaten, die den jeweiligen I-Frame anzeigen, der zu der Wichtungsmatrix korrespondiert. Nachdem der Filter **102** die Wichtungsmatrix und die Korrespondenzdaten speichert, kehrt der Filter **102** zu Schritt **202** zurück, um den nächsten Block der MPEG-Daten zu untersuchen.

**[0064]** Somit untersucht der Filter **102** alle Kopfzeilen und Einzelbilder in einem MPEG-Strom. Dies ist notwendig, da eine Wichtungsmatrix in jeder Kopfzeile oder jedem Einzelbild in dem MPEG-Strom auftreten kann. Der Filter **102** speichert nur Sequenzkopfzeilen und I-Frames sowie Wichtungsmatrizen, die irgendwo in dem MPEG-Strom lokalisiert werden, wie z. B. in P- oder B-Frames. Der Filter **102** speichert keine P-Frame- oder B-Frame-Daten. Der Filter **102** verknüpft ebenso Sequenzkopfzeilen und Wichtungsmatrizen mit den korrespondierenden I-Frames.

**[0065]** Wenn somit eine Matrix in einem der dazwischenliegenden P- oder B-Frames beinhaltet ist, speichert der Filter **102** diese Matrix in einer Datei für die Verwendung, wenn die Trickplay-Mastersequenz rekonstruiert wird. Wie weiter unten erörtert wird, erzeugt die vorliegende Erfindung während der Konstruktion des Trickplay-Stroms eine Dummysequenzkopfzeile oder fügt die neue Matrix auf andere Art in

den zusammengesetzten Bitstrom ein.

**[0066]** Nachdem eine Sequenzkopfzeile oder ein Einzelbild untersucht wurde, untersucht der Filter **102** das nächste Stück der MPEG-Daten und wiederholt diese Operation. In gleicher Weise untersucht der Filter **102** jede Kopfzeile und jedes Einzelbild in der MPEG-Sequenz, speichert die Sequenzkopfzeilen, die I-Frames und die Wichtungsmatrizen und speichert die P- und B-Einzelbilder nicht. Somit speichert der Filter **102** einen Teil des MPEG-Datenstroms, der nur Sequenzkopfzeilen, Bildkopfzeilen, I-Frames und Wichtungsmatrizen aufweist.

**[0067]** In einer Ausführungsform werden die Ergebnisse des Filters **102** aneinandergehängt, um den Trickplay-Bitstrom zu erzeugen. Es sei jedoch bemerkt, daß, wenn die Ergebnisse des Filters **102** einfach aneinandergehängt werden, der Bitstrom kein gültiger MPEG-Bitstrom sein wird. In der bevorzugten Ausführungsform erzeugt die vorliegende Erfindung einen MPEG-komprimierten Bitstrom, da es wünschenswert ist, daß der resultierende Trickplay-Bitstrom durch einen Standard-MPEG-Decodierer geleitet werden kann.

**Fig. 4** – Verifizierer/Fixierer-Flußdiagramm

**[0068]** In **Fig. 4** ist ein Flußdiagramm gezeigt, das die Funktion des Verifizierers/Fixierers **104**, der einen Trickplay-Strom zusammensetzt, illustriert. In Schritt **302** untersucht der Verifizierer/Fixierer **104** einen gespeicherten Block von MPEG-Daten, d. h. einen Block von MPEG-Daten, die von dem Filter **102** gespeichert sind. Hier wird bemerkt, daß der Block der MPEG-Daten entweder eine Sequenzkopfzeile, eine Bildkopfzeile, ein I-Einzelbild oder eine Wichtungsmatrix ist. Falls in Schritt **302** die MPEG-Daten, die von dem Verifizierer/Fixierer **104** untersucht werden, eine Sequenzkopfzeile, eine Bildkopfzeile oder ein I-Einzelbild ist, untersucht in Schritt **304** der Verifizierer/Fixierer **104** die entsprechenden Daten, die von dem Filter **102** erzeugt werden, um die Bildkopfzeile oder die Sequenzkopfzeile mit dem geeigneten I-Frame zu verknüpfen. In Schritt **306** gruppiert der Verifizierer/Fixierer **104** die I-Frames und ihre jeweiligen Bildkopfzeilen und/oder Sequenzkopfzeilen oder hängt sie aneinander, wobei diese Gruppierungen als Kopfzeilen/I-Frame-Tupel bezeichnet werden. In Schritt **308** hängt der Verifizierer/Fixierer **104** das neue Kopfzeilen/I-Frame-Tupel an den Bitstrom, der zusammengesetzt wird, an, um den neuen Bitstrom zusammenzusetzen. Nachdem Schritt **308** durchgeführt ist, kehrt der Verifizierer/Fixierer **104** zu Schritt **302** zurück, um den nächsten gespeicherten Block der MPEG-Daten zu untersuchen und die Funktion wiederholt sich.

**[0069]** Wenn das System einen schnellen Vorlaufstrom erzeugt, dann hängt in Schritt **308** der Verifizie-

rer/Fixierer **104** die Gruppierungen der I-Frames und ihre jeweiligen Bildkopfzeilen und/oder Sequenzkopfzeilen, die als Kopfzeilen/I-Frame-Tupel bezeichnet werden, in einer Vorwärtszeitrichtung aneinander, d. h. in der Zeitordnung, in der sie in dem ursprünglichen Strom erscheinen. Falls das System einen schnellen Rücklaufstrom erzeugt, hängt in Schritt **308** der Verifizierer/Fixierer **104** die Kopfzeilen/I-Frame-Tupel in einer umgekehrten Zeitordnung aneinander. Es wird somit bemerkt, daß für eine schnelle Rücklaufsequenz die Bildkopfzeilen und die Sequenzkopfzeilen, die den jeweiligen I-Einzelbildern entsprechen, immer noch zeitlich vor ihren jeweiligen I-Einzelbildern aneinandergehängt werden. Die Kopfzeilen/I-Frame-Gruppierungen oder -Tupel werden jedoch in umgekehrter zeitlicher Ordnung aneinandergehängt.

**[0070]** Falls in Schritt **302** die von dem Verifizierer/Fixierer **104** untersuchten MPEG-Daten eine Wichtungsmatrix aufweisen, erzeugt in Schritt **312** der Verifizierer/Fixierer **104** vorzugsweise eine neue Sequenzkopfzeile, die die Wichtungsmatrix aufweist. Der Verifizierer/Fixierer **104** setzt dann zu Schritt **304** fort, wo der Verifizierer/Fixierer **104** die Korrespondenzdaten, die von dem Filter **102** erzeugt wurden, untersucht, und gruppiert in Schritt **306** die neue Sequenzkopfzeile mit dem entsprechenden I-Einzelbild. In Schritt **306** plaziert der Verifizierer/Fixierer **104** die neu erzeugte Sequenzkopfzeile vor dem jeweiligen I-Frame. In Schritt **308** hängt der Verifizierer/Fixierer **104** das neue Sequenzkopfzeilen-/I-Frame-Tupel an den Bitstrom, der zusammengesetzt wird, an, um den neuen Bitstrom zusammenzusetzen.

**[0071]** Falls ein I-Frame keine Wichtungsmatrix hat, verwendet der Verifizierer/Fixierer **104** die Wichtungsmatrix von einem vorherigen I-Frame oder einen voreingestellten Wert. In der bevorzugten Ausführungsform wird die voreingestellte Matrix, die verwendet wird, implizit aufgrund der Tatsache, daß der Strom nicht explizit eine Matrix enthält, bestimmt, und diese Operation ist Teil des MPEG-Standards.

#### Alternative Ausführungsformen

**[0072]** In einer ersten alternativen Ausführungsform der Erfindung beinhaltet für schnelle Vorlauf-Trickplay-Ströme das Trickplay-Erzeugungssystem P-Frames sowie I-Frames in dem schnellen Vorlauf-Trickplay-Strom. Diese Ausführungsform wird vorzugsweise verwendet für 2X, 3X oder 4X schnelle Vorlauf-Trickplay-Ströme, vorzugsweise 3X schnelle Vorlaufströme. In dieser Ausführungsform untersucht das System den MPEG-Strom nach Interframematrizen in B-Frames und bewegt diese Interframematrizen zu nachfolgenden P-Frames. Es sei bemerkt, daß diese Ausführungsform nur für Vorwärts-Trickplay-Ströme möglich ist, da P-Frames in dem ursprünglichen MPEG-Strom nur Veränderungen rela-

tiv zu vorherigen P-Frames aufweisen.

**[0073]** In **Fig. 5** ist ein Diagramm gezeigt, das die Operation gemäß einer anderen alternativen Ausführungsform darstellt. Wie gezeigt, beinhaltet diese Ausführungsform der vorliegenden Erfindung eine MPEG-2-Decodieroperation **502**, eine Extraktionsoperation **504** und eine MPEG-2-Codieroperation **506**. Jede dieser Operationen kann entweder in Hardware oder Software durchgeführt werden, wie dies gewünscht wird.

**[0074]** Wie gezeigt, empfängt das System einen normalen Abspielbitstrom, vorzugsweise einen MPEG-komprimierten normalen Abspielbitstrom. Dieser komprimierte normale Abspielbitstrom wird einem Decoderblock **502**, vorzugsweise einem MPEG-2-Decoderblock **502** bereitgestellt. Der MPEG-2-Decoderblock **502** decodiert jedes der jeweiligen Einzelbilder, um die ursprüngliche nichtkomprimierte Videosequenz zu erzeugen. Die ursprüngliche nicht-komprimierte Videosequenz wird dann einem Extraktionsblock **504** bereitgestellt, der derart arbeitet, daß er eines von jeweils X Einzelbildern der Videosequenz extrahiert. Der Extraktionsblock **504** hängt ebenso diese extrahierten Einzelbilder in entweder der Vorwärts- oder der Rückwärtszeitordnung aneinander. Diese aneinandergehängten Einzelbilder weisen somit einen Bitstrom auf, der nur eines von jeweils X Bildern des ursprünglichen nicht-komprimierten normalen Abspielbitstroms aufweist. Wenn der ursprüngliche normale Abspielbitstrom I-Frames mit einer Frequenz von einem von X Einzelbildern hatte, weist die neue Videosequenz nur eines von jeweils X Bildern oder Einzelbildern des ursprünglichen nichtkomprimierten normalen Abspielbitstroms auf.

**[0075]** Dieser Ausgangsbildstrom wird dann einem Codierblock **506** bereitgestellt, wo der Strom mit den jeweiligen MPEG-Parametern erneut codiert wird, die für den Trickplay-Strom gewünscht werden. Diese MPEG-Parameter beinhalten unter anderem die Bitrate und die Bildgröße. Der MPEG-2-Codierer **506** erzeugt einen Trickplay-Strom, der ein gültiger MPEG-codierter Strom ist, der jedoch nur eines von jeweils X Einzelbildern beinhaltet. Wenn dieser neue MPEG-codierte Trickplay-Strom nachher zu einem Benutzer übertragen wird und von einem MPEG-Decoder decodiert wird, weist der resultierende Bitstrom einen Strom auf, der nur eines von jeweils X Einzelbildern des ursprünglichen nicht-komprimierten normalen Abspielbitstroms aufweist.

Erzeugung eines Rückwärts-Trickplay-Stroms – Alternative Ausführungsform

**[0076]** In **Fig. 6** ist eine alternative und bevorzugte Ausführungsform für das Erzeugen eines Rückwärts-Trickplay-Stroms gemäß der vorliegenden Er-

findung gezeigt. **Fig. 6** ist in drei Abschnitten gezeigt, die aus Bequemlichkeit als die **Fig. 6A**, **6B** und **6C** bezeichnet werden. In der Ausführungsform von **Fig. 6** wird der Rückwärts-Trickplay-Strom durch Abtasten des Videostroms vom Ende bis zum Beginn erzeugt. Diese Ausführungsform verwendet ebenso einen Speicherstapel, der zeitweilig Gruppierungen oder Tupel von MPEG-Daten speichert, so daß sie in der richtigen Ordnung in den Rückwärts-Trickplay-Strom aufgenommen werden.

**[0077]** Wie in **Fig. 6** gezeigt ist, wird in Schritt **602** eine Initialisierungsphase durchgeführt, wo der Stapel gelöscht wird. In Schritt **604** wird dann eine Suche nach dem ersten Bildstartcode in der Videosequenz durchgeführt. In Schritt **604** wird die Suche vom Ende zu dem Beginn des Videostroms durchgeführt. Wenn der erste Bildstartcode während der Suche vom Ende zum Beginn gefunden wird, wird in Schritt **606** eine Markierung auf den Stapel geschoben. Wie weiter unten erläutert wird, wird diese Markierung verwendet, um unterschiedliche Blöcke oder Abschnitte des Videostroms zu identifizieren. Nach Schritt **606** startet die Suchphase.

**[0078]** In Schritt **612** sucht das Verfahren nach einem Startcode in dem Videostrom. In Schritt **614** bestimmt das Verfahren, ob eine Datei-Ende-Bedingung während der Suche nach dem Startcode in Schritt **612** gefunden wird. Falls dies der Fall ist, wird in Schritt **616** der Ausgabestrom geschlossen, das Programm wird verlassen und die Operation ist vollständig. Wenn eine Datei-Ende-Bedingung in Schritt **614** nicht gefunden wird, findet in Schritt **620** das Verfahren voraussichtlich einen Startcode in dem Videostrom.

**[0079]** Es sei bemerkt, daß der gefundene Startcode den Start eines Benutzerdatenblockes, den Start eines Erweiterungsblockes, den Start eines B- oder P-Einzelbildkopfzeilenblockes oder den Start einer I-Frame-Kopfzeile aufweisen kann. Wie in **Fig. 6** gezeigt ist, werden unterschiedliche Operationen durchgeführt abhängig davon, welcher Typ von Startcode in Schritt **620** gefunden wird. Es sei bemerkt, daß die ovalen Blöcke in **Fig. 6** Kopfzeilen aufweisen und keine funktionalen Schritte sind.

**[0080]** Wenn der Startcode, der in Schritt **620** gefunden wird, der Start eines Benutzerdatenblockes ist, dann stellt in Schritt **622** das Verfahren die Inhalte des Benutzerdatenblockes, so wie es notwendig ist, ein. Die Inhalte des Benutzerdatenblockes werden eingestellt, um möglichen neuen Parametern zu entsprechen, die bei der Herstellung der Trickplay-Ströme verwendet werden, wie oben erwähnt wurde. Nachdem die Inhalte in Schritt **622** eingestellt wurden, verschiebt in Schritt **622** das Verfahren die Koordinaten des Benutzerdatenblockes auf den Stapel. Die Operation kehrt dann zu Schritt **612** zurück, wo

eine Suche nach einem neuen Startcode durchgeführt wird, wie oben beschrieben wurde.

**[0081]** Wenn der in Schritt **620** gefundene Startcode der Start eines Erweiterungsblockes ist, dann verschiebt in Schritt **632** das Verfahren die Koordinaten des Erweiterungsblockes auf den Stapel. Das Verfahren kehrt dann zu Schritt **612** zurück, um nach dem nächsten Startcode zu suchen.

**[0082]** Wenn der Startcode, der in Schritt **620** gefunden wird, der Start eines P- oder B-Frame-Kopfzeilenblockes ist, dann holt das Verfahren in Schritt **642** alle Koordinaten aus dem Stapel hervor, bis die Markierung erfaßt wird. Mit anderen Worten, falls der Start des P- oder B-Frame-Kopfzeilenblockes erfaßt wird, werden in Schritt **642** alle Koordinaten auf dem Stapel oberhalb der Markierung entfernt oder aus dem Stapel hervorgeholt. Danach wird die Markierung in Schritt **644** auf den Stapel geschoben oder auf den Stapel zurückgegeben. Die Operation kehrt dann zu Schritt **612** zurück, wo eine Suche nach einem neuen Startcode durchgeführt wird, wie oben beschrieben wurde.

**[0083]** Falls der Startcode, der in Schritt **620** gefunden wurde, der Start einer I-Frame-Kopfzeile ist, stellt in Schritt **652** das Verfahren die I-Frame-Kopfzeileninformation ein, so wie dies notwendig ist. Die I-Frame-Kopfzeileninformation wird eingestellt, damit sie an mögliche neue Parameter paßt, die bei der Erzeugung der Trickplay-Ströme verwendet wurden, wie oben erwähnt wurde. Nach dem Einstellen der I-Frame-Kopfzeile verschiebt das Verfahren in Schritt **654** die Koordinaten des I-Frame-Kopfzeilenblockes auf den Stapel. In Schritt **656** holt das Verfahren die Koordinaten, die auf den Stapel verschoben wurden, hervor oder entfernt sie, bis die Markierung erfaßt wird. In Schritt **648** schreibt das Verfahren die Daten in den entsprechenden Videostrom zu dem Rückwärts-Trickplay-Ausgabestrom, wenn jeder Koordinatensatz hervorgeholt wird. Hier wird bemerkt, daß Schritte **656** und **658** sequentiell arbeiten und im wesentlichen parallel arbeiten, wo, wenn Koordinaten von dem Stapel in Schritt **656** hervorgeholt werden, die Daten von dem entsprechenden Videostrom in den Rückwärts-Trickplay-Ausgangsstrom geschrieben werden, wenn die Koordinaten für den Stapel hervorgeholt werden.

**[0084]** Nachdem alle Koordinaten von dem Stapel in den Schritten **656** hervorgeholt wurden und in den Rückwärts-Trickplay-Ausgangsstrom **658** geschrieben wurden, verschiebt dann in Schritt **660** das Verfahren die Markierung zurück auf den Stapel. In Schritt **662** sucht das Verfahren dann nach dem ersten Sequenzkopfzeilenstartcode. Wenn dieser erste Sequenzkopfzeilenstartcode gefunden wurde, kehrt das Verfahren zu Schritt **612** zurück, um erneut nach dem nächsten Startcode zu suchen, und die Funktion

wiederholt sich, wie oben beschrieben.

**[0085]** Daher wird ein System und ein Verfahren bereitgestellt für das Erzeugen von Trickplay-Videostreamen aus einem komprimierten normalen Abspielvideostream. Das System oder das Verfahren untersucht jede Kopfzeile oder jedes Einzelbild in einer MPEG-Sequenz und speichert die Sequenzkopfzeilen, die I-Frames und die verknüpften Wichtungsmatrizen. Das System oder das Verfahren setzt einen neuen schnellen Vorlauf- oder schnellen Rücklaufbitstrom, der diese gespeicherten Komponenten enthält, intelligent zusammen. Der neue zusammengesetzte Bitstrom wird dann decodiert, um eine Mehrzahl von nicht-komprimierten Einzelbildern zu erzeugen. Diese nichtkomprimierten Einzelbilder werden dann erneut codiert gemäß dem MPEG-Standard, um einen neuen MPEG-Strom zu erzeugen, der ein schneller Vorlauf- oder schneller Rücklauf-Trickplay-Strom ist.

**[0086]** Obgleich das System und das Verfahren der vorliegenden Erfindung in Verbindung mit den beschriebenen Ausführungsformen beschrieben wurden, ist es nicht beabsichtigt, die Erfindung auf die speziellen, hier ausgeführten Formen zu beschränken, sondern es ist im Gegenteil beabsichtigt, alle solche Alternativen, Modifikationen und Äquivalente, sofern sie vernünftig innerhalb des Schutzbereiches der Erfindung aufgenommen werden können, abzudecken.

### Patentansprüche

1. Computerimplementiertes Verfahren für die Erzeugung von Trickplayströmen aus einem komprimierten normalen Abspielbitstrom, das aufweist:  
Empfangen eines komprimierten normalen Abspielbitstroms, wobei der komprimierte normale Abspielbitstrom eine Mehrzahl von intracodierten Einzelbildern und eine Mehrzahl von intercodierten Einzelbildern beinhaltet,  
Extrahieren der intracodierten Einzelbilder aus dem komprimierten normalen Abspielbitstrom, wobei das Extrahieren das Speichern der intracodierten Einzelbilder in einem Ablagespeicher (**102**) beinhaltet,  
Zusammensetzen der intracodierten Einzelbilder nach dem Extrahieren (**104**), um einen zusammengesetzten Bitstrom zu bilden,  
Decodieren des zusammengesetzten Bitstroms, um eine Mehrzahl von nicht komprimierten Einzelbildern (**106**) zu erzeugen, und  
Codieren der Mehrzahl von nicht komprimierten Einzelbildern nach der Decodierung, um einen komprimierten Trickplay-Bitstrom zu erzeugen, wobei der komprimierte Trickplay-Bitstrom nur eine Untergruppe von Einzelbildern des normalen Abspielbitstroms (**108**) beinhaltet.

2. Verfahren nach Anspruch 1, wobei der kompri-

mierte normale Abspielbitstrom die Mehrzahl von intracodierten Einzelbildern bei einer spezifizierten Frequenz beinhaltet, wobei das Extrahieren **(102)** das Extrahieren von Daten aufweist, die den intracodierten Einzelbildern bei der spezifizierten Frequenz entsprechen.

3. Verfahren nach Anspruch 1, wobei der komprimierte normale Abspielbitstrom eine Mehrzahl von Sequenzkopfzeilen beinhaltet, die Informationen für zumindest eine Mehrzahl der intracodierten Einzelbilder aufweisen, wobei das Extrahieren **(102)** das Extrahieren der Sequenzkopfzeilen aus dem komprimierten normalen Abspielbitstrom aufweist, wobei das Extrahieren das Speichern der Sequenzkopfzeilen in einem Ablagespeicher beinhaltet, und wobei das Zusammensetzen **(104)** das Zusammensetzen der Sequenzkopfzeilen und der intracodierten Einzelbilder aufweist, um den zusammengesetzten Bitstrom zu bilden.

4. Verfahren nach Anspruch 1, wobei das Verfahren einen Schneller-Vorlauf-Trickplay-Bitstrom erzeugt, wobei das Zusammensetzen **(104)** das Zusammensetzen der intracodierten Einzelbilder in einer Vorwärtszeitrichtung aufweist.

5. Verfahren nach Anspruch 1, wobei das Verfahren einen Schneller-Rücklauf-Trickplay-Bitstrom erzeugt, wobei das Zusammensetzen **(104)** das Zusammensetzen der intracodierten Einzelbilder in einer rückwärts gerichteten Zeitordnung aufweist.

6. Verfahren nach Anspruch 1, wobei der komprimierte normale Abspielbitstrom eine Mehrzahl von Matrizen beinhaltet, die den intracodierten Einzelbildern entsprechen, wobei das Verfahren weiterhin aufweist: Lokalisieren der Matrizen in dem komprimierten normalen Abspielbitstrom, wobei das Zusammensetzen **(104)** der intracodierten Einzelbilder, um den zusammengesetzten Bitstrom zu bilden, das Einschließen der Matrizen in den zusammengesetzten Bitstrom beinhaltet.

7. Verfahren nach Anspruch 1, wobei jede der Mehrzahl von Matrizen einem der intracodierten Einzelbilder entspricht, wobei das Zusammensetzen **(104)** das Zusammensetzen einer entsprechenden Matrix der Matrizen mit einem entsprechenden Einzelbild der intracodierten Einzelbilder aufweist.

8. Verfahren nach Anspruch 1, wobei der komprimierte normale Abspielbitstrom ein MPEG-komprimierter Bitstrom ist, wobei das Decodieren **(106)** des zusammengesetzten Bitstroms die MPEG-Decodierung des zusammengesetzten Bitstroms aufweist, um die Mehrzahl

von nicht komprimierten Einzelbildern zu erzeugen, und wobei das Codieren **(108)** der Mehrzahl von nicht komprimierten Einzelbildern die MPEG-Codierung der Mehrzahl von nicht komprimierten Einzelbildern aufweist, um einen MPEG-komprimierten Abspielbitstrom zu erzeugen.

9. System für das Erzeugen von Trickplay-Strömen aus einem komprimierten normalen Abspielbitstrom, das aufweist:

ein Speichermedium **(90)**, das einen komprimierten normalen Abspielbitstrom speichert, wobei der komprimierte normale Abspielbitstrom eine Mehrzahl von intracodierten Einzelbildern und eine Mehrzahl von intercodierten Einzelbildern aufweist, einen Filter **(102)** für das Extrahieren der intracodierten Einzelbilder aus dem komprimierten normalen Abspielbitstrom, einen Ablagespeicher **(84)** für das Speichern der extrahierten intracodierten Einzelbilder, einen Verifizierer/Fixierer **(104)** für das Zusammensetzen der gespeicherten intracodierten Einzelbilder, um einen zusammengesetzten Bitstrom zu bilden, einen Decoder **(106)**, der derart betreibbar ist, daß er den zusammengesetzten Bitstrom decodiert, um eine Mehrzahl von nicht komprimierten Einzelbildern zu erzeugen, und einen Codierer **(108)**, der derart betreibbar ist, daß er die Mehrzahl von nicht komprimierten Einzelbildern codiert, um einen komprimierten Trickplay-Bitstrom zu erzeugen, wobei der komprimierte Trickplay-Bitstrom nur eine Untergruppe von Einzelbildern des normalen Abspielbitstroms beinhaltet.

10. System nach Anspruch 9, wobei der komprimierte normale Abspielbitstrom die Mehrzahl von intracodierten Einzelbildern bei einer spezifizierten Frequenz beinhaltet, wobei der Filter **(102)** derart betreibbar ist, daß er Datenbits entsprechend den intracodierten Einzelbildern bei der spezifizierten Frequenz extrahiert.

11. System nach Anspruch 9, wobei der komprimierte normale Abspielbitstrom eine Mehrzahl von Sequenzkopfzeilen beinhaltet, die Informationen für zumindest eine Mehrzahl der intracodierten Einzelbilder aufweist, wobei der Filter **(102)** derart betreibbar ist, daß er die Sequenzkopfzeilen aus dem komprimierten normalen Abspielbitstrom extrahiert und die Sequenzkopfzeilen in dem Ablagespeicher ablegt, und wobei der Verifizierer/Fixierer **(104)** derart betreibbar ist, daß er die Sequenzkopfzeilen und die intracodierten Einzelbilder zusammensetzt, um den zusammengesetzten Bitstrom zu bilden.

12. System nach Anspruch 9, wobei das System derart betreibbar ist, daß es einen Schneller-Vorlauf-Trickplay-Bitstrom erzeugt, wobei der Verifizie-

rer/Fixierer (**104**) derart betreibbar ist, daß er die intracodierten Einzelbilder in einer vorwärts gerichteten Zeitordnung zusammensetzt.

13. System nach Anspruch 9, wobei das System derart betreibbar ist, daß es einen Schneller-Rücklauf-Trickplay-Bitstrom erzeugt, wobei der Verifizierer/Fixierer (**104**) derart betreibbar ist, daß er die intracodierten Einzelbilder in einer umgekehrten Zeitordnung zusammensetzt.

14. System nach Anspruch 9, wobei der komprimierte normale Abspielbitstrom eine Mehrzahl von Matrizen beinhaltet, die den intracodierten Einzelbildern entsprechen, wobei der Filter (**102**) derart betreibbar ist, daß er die Matrizen in dem komprimierten normalen Abspielbitstrom lokalisiert und die Matrizen in dem Ablagespeicher ablegt, wobei der Verifizierer/Fixierer (**104**) derart betreibbar ist, daß er die intracodierten Einzelbilder und die Matrizen zusammensetzt, um den zusammengesetzten Bitstrom zu bilden.

15. System nach Anspruch 9, wobei jede der Mehrzahl von Matrizen einem der intracodierten Einzelbilder entspricht, wobei der Verifizierer/Fixierer (**104**) derart betreibbar ist, daß er entsprechende Matrizen mit entsprechenden intracodierten Einzelbildern zusammensetzt.

16. System nach einem der Ansprüche 9 bis 15, wobei der komprimierte normale Abspielbitstrom ein MPEG-komprimierter Bitstrom ist, wobei der Decoder (**106**) ein MPEG-Decoder ist und wobei der Codierer (**108**) ein MPEG-Codierer ist.

Es folgen 9 Blatt Zeichnungen

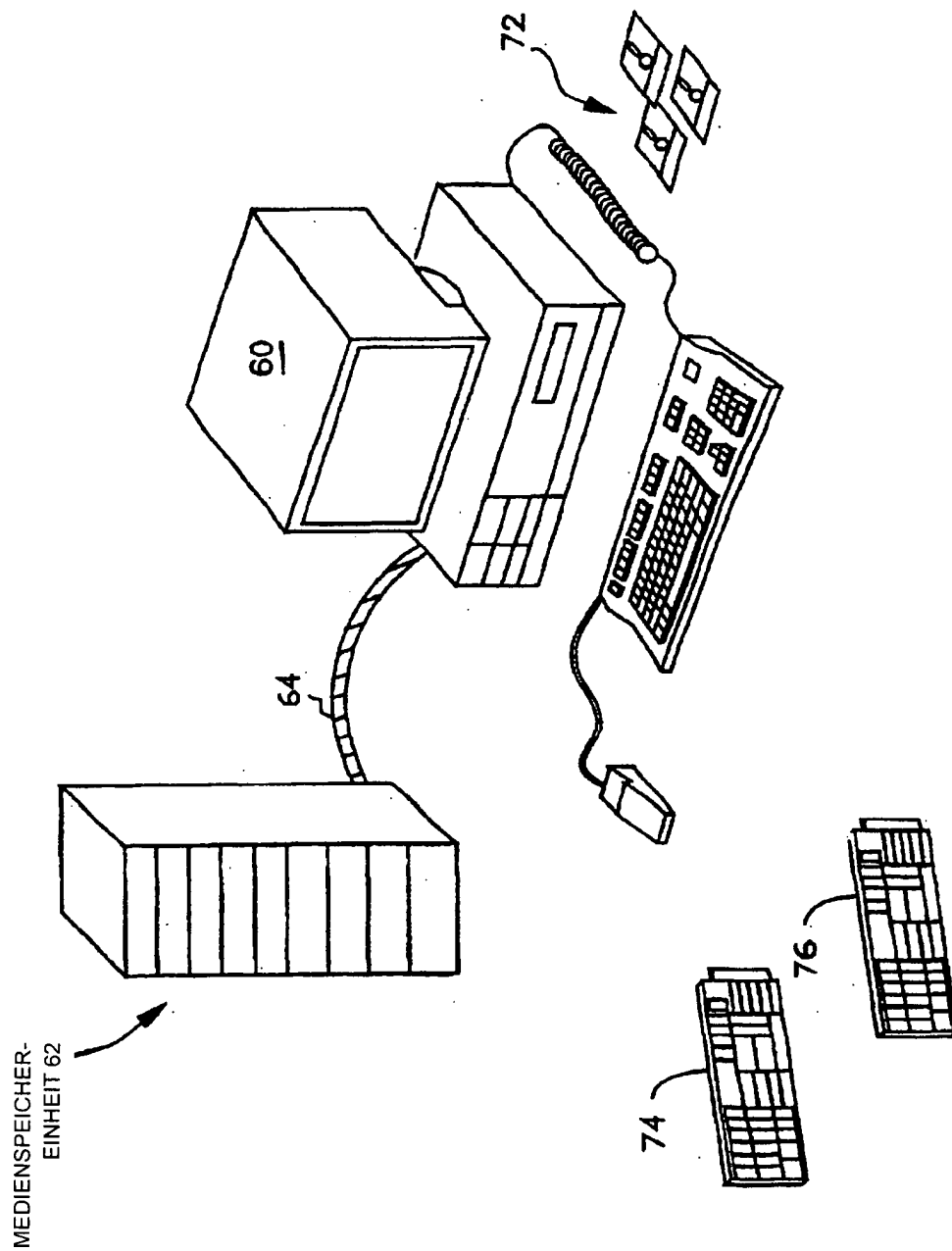


FIG. 1

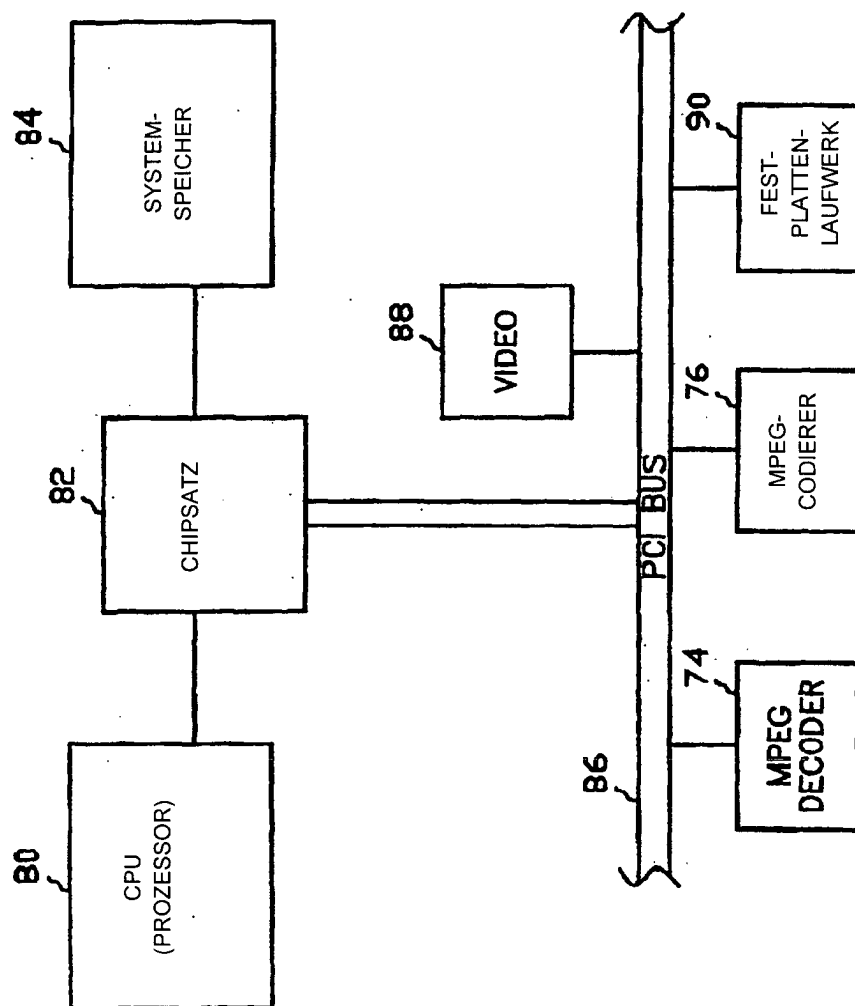


FIG. 1A

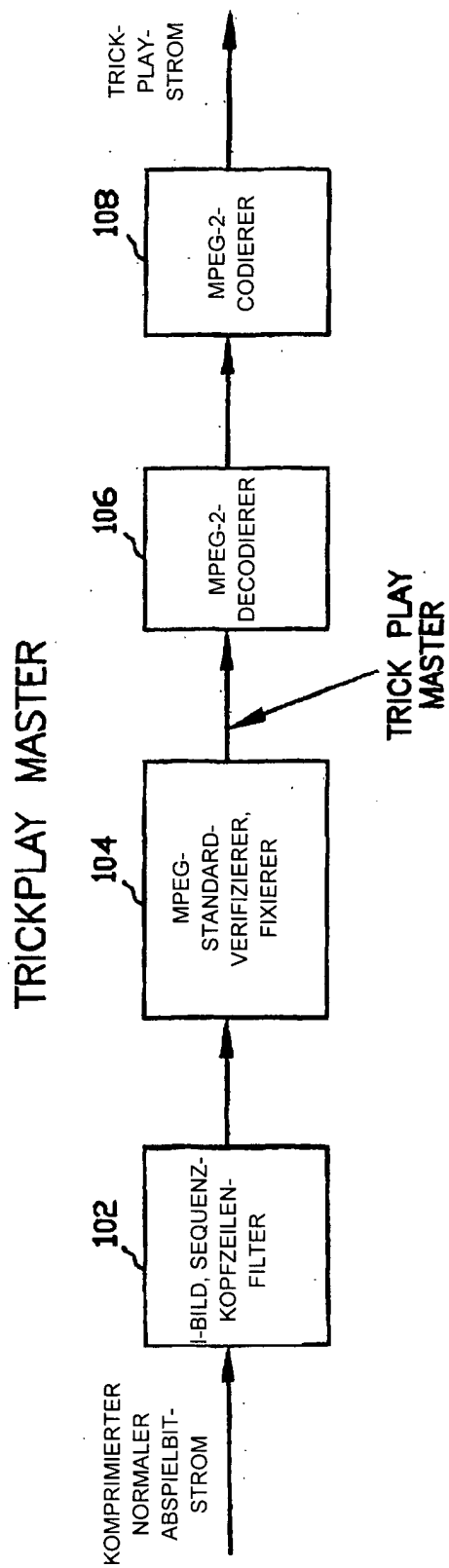


FIG. 2

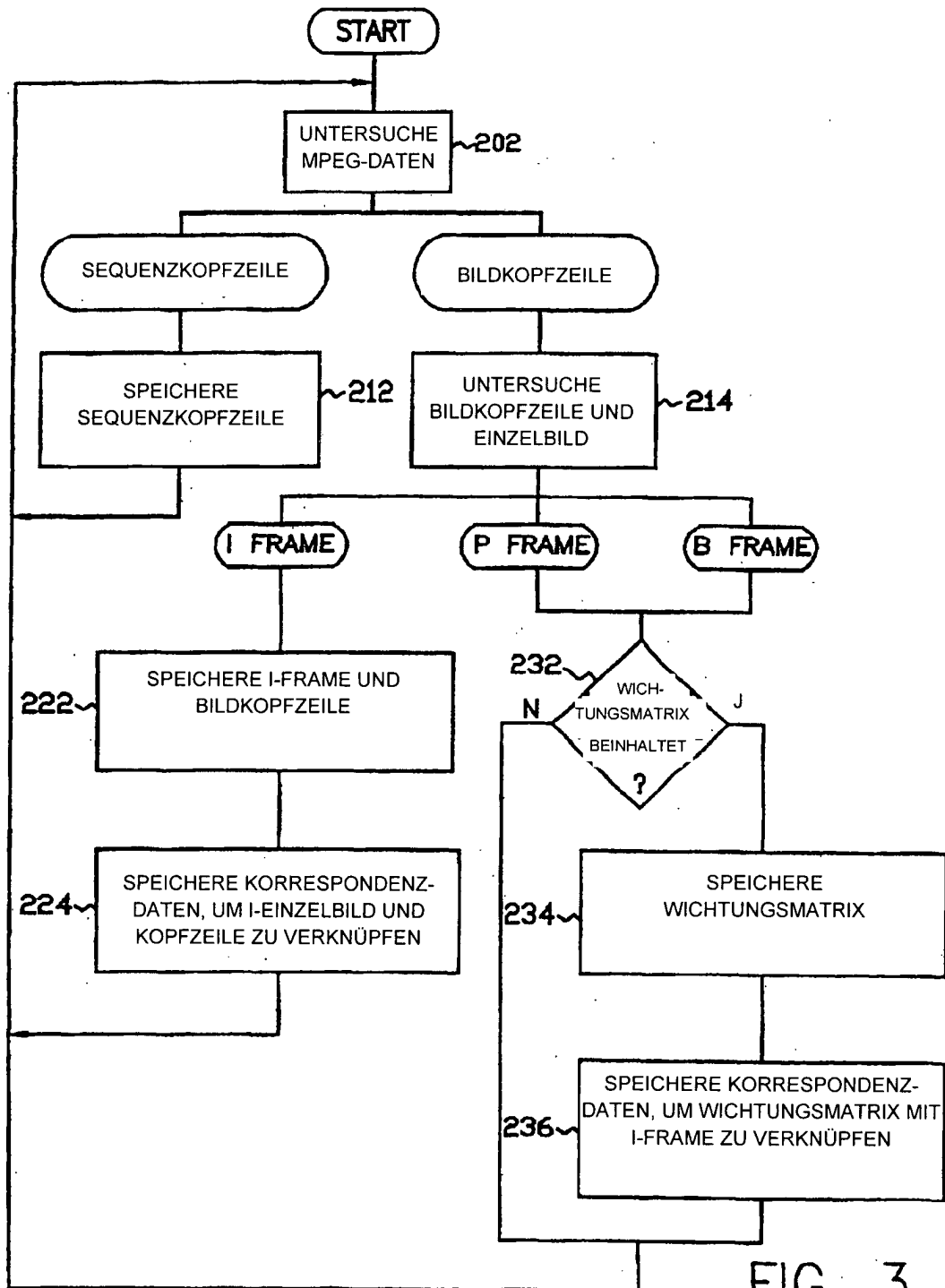


FIG. 3  
FILTER OPERATION

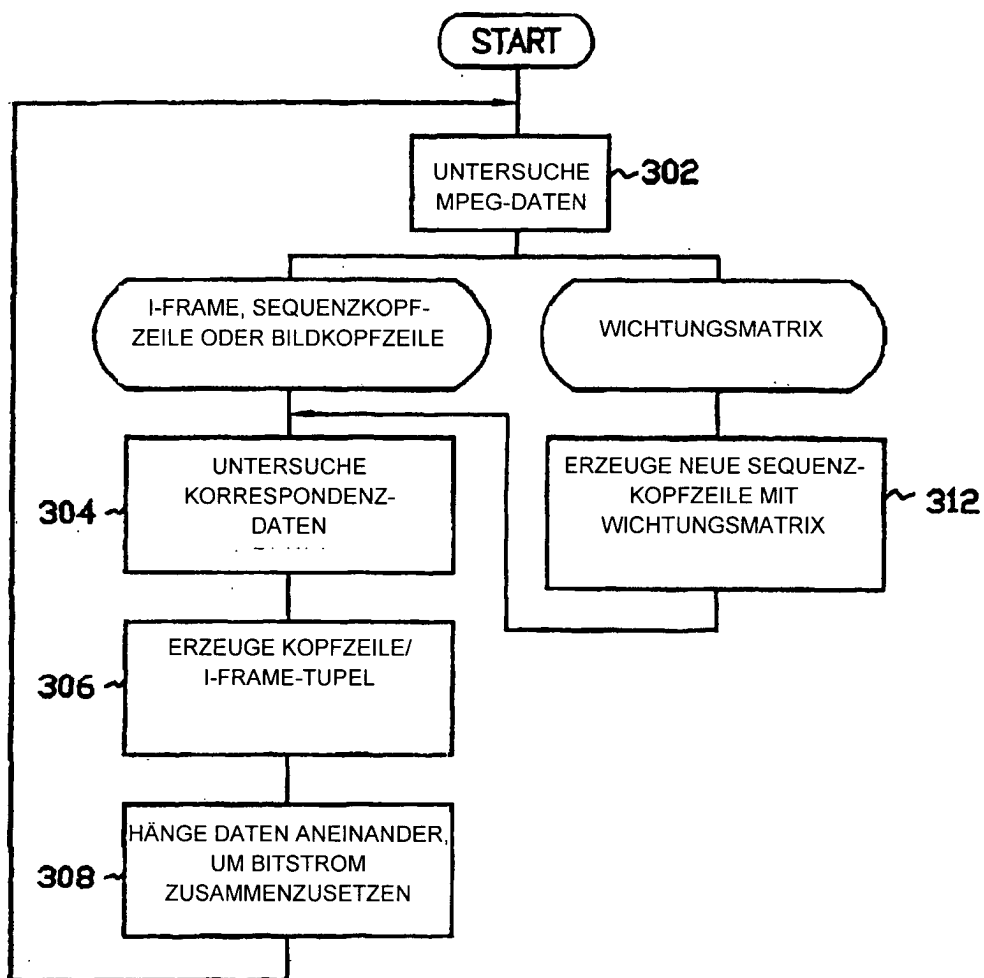


FIG. 4  
VERIFIZIER-/FIXIEROPERATION

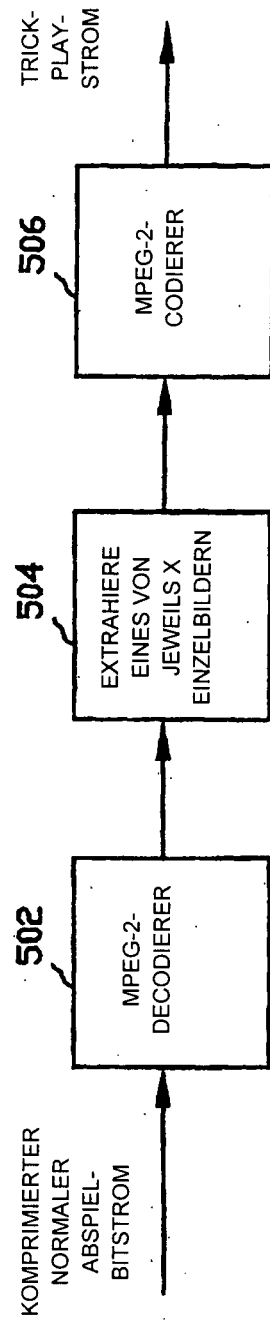


FIG. 5  
ALTERNATIVE AUSFÜHRUNGSFORM

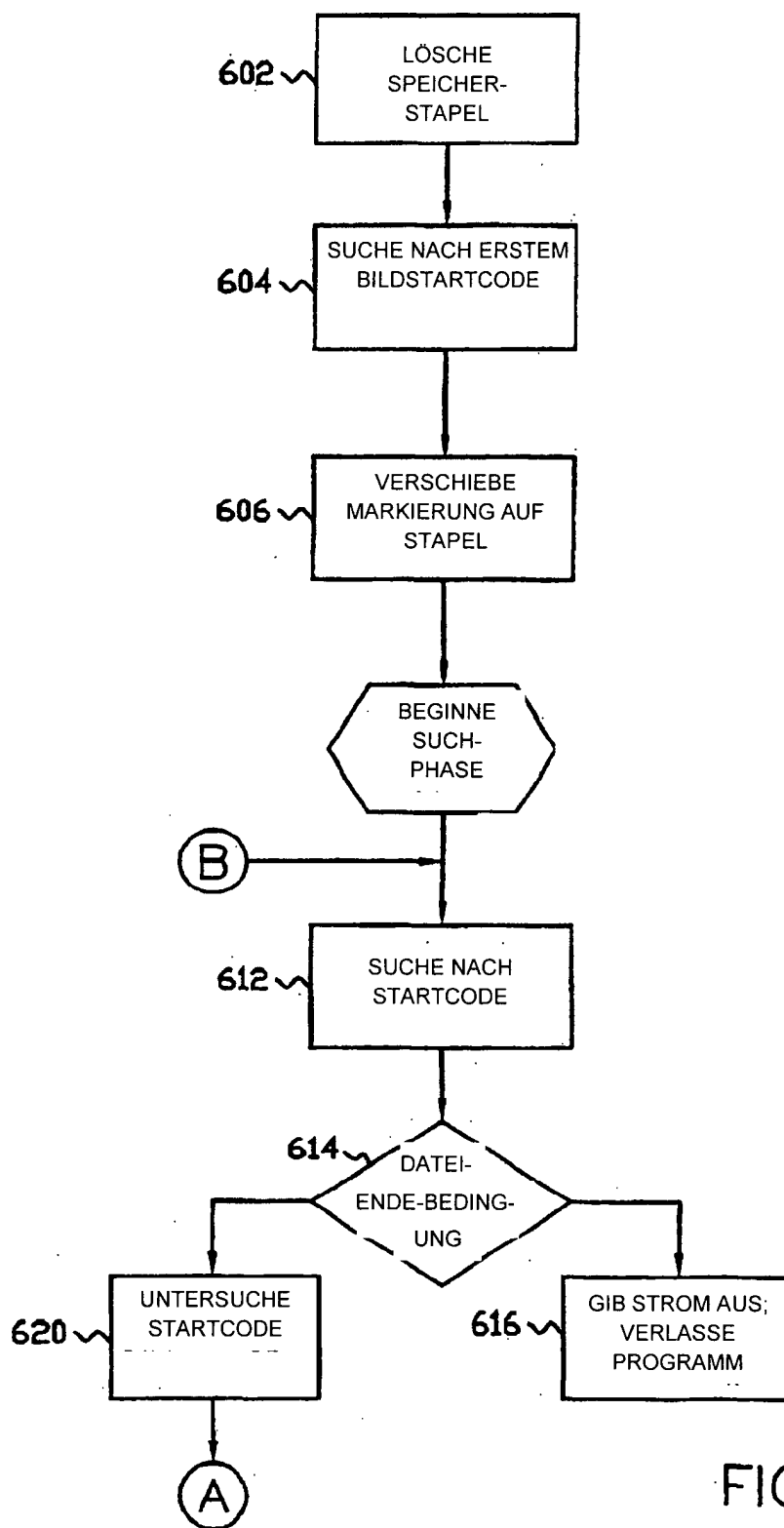


FIG. 6A

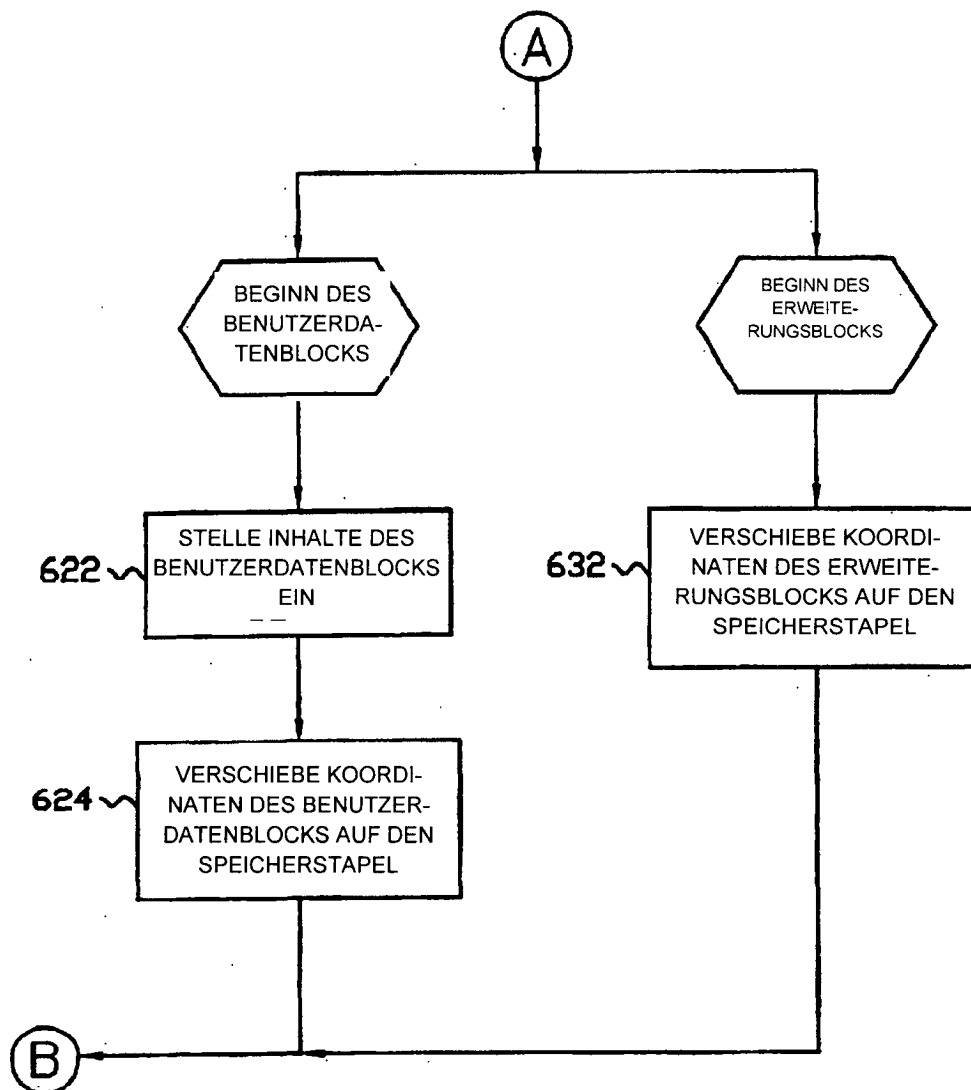


FIG. 6B

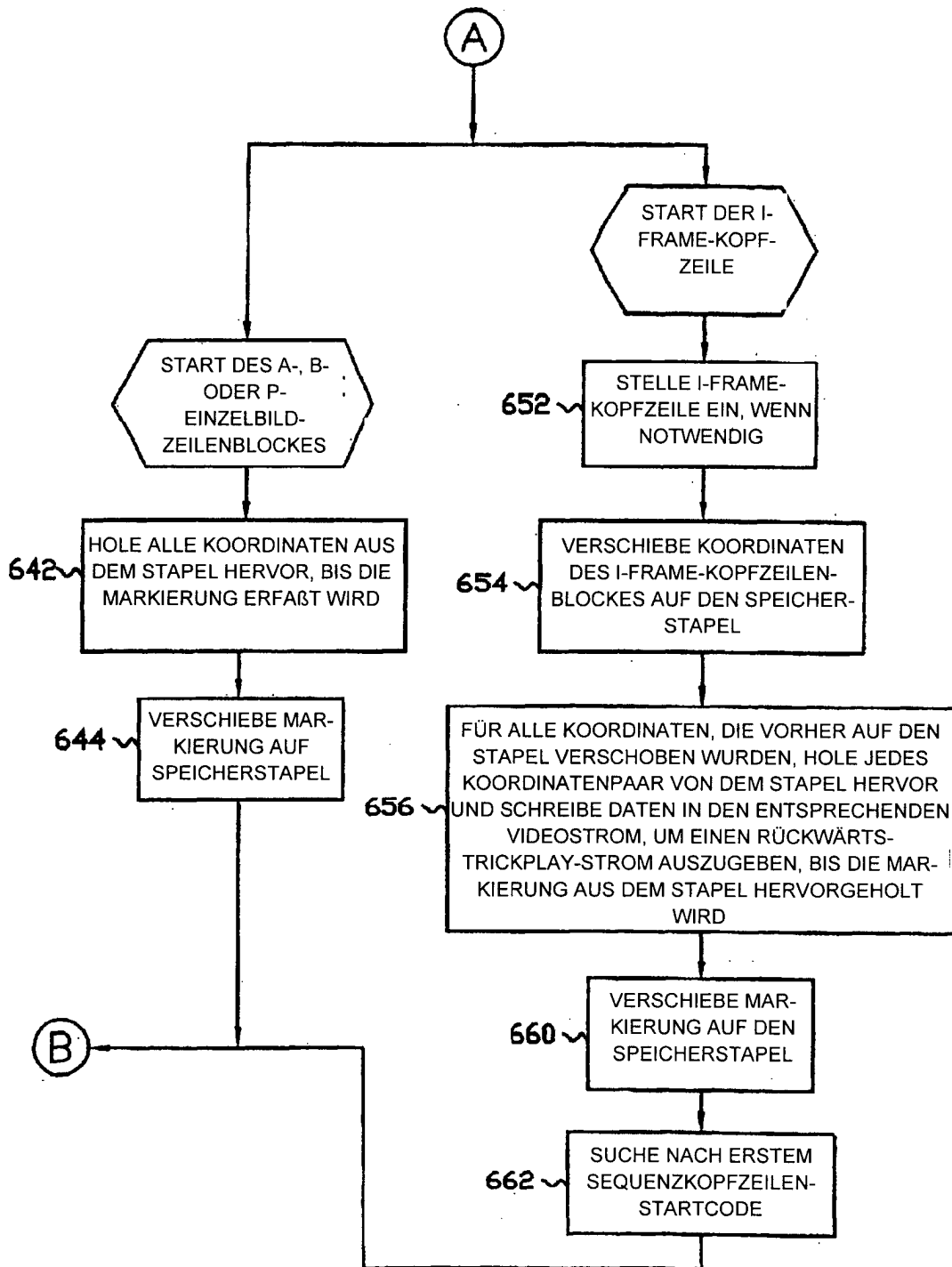


FIG. 6C