



(19)
Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) DE 698 37 726 T2 2008.01.10

(12)

Übersetzung der europäischen Patentschrift

(97) EP 1 162 828 B1

(21) Deutsches Aktenzeichen: 698 37 726.5

(96) Europäisches Aktenzeichen: 01 117 966.0

(96) Europäischer Anmeldetag: 19.10.1998

(97) Erstveröffentlichung durch das EPA: 12.12.2001

(97) Veröffentlichungstag

der Patenterteilung beim EPA: 02.05.2007

(47) Veröffentlichungstag im Patentblatt: 10.01.2008

(51) Int Cl.⁸: G11B 27/00 (2006.01)

H04N 7/173 (2006.01)

H04N 5/00 (2006.01)

H04N 7/24 (2006.01)

(30) Unionspriorität:

956262 22.10.1997 US

(84) Benannte Vertragsstaaten:

DE, FR, GB, NL

(73) Patentinhaber:

nCUBE Corp., Beaverton, Oreg., US; Thirdspace
Living,Ltd., Maidenhead, Berkshire, GB

(72) Erfinder:

Weaver, Daniel, Redwood City, CA 94061, US;
Pawson, David J., San Mateo, CA 94402, US

(74) Vertreter:

Viering, Jentschura & Partner, 81675 München

(54) Bezeichnung: **Verfahren und Vorrichtung zum Realisieren einer nahtlosen Wiedergabe von kontinuierlichen
Medien-Zuführungen**

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 eingelebt, 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**Verwandte Anmeldungen**

[0001] Die Anmeldung bezieht sich auf: das US-Patent Nr. 08/956,261, betitelt mit „Method and Apparatus for Concurrently Encoding and Tagging Media“, eingereicht von Daniel Weaver, Mark A. Porter und David J. Pawson am 22. Oktober 1997 und

das US-Patent Nr. 08/956,263, betitelt mit „Method and Apparatus for Non-Sequential Access To An In-Progress Video Feed“, eingereicht von Daniel Weaver, Mark A. Porter und David J. Pawson am 22. Oktober 1997.

Gebiet der Erfindung

[0002] Die vorliegende Erfindung betrifft ein Verfahren und eine Vorrichtung zum Verarbeiten audio-visueller Information und insbesondere ein Verfahren und eine Vorrichtung zum Bereitstellen eines nicht-sequenziellen Zugriffs auf audio-visuelle Information, die in einem Live-Content-Strom dargestellt ist.

Hintergrund der Erfindung

[0003] In den letzten Jahren hat die Medienindustrie ihre Horizonte über traditionelle analoge Technologien hinaus erweitert. Töne, Fotografien und selbst Spielfilme werden nun in digitale Formate aufgezeichnet oder umgewandelt. Um die Kompatibilität zwischen Produkten zu fördern, wurden in vielen der Medienkategorien Standardformate entwickelt.

[0004] Wie erwartet wurde, wünschen die Zuschauer eines digitalen Videos von den Herstellern digitaler Videos die gleiche Funktionalität, wie sie sie nun genießen, während sie sich analoge Videobänder in Video-Kassettenrekordern anschauen. Beispielsweise möchten Zuschauer befähigt sein, im Video vorwärts zu springen, zurück zu springen, schnell vorwärts, schnell zurückzuspulen, langsam vorwärts, langsam rückzuspulen und einen Rahmen einzufrieren.

[0005] Es wurden verschiedene Ansätze entwickelt, um eine nicht-sequenzielle Wiedergabe digitaler Videodaten bereitzustellen. In Bezug auf digitale Videodaten betrifft die nicht-sequenzielle Wiedergabe eine Wiedergabe-Operation, bei der nicht alle der kodierten Rahmen genau in der Reihenfolge in der Sequenz abgespielt werden, in der sie kodiert worden sind. Beispielsweise sind die Operationen des Vorwärtsspringens und des schnellen Vorspulens in der Hinsicht nicht-sequenziell, dass einige Rahmen übersprungen werden. Die Operationen des Zurückspulens mit irgendeiner Geschwindigkeit in der Hinsicht nicht-sequenziell, dass während einer Operation des Zurückspulens Rahmen nicht in der Reihenfolge abgespielt werden, in der sie kodiert sind.

[0006] Ein Ansatz zum Bereitstellen einer nicht-sequenziellen Wiedergabe digitaler Videodaten, an dieser Stelle bezeichnet als Tag-basierter Ansatz, ist im US-Patent Nr. 5,659,539, betitelt mit „Method and Apparatus for Frame Accurate Access of Digital Audio-visual Information“, das Porter et al am 19. August 1997 erteilt worden ist, beschrieben. Gemäß dem Tagbasierten Ansatz wird eine gespeicherte digitale Videodatei geparsst, um „Tag-Information“ über einzelne Rahmen innerhalb der Datei zu erzeugen.

[0007] Insbesondere enthält die Tag-Datei Informationen über den Zustand einer oder mehrerer Zustandsmaschinen, die verwendet werden, um die digitale Darstellung zu dekodieren. Die Zustandsinformation variiert abhängig von der spezifischen Technik, die verwendet wird, um die audio-visuelle Arbeit zu kodieren. Für MPEG 2-Dateien beispielsweise weist die Tag-Datei Informationen über den Zustand der Programm-Elementarstrom-Zustandsmaschine, der Video-Zustandsmaschine und der Transportschicht-Zustandsmaschine auf.

[0008] Während der Durchführung der audio-visuellen Arbeit werden Daten von der digitalen Darstellung von einer Videopumpe an einen Dekoder gesendet. Die Information in der Tag-Datei wird verwendet, um die Operationen des Suchens, des schnellen Vorspulens, des schnellen Zurückspulens, des langsam Vorspulens und des langsamen Zurückspulens während der Durchführung der audio-visuellen Arbeit durchzuführen. Such-Operationen werden durchgeführt, indem die Videopumpe dazu gebracht wird, das Übertragen von Daten von der aktuellen Position in der digitalen Darstellung anzuhalten und das Übertragen von Daten von einer neuen Position in der digitalen Darstellung an zu starten. Die Information in der Tag-Datei wird untersucht, um die neue Position zu ermitteln, von der begonnen werden soll, die Daten zu übertragen. Um sicherzustellen, dass der Datenstrom, der mittels der Videopumpe übertragen wird, gemäß dem anwendbaren Videoformat verwaltet wird, werden Präfix-Daten, die eine geeignete Header-Information aufweisen, mittels der Videopumpe übertragen, bevor die Daten von der neuen Position an übertragen werden.

[0009] Die Operationen des schnellen Vorspulens, des schnellen Zurückspulens, des langsamen Vorspulens und des langsamen Zurückspulens werden durch Auswählen von Videorahmen basierend auf der Information, die in der Tag-Datei enthalten ist, und der gewünschten Darstellungsrate und mittels Erzeugens eines Datenstroms durchgeführt, der Daten enthält, die die ausgewählten Videorahmen darstellen. Bei dem Auswahlprozess wird eine Vielzahl von Faktoren beachtet, inklusive der Datentransferrate des Kanals, auf dem die Daten zu senden sind, dem Rahmen-Typ der Rahmen, einer minimalen Auffüllrate und der Möglichkeit eines Puffer-Überlaufs im Dekoder. Präfix- und Suffix-Daten werden in den übertragenen Datenstrom vor und nach den Daten für jeden Rahmen eingefügt, um die Einhaltung des Datenstrom-Formats aufrechtzuerhalten, das vom Dekoder erwartet wird.

[0010] Der Tag-basierte Ansatz funktioniert gut, wenn es genügend Zeit zwischen dem Erzeugen des ursprünglichen digitalen Videostroms und dem Anschauen des digitalen Videostroms gibt, sodass es möglich ist, dass der ursprüngliche digitale Videostrom geparst wird, um die Tag-Information zu erzeugen. Wenn jedoch der digitale Videostrom angeschaut wird, während er erzeugt wird, wird das Parsen des digitalen Videostroms unpraktikabel. Die Größe der Rechenkraft, die erforderlich ist, um den digitalen Videostrom zu parsen, sobald er ankommt, würde unerschwinglich teuer sein. Andererseits wird es als nicht akzeptabel betrachtet, die Latenzzeit zwischen dem Auftreten vieler Typen von Videozuführungen (beispielsweise Sportereignissen) und der Zeit zu erhöhen, zu der solche Zuführungen einem Publikum zum Anschauen verfügbar sind.

[0011] Wird ein Videostrom zum Anschauen verfügbar gemacht, bevor die Erzeugung des Streams abgeschlossen ist, ist der Videostrom eine so genannte „Live-Zufuhr“. Auf professioneller Ebene können nichtlineare digitale Editoren verwendet werden, um Filmmaterial einer Live-Zufuhr für einen einzelnen Nutzer schnell zu überprüfen. Jedoch sind diese Systeme nicht ausgerichtet und können nicht einfach angepasst werden daran, vielen Nutzern zu dienen. Wenn beispielsweise hundert Nutzer die gleiche Live-Zufuhr sehen würden, aber zu unterschiedlichen Zeiten die Zufuhr zurückspulen, anhalten und schnell vorspulen möchten, würde jeder einen separaten nichtlinearen digitalen Editor benötigen.

[0012] Ein anderes Problem, das mit dem Bereitstellen eines nichtlinearen Zugriffs auf digitale Live-Videoströme verbunden ist, ist, dass Nutzer versuchen könnten, in Abschnitte des Videostroms schnell vorzuspulen, die noch nicht existieren. Beispielsweise kann ein Zuschauer versuchen, eine Live-Zufuhr schnell vorzuspulen, um den Endstand eines Spiels zu sehen, das in Wirklichkeit noch nicht beendet ist. Es ist wünschenswert, Techniken zum Behandeln dieser Typen von Situationen in einer Weise bereitzustellen, die sicherstellt, dass weder der Dekoder den Videostrom einfriert, noch dass der Videostrom zerstört wird.

[0013] Ein System und Verfahren, das ein graphisches Symbol, wie zum Beispiel einen Schiebereglernopf, auf einer Fernseh- oder Anzeigeeinheit eines Teilnehmers zum Indizieren verschiedener Positionen in einem Videostrom in einem interaktiven Video-Zuführsystem anzeigt, ist in der WO 98/00973 A1, die am 8 Januar 1998 veröffentlicht wurde, offenbart.

[0014] Basierend auf dem Vorangegangenen ist es klar wünschenswert, ein Verfahren und eine Vorrichtung zum sequenziellen Anzeigen nicht-sequenzieller Rahmen eines digitalen Live-Videos bereitzustellen. Es ist ferner wünschenswert, einen nicht-sequenziellen Zugriff auf ein digitales Live-Video in einer Weise bereitzustellen, sodass es nicht erforderlich ist, dass jeder Zuschauer eine unerschwinglich teure Hardware betreibt. Es ist ferner wünschenswert, Vorkehrungen zu treffen gegen Versuche, auf Bereiche eines digitalen Live-Videos zuzugreifen, die noch nicht existieren.

Zusammenfassung der Erfindung

[0015] Ein Verfahren zum Bereitstellen eines nichtsequentiellen Zugriffs auf Videos von einer kontinuierlichen Zufuhr ist in dem unabhängigen Anspruch offenbart. Bevorzugte Ausführungsbeispiele der vorliegenden Erfindung werden in den abhängigen Patentansprüchen beschrieben.

[0016] Gemäß der Erfindung wird Tag-Information, die Information über Rahmen anzeigt, die in dem digitalen Datenstrom enthalten sind, erzeugt. Die Tag-Information weist Zeitstempel auf, die die Zeitsteuerung von Rahmen relativ zu einem Anfang des digitalen Datenstroms anzeigen. Ein Anfangszeitwert zeigt eine Absolutzeit an, die dem Anfang des Datenstroms entspricht.

[0017] Wenn eine Anforderung von einem Lesegerät für eine Wiedergabe empfangen wird, die bei einer festgelegten Absolutzeit startet, wird der Anfangszeitwert von der festgelegten Absolutzeit subtrahiert, um eine Relativzeit zu bestimmen. Die Tag-Information wird verwendet, um eine Stelle in dem digitalen Datenstrom zu

identifizieren, die zu der Relativzeit korrespondiert. Der digitale Datenstrom wird dann an das Lesegerät übertragen, wobei bei der Stelle in dem digitalen Datenstrom gestartet wird, die zu der Relativzeit korrespondiert.

Kurzbeschreibung der Zeichnungen

[0018] Die vorliegende Erfindung ist im Wege eines Beispiels und nicht im Wege einer Beschränkung in den Figuren der beigefügten Zeichnungen dargestellt, bei denen sich gleiche Bezugszeichen auf gleiche Elemente beziehen und bei denen:

[0019] [Fig. 1](#) ein Blockdiagramm ist, das ein Video-Zuführsystem gemäß einem Ausführungsbeispiel der Erfindung zeigt;

[0020] [Fig. 2A](#) ein Blockdiagramm ist, das das Format einer MPEG-Datei zeigt;

[0021] [Fig. 2B](#) ein Blockdiagramm einer beispielhaften Tag-Datei ist;

[0022] [Fig. 2C](#) ein Blockdiagramm ist, das die Tag-Information darstellt, die für jeden Rahmen in einer MPEG 1-Datei erzeugt worden ist;

[0023] [Fig. 3A](#) ein Blockdiagramm ist, das ein Speichersystem darstellt, bei dem RAID-Fehlerkorrektur-Techniken angewendet werden;

[0024] [Fig. 3B](#) ein Blockdiagramm ist, das ein Speichersystem darstellt, bei dem eine RAID-Fehlerkorrektur und ein Platten-Striping kombiniert sind;

[0025] [Fig. 4](#) ein Blockdiagramm ist, das eine Folge von Inhalts-Dateien darstellt, die verwendet werden, um den Inhalt einer kontinuierlichen Zufuhr zu speichern; und

[0026] [Fig. 5](#) ein Blockdiagramm ist, das die Migration von Tag-Information von einer alten Tag-Datei in eine neue Tag-Datei als Antwort auf das Ablaufen der Tag-Daten innerhalb der alten Tag-Datei darstellt.

Ausführliche Beschreibung des bevorzugten Ausführungsbeispiels

[0027] Ein Verfahren und eine Vorrichtung zum Bereitstellen eines nicht-sequenziellen Zugriffs auf einen digitalen Live-Videostrom werden beschrieben. In der folgenden Beschreibung werden zu Erläuterungszwecken zahlreiche spezifische Details dargelegt, um ein gründliches Verständnis der Erfindung zu gewährleisten. Es wird jedoch einem Fachmann offenbar, dass die vorliegende Erfindung ohne diese spezifischen Details ausgeführt werden kann. Bei anderen Beispielen sind wohlbekannte Strukturen und Vorrichtungen in Blockdiagramm-Form gezeigt, um eine unnötige Verdunklung der vorliegenden Erfindung zu vermeiden.

Funktions-Übersicht

[0028] Der Schwierigkeit, die mit dem Anwenden des Tagbasierten Ansatzes auf Live-Digital-Videozuführungen verbunden ist, wird gemäß einem Aspekt des vorliegenden Systems durch das Beseitigen des Erfordernisses begegnet, einen eingehenden digitalen Videostrom in Echtzeit zu parsen. Anstelle des Erzeugens von Tag-Daten mittels Parsens des digitalen Videostroms behält die Einheit, die für das Kodieren der Live-Zufuhr zuständig ist, die Information darüber, wie die Daten kodiert worden sind, und überträgt diese Information zusammen mit den kodierten Daten zum Video-Server. Die Tag-Information kommt am Video-Server zusammen mit dem entsprechenden Inhalt an, so dass der Inhalt selbst nicht geparszt werden muss.

[0029] Der Video-Server ist derart eingerichtet, dass sichergestellt ist, dass der Client nicht hinter dem Ende des empfangenen Inhalts suchen oder abtasten kann. Infolge der Tatsache, dass es eine gewisse Menge an Verzerrung zwischen der Ankunftszeit des Inhalts und der entsprechenden Tags geben kann, ist der Server eingerichtet sicherzustellen, dass die Tags nicht verfrüh verwendet werden, d. h., dass sie den Server dazu bringen würden, sich hinter das Ende des verfügbaren Inhalts zu bewegen.

Beispielhaftes System

[0030] [Fig. 1](#) ist ein Blockdiagramm, das ein beispielhaftes audio-visuelles Informations-Zuführsystem **100** zum Zuführen und Bereitstellen eines nicht-sequenziellen Zugriffs auf digitale Live-Video-Zuführungen dar-

stellt. Das audio-visuelle Informations-Zuführsystem **100** weist im Allgemeinen einen Koder **101**, einen Video-Server **106**, einen Medien-Datenspeicher (MDS) **112**, eine Datenbank **116**, einen Stream-Server **118**, eine Videopumpe **120** und einen Client **122** auf.

Der Koder

[0031] Der Koder **101** empfängt eine audio-visuelle Eingabe und erzeugt einen digitalen Datenstrom, in dem die audio-visuelle Eingabe gemäß einem bestimmten Format kodiert ist. Es wurden verschiedene Video-Kodierformate entwickelt und sind in der Industrie wohlbekannt. Beispielsweise sind die MPEG-Formate ausführlich in den folgenden internationalen Standards beschrieben: ISO/IEC 13818-1, 2, 3 (MPEG 2) und ISO/IEC 11172-1, 2, 3 (MPEG 1). Dokumente, in denen diese Standards beschrieben sind, (weiterhin bezeichnet als „MPEG-Spezifikationen“), sind verfügbar bei ISO/IEC Copyright Office, Postfach 56, CH-1211 Genf 20, Schweiz. Während sich an dieser Stelle zum Zwecke der Erläuterung auf bestimmte Formate bezogen wird, ist die vorliegende Erfindung nicht auf irgendein spezielles digitales Stream-Format beschränkt.

[0032] Der Koder **101** weist einen Koder/Dekoder (CODEC) **102** und einen Multiplexer (MUX) **104** auf. Der CODEC **102** wandelt visuelle oder audio-visuelle Information von einer Eingabequelle in komprimierte digitale Daten um. Der CODEC **102** kann beispielsweise ein Fraktal-Kompressor oder ein MPEG-Kompressor sein. Zum Zwecke der Darstellung soll angenommen werden, dass die Videoquelle, die mittels des CODEC **102** abgefangen wird, eine Live-Quelle ist, und dass folglich der CODEC **102** das Video mit einfacher Rate (1X) in Bezug auf die Echtzeit kodiert. Jedoch kann die Video-Quelle alternativ eine gespeicherte Video-Quelle sein, die der CODEC **102** mit irgendeiner Rate in Bezug auf die Echtzeit kodiert.

[0033] Der MUX **104** multiplext die komprimierte Audio- und visuelle Information, die vom CODEC **102** erzeugt worden ist, um einen komprimierten Videostrom zu erzeugen. In dem komprimierten Videostrom werden die Daten, die Videorahmen und Töne darstellen, gemäß dem speziellen digitalen Format gemischt und formatiert, das vom Koder **101** unterstützt wird. Die spezifischen Operationen, die während des Mischprozesses durchgeführt werden, variieren basierend auf dem Typ des angewendeten Koders. Beispielsweise kann der Mischprozess das Ermitteln der Reihenfolge und der Anordnung von Abschnitten digitalisierter Töne und Videos im Strom sowie das Einfügen von Metadaten an verschiedenen Punkten innerhalb des Stroms beinhalten. Die Metadaten können beispielsweise die Form von Header-Information annehmen, die den Startpunkt und den Inhalt von „Paketen“ innerhalb des Stroms identifiziert. Der Strom komprimierter audio-visueller Information, der mittels des MUX **104** aufgebaut worden ist, wird vom Koder **101** zum Video-Server **106** mittels eines Kommunikationskanals **128** übertragen.

Steuerinformation

[0034] Der Koder **101** sendet gemäß einem Aspekt des vorliegenden Systems Steuerinformation mittels eines Kommunikationskanals **130** an den Video-Server **106** parallel mit dem Videostrom. Die Steuerinformation, die mittels des Kanals **130** gesendet worden ist, weist spezifische Information darüber auf, wie der Koder **101** den Videostrom aufgebaut hat. Diese Steuerinformation weist Tag-Daten auf, die vom Stream-Server **188** verwendet werden, um einen nicht-sequenziellen Zugriff auf den Videostrom bereitzustellen. Insbesondere kann die Steuerinformation Informationen über den Typ, die Länge und die Grenzen der verschiedenen Rahmen, die in dem Videostrom kodiert sind, ebenso wie Header-Information aufweisen, die die Komprimierungsrate, die Bitrate und andere Typen von Information spezifiziert, die der Video-Server **106** benötigt, um zu ermitteln, wie der Videostrom zu verarbeiten ist.

[0035] Es ist bedeutsam, dass die Erzeugung der Steuerinformation eine minimale zusätzliche Rechenkraft einbezieht, da der MUX **104** das meiste der Information bereits während des Aufbaus des Content-Stroms erzeugt. Insbesondere ordnet der MUX **104** die digitalen Video- und Audiodaten von dem CODEC **102** an und kapselt sie ein. Da der MUX **104** den Inhalt paketiert, kennt der MUX **104** die Inhalte und die Grenzen zwischen den Paketen.

Kommunikation zwischen dem Koder und dem Video-Server

[0036] Während der CODEC **102** typischerweise in einer hartverdrahteten Schaltung implementiert ist, ist der MUX **104** bevorzugt mittels einer programmgesteuerten Schaltung, wie beispielsweise eines Prozessors, implementiert, der programmiert ist, um eine bestimmte Sequenz von Befehlen auszuführen, die in einem Speicher gespeichert sind. Folglich kann der MUX **104** einen Prozessor aufweisen, der ein herkömmliches Multiplex-Programm ausführt, das mit einer Software-Bibliothek gekoppelt worden ist und an diese Aufrufe durch-

führt, wobei mittels dieser Bibliothek die Kommunikation mit dem Video-Server **106** gesteuert wird.

[0037] Alle zwischen dem Koder **101** und dem Video-Server **106** übertragenen Daten werden bevorzugt unter Verwenden eines zuverlässigen Kommunikationsmechanismus' gesendet. Gemäß einem Ausführungsbeispiel wird der Video-Inhalt auf dem Kommunikationskanal **128** als ein einfacher Byte-Strom behandelt und wird mittels eines Leichtgewicht (Light Weight)- sowie zuverlässigen Protokolls übertragen. Beispielsweise ist TOP bei gering belasteten Netzwerken ausreichend. Die Steuerinformation und Metadaten auf dem Kommunikationskanal **130** beinhalten kompliziertere Datentypen und werden mittels eines objektorientierten Protokolls übertragen, wie beispielsweise Common Object Request Broker Architecture Interface Definition Language („CORBA IDL“).

[0038] Die Kommunikation zwischen dem Koder **101** und dem Video-Server **106** erfolgt in Sitzungen. Gemäß einem Ausführungsbeispiel wird eine Sitzung in 3 Phasen durchgeführt: Öffnen (OPEN), Senden (SEND) und Schließen (CLOSE). Die während jeder dieser Phasen durchgeföhrten Operationen sind wie folgt:
 OPEN – jede Bereitstellung, die der Video-Server **106** für Netzwerk- oder Platten-Bandbreite- oder Plattenspeicher-Vorkommnisse durchführen muss. Eine Pipe für die Videostrom-Daten (den „Inhalt“) wird erzeugt.
 SEND TAGS und SEND DATA – diese Aufrufe werden so viele Male durchgeführt, wie der Inhalt kodiert wird. Der Video-Server **106** speichert den gesamten Inhalt unmittelbar auf eine Platte und aktualisiert eine Ende-der-Datei-Position. Tags werden im Speicher gehalten, bis die beigefügten Inhalts-Daten gespeichert sind. Tags werden für eine zusätzliche Zeitdauer gehalten, um zu garantieren, dass eine Suche nach solch einem Tag erfolgreich sein wird, d. h., dass die Videopumpe **120** nicht in Bezug auf die Daten stillstehen wird.
 CLOSE – die Inhalts-Pipe wird abgeschnitten (torn down). Server-Ressourcen werden freigegeben, und inhaltsbezogene Dienste sowie Clients werden benachrichtigt, dass die Zufuhr zu einem normalen statischen Bestandteil des Inhalts geworden ist.

[0039] Der Koder **101** erzeugt parallel Inhalts-Daten und Steuerdaten. Jedoch werden die Steuerdaten, die einem bestimmten Abschnitt des Inhalts zugeordnet sind, nicht notwendigerweise vom Koder **101** zur gleichen Zeit erzeugt, wie der spezielle Inhalts-Abschnitt. Beispielsweise kann der Koder **101** tatsächlich ermitteln, wie das Aufreihen von Inhalts-Rahmen stattfindet, bevor der Koder **101** die Rahmen eigentlich aufreihrt. Unter diesen Umständen können die Steuerdaten, die kennzeichnen, wie die Rahmen aufgereiht worden sind, vom Koder **101** vor den Inhalts-Daten übertragen werden, die die Rahmen enthalten.

Der Video-Server

[0040] Der Video-Server **106** empfängt den Videostrom und die Steuerdaten von dem Koder **101** und bewirkt, dass die Daten im MDS **112** gespeichert werden. Bei dem dargestellten System sendet der Video-Server **106** einen MPEG-Videostrom an den MDS-Server **110**, und der MDS-Server **110** speichert den MPEG-Videostrom in einer MPEG-Datei **134**. Parallel dazu sendet der Video-Server **106** Tag-Information an den MDS-Server **110**, die aus den Steuerdaten extrahiert worden ist, die auf der Leitung **130** empfangen worden sind. Die Tag-Daten werden in einer Tag-Datei **132** auf Platten **114** gespeichert. Der Video-Server **106** kann ferner die Information über den Inhalt, der die Tag-Daten aufweist, senden, um sie in der Datenbank **116** zu speichern.

[0041] Sind die Daten einmal mittels des Video-Servers **106** übertragen, kann irgendein anderes Gerät in dem System, inklusive der Videopumpe **120**, die Tag-Daten verwenden, um einen Zugriff auf den Inhalt zu versuchen, der den Tag-Daten zugeordnet ist. Folglich kann die unmittelbare Übertragung von Tag-Daten an den MDS-Server **110** zu Fehlern führen, wenn die Tag-Daten beispielsweise am Video-Server **106** vor den entsprechenden Inhalts-Daten ankommen. Daher puffert der Video-Server **106** vor dem Senden der Tag-Daten an den MDS-Server **110** jedes Tag-Daten-Item in einem Tag-Puffer **108**, bis es für Geräte, wie beispielsweise eine Videopumpe **120**, sicher ist, auf den Inhalt zuzugreifen, der dem Tag-Daten-Item zugeordnet ist. Die Verwendung des Tag-Puffers **108**, um ein verfrühtes Lesen von Inhalts-Daten zu verhindern, ist ausführlicher weiter unten beschrieben.

Beispielhafte MPEG-Datei

[0042] Bei digitalen audio-visuellen Speicherformaten, ob komprimiert oder nicht, werden Zustandsmaschinen und Pakete verschiedener Strukturen verwendet. Die an dieser Stelle beschriebenen Techniken werden auf all diese Speicherformate angewendet. Während die vorliegende Erfindung nicht auf irgendein bestimmtes digitales audio-visuelles Format beschränkt ist, soll die MPEG 2-Transportdatei-Struktur zu Darstellungszwecken beschrieben werden.

[0043] Bezugnehmend auf [Fig. 2A](#) stellt diese die Struktur einer MPEG 2-Transportdatei **134** in größerem Detail dar. Die Daten innerhalb der MPEG-Datei **134** sind in drei Schichten paketiert: einer Programm-Elementarstrom („PES“)-Schicht, einer Transportschicht und einer Videoschicht. Diese Schichten sind ausführlich in den MPEG 2-Spezifikationen beschrieben. In der PES-Schicht besteht die MPEG-Datei **134** aus einer Sequenz von PES-Paketen. In der Transportschicht besteht die MPEG-Datei **134** aus einer Sequenz von Transportpaketen. In der Videoschicht besteht die MPEG-Datei **134** aus einer Sequenz von Bildpaketen. Jedes Bildpaket enthält die Daten für einen Videorahmen.

[0044] Jedes PES-Paket weist einen Header auf, der die Länge und die Inhalte des PES-Paketes identifiziert. Bei dem dargestellten Beispiel enthält ein PES-Paket **250** einen Header **248**, gefolgt von einer Sequenz von Transportpaketen **251-262**. Die PES-Paket-Grenzen fallen mit gültigen Transportpaket-Grenzen zusammen. Jedes Transportpaket enthält ausschließlich einen Datentyp. Bei dem dargestellten Beispiel enthalten die Transportpakete **251**, **256**, **258**, **259**, **260** und **262** Videodaten. Die Transportpakete **252**, **257** und **261** enthalten Audio-Daten. Das Transportpaket **253** enthält Steuer-Daten. Das Transportpaket **254** enthält Zeitsteuerungs-Daten. Das Transportpaket **255** ist ein Auffüll-Paket.

[0045] Jedes Transportpaket weist einen Header auf. Der Header weist einen Paket-ID („PID“) für das Paket auf. Pakete, denen der PID 0 zugeordnet ist, sind Steuerpakete. Beispielsweise kann dem Paket **253** der PID 0 zugeordnet sein. Andere Pakete, inklusive andere Steuerpakete, werden in den PID 0-Paketen referenziert. Insbesondere weisen die PID 0-Steuerpakete Tabellen auf, die die Pakettypen der Pakete aufweisen, die den PID 0-Steuerpaketen unmittelbar folgen. Für alle Pakete, die keine PID 0-Steuerpakete sind, enthalten die Header PIDs, die als Zeiger auf die Tabelle dienen, die in dem PID 0-Steuerpaket enthalten ist, das den Paketen unmittelbar vorausgeht. Beispielsweise würde der Datentyp, der in einem Paket mit einem PID **100** enthalten ist, durch das Untersuchen des Eintrags, der dem PID **100** zugeordnet ist, in der Tabelle des PID 0-Steuerpaketes ermittelt, das dem Paket am nächsten vorgelagert ist.

[0046] In der Videoschicht wird die MPEG-Datei **134** gemäß den Grenzen der Rahmen-Daten aufgeteilt. Wie oben erwähnt, gibt es keine Korrelation zwischen den Grenzen der Daten, die Videorahmen repräsentieren, und den Transportpaket-Grenzen. Bei dem dargestellten Beispiel sind die Rahmen-Daten für einen Videorahmen „F“ positioniert, wie mittels Klammern **270** gekennzeichnet. Insbesondere sind die Rahmen-Daten für den Rahmen „F“ von einem Punkt **280** innerhalb des Video-Paketes **251** bis zum Ende des Video-Paketes **251**, im Video-Paket **256** und vom Beginn des Video-Paketes **258** bis zu einem Punkt **282** innerhalb des Video-Paketes **258** positioniert. Daher stellen die Punkte **280** und **282** die Grenzen für das Bildpaket für den Rahmen „F“ dar. Die Rahmen-Daten für einen zweiten Videorahmen „G“ sind positioniert, wie mittels Klammern **272** gekennzeichnet. Die Grenzen für das Bildpaket für den Rahmen „G“ sind mittels der Klammern **276** gekennzeichnet.

[0047] Strukturen, die analog denen sind, wie oben für die MPEG 2-Transport-Ströme beschrieben, gibt es auch bei anderen digitalen audio-visuellen Speicherformaten, inklusive MPEG 1, Quicktime, AVI, Proshare und H.261-Formaten. Bei dem bevorzugten Ausführungsbeispiel werden Kennzeichner für Video-Zugriffspunkte, Zeitstempel, Datei-Positionen usw. derart gespeichert, dass auf die vielen digitalen audio-visuellen Speicherformate vom gleichen Server zugegriffen werden kann, um gleichzeitig unterschiedlichen Clients mit einer großen Vielfalt von Speicherformaten zu dienen. Bevorzugt sind all die formatspezifischen Informationen und Techniken im Tag-Generator und im Stream-Server integriert. All die anderen Elemente des Servers sind formunabhängig.

Beispielhafte Tag-Datei

[0048] Die Inhalte einer beispielhaften Tag-Datei **132** sollen nun unter Bezugnahme auf [Fig. 2B](#) gezeigt werden. In [Fig. 2B](#) weist die Tag-Datei **132** einen Dateityp-Identifizierer **202**, einen Längen-Kennzeichner **204**, einen Bitraten-Kennzeichner **206**, einen Spieldauer-Kennzeichner **208**, einen Rahmennummer-Kennzeichner **210**, eine Stream-Zugriffsinformation **212** und einen initialen MPEG-Zeit-Offset **213** auf. Der Dateityp-Identifizierer **202** kennzeichnet das physikalische Wrapping in der MPEG-Datei **134**. Beispielsweise würde der Dateityp-Identifizierer **202** anzeigen, ob die MPEG-Datei **134** eine MPEG 2- oder eine MPEG 1-Datei ist.

[0049] Der Längen-Kennzeichner **204** zeigt die Länge der MPEG-Datei **134** an. Der Bitraten-Kennzeichner **206** zeigt die Bitrate an, mit der die Inhalte der MPEG-Datei **134** an einen Client während einer Wiedergabe zu senden sind. Der Spieldauer-Kennzeichner **208** spezifiziert die Zeitspanne in Millisekunden, die erforderlich ist, die gesamten Inhalte der MPEG-Datei **134** während eines normalen Wiedergabebetriebs wiederzugeben. Der Rahmennummer-Kennzeichner **210** zeigt die Gesamtanzahl der Rahmen an, die in der MPEG-Datei **134** dargestellt sind.

[0050] Die Stream-Zugriffsinformation **212** ist eine Information, die erforderlich ist, um auf die Video- und Audio-Ströme zuzugreifen, die in der MPEG-Datei **134** gespeichert sind. Die Stream-Zugriffsinformation **212** weist einen Video-Elementar-Stream-ID und einen Audio-Elementar-Stream-ID auf. Für MPEG 2-Dateien weist die Stream-Zugriffsinformation **212** ferner einen Video-PID und einen Audio-PID auf. Der Tag-Datei-Header kann ferner andere Information enthalten, die verwendet werden kann, um andere Merkmale zu implementieren.

[0051] Zusätzlich zu der oben beschriebenen eigenen Information enthält die Tag-Datei **132** einen Eintrag für jeden Rahmen innerhalb der MPEG-Datei **134**. Der Eintrag für einen Videorahmen weist Informationen über den Zustand der verschiedenen MPEG-Schichten in Bezug auf die Position der Daten auf, die den Rahmen darstellen. Für eine MPEG 2-Datei weist jeder Eintrag den Zustand der MPEG 2-Transport-Zustandsmaschine, den Zustand der Programm-Elementarstrom-Zustandsmaschine und den Zustand der Video-Zustandsmaschine auf. Für eine MPEG 1-Datei weist jeder Eintrag den aktuellen Zustand des Pack-System-MPEG-Stroms und den Zustand der Video-Zustandsmaschine auf.

[0052] Der Tag-Datei-Eintrag **214** stellt in größerem Detail die Tag-Information dar, die für einen einzelnen MPEG 2-Videorahmen „F“ gespeichert ist. Bezüglich des Zustands der Programm-Elementarstrom-Zustandsmaschine weist der Tag-Eintrag **214** die Information auf, wie in Tabelle 1 gezeigt.

Tabelle 1

Daten	Bedeutung
PES-Offset am Beginn des Bildes 217	der Offset des ersten Bytes für den ersten Rahmen „F“ innerhalb des PES-Paketes, das die Rahmen-Daten für den Rahmen „F“ enthält
PES-Offset am Ende des Bildes 219	der Offset zwischen dem letzten Byte in den Rahmen-Daten für den Rahmen „F“ und dem Ende des PES-Paketes, in dem sich die Rahmen-Daten für den Rahmen „F“ befinden

[0053] Bezüglich des Zustands der Video-Zustandsmaschine weis der Tag-Eintrag **214** die Information auf, wie in Tabelle 2 gezeigt.

Tabelle 2

Daten	Bedeutung
Bildgröße 220	die Größe des Paketes für den Rahmen „F“
Start-Position 226	die Position des ersten Bytes der dem Rahmen „F“ entsprechenden Daten innerhalb der MPEG-Datei
Zeitwert 228	die Zeit in Bezug auf den Beginn des Films, in der der Rahmen „F“ während einer normalen Wiedergabe der MPEG-Datei 134 angezeigt würde,
Rahmen-Typ 232	die Technik, die verwendet wird, um den Rahmen zu kodieren (beispielsweise I-Rahmen, P-Rahmen oder B-Rahmen)
Zeitsteuerungs-Puffer-Information 238	kennzeichnet, wie voll der Puffer des Dekoders ist (wird an den Dekoder gesendet, um zu ermitteln, wann die Information aus dem Puffer heraus transportiert werden soll, um neu ankommende Information zu empfangen)

[0054] Unter Bezugnahme auf den Zustand der Transportschicht-Zustandsmaschine weist der Tag-Eintrag **214** die Information auf, wie in Tabelle 3 gezeigt.

Tabelle 3

Daten	Bedeutung
Start-Offset 234	der Abstand zwischen dem ersten Byte in den Rahmen-Daten und dem Beginn des Transport-Paketes, in dem das erste Byte vorkommt
# Nicht-Video-Pakete 222	die Anzahl der Nicht-Video-Pakete (d. h. Audio-Pakete, Auffüll-Pakete, Steuerpakete und Zeitsteuerungs-Pakete), die innerhalb des Bild-Paketes für den Rahmen „F“ positioniert sind
# Auffüll-Pakete 224	die Anzahl der Auffüll-Pakete, die innerhalb des Bild-Paketes für den Rahmen „F“ positioniert sind
Ende-Offset 236	der Abstand zwischen dem letzten Byte in den Rahmen-Daten und dem Ende des Paketes, in dem das letzte Byte vorkommt
aktueller Kontinuitäts-Zähler 215	der Kontinuitäts-Wert, der dem Rahmen „F“ zugeordnet ist
Diskontinuitäts-Flag 230	kennzeichnet, ob es eine zeitliche Diskontinuität zwischen dem Rahmen „F“ und dem Rahmen gibt, das in dem vorangegangenen Tag-Eintrag dargestellt ist

[0055] Angenommen, dass der Eintrag **214** beispielsweise für den Rahmen „F“ von [Fig. 2B](#) ist. Die Größe **220**, die dem Rahmen „F“ zugeordnet ist, würden die Bits sein, die von Klammern **274** umgeben sind. Die Anzahl **222** von Nicht-Video-Paketen würde 5 sein (Pakete **252**, **253**, **254**, **255** und **257**). Die Anzahl **224** von Auffüll-Paketen würde 1 sein (Paket **255**). Die Anfangsposition **226** würde der Abstand zwischen dem Start der MPEG-Datei **134** und dem Punkt **280** sein. Der Start-Offset **234** würde der Abstand zwischen dem Start des Paketes **251** und dem Punkt **280** sein. Der Ende-Offset **236** würde der Abstand zwischen dem Punkt **282** und dem Ende des Paketes **258** sein.

[0056] Die Tag-Information, die für jeden Rahmen in einer MPEG 1-Datei erzeugt wird, ist in [Fig. 2C](#) dargestellt. Bezugnehmend auf [Fig. 2C](#) weist der Eintrag **214** Daten auf, die den Zustand dreier Zustandsmaschinen anzeigen: einer System-Zustandsmaschine, einer Paketierungs-Zustandsmaschine und einer Video-Zustandsmaschine. Insbesondere weist der Tag-Eintrag **214** die Information auf, wie in Tabelle 4 gezeigt.

Tabelle 4

Daten	Bedeutung
Menge der Nicht-Video-Daten 221	die Menge der Nicht-Video-Daten (in Byte), die innerhalb der Grenzen des Starts und des Endes der Rahmen-Daten für den Rahmen „F“ enthalten sind
Menge der Auffüll-Daten 223	die Menge der Auffüll-Daten (in Byte), die innerhalb der Grenzen des Starts und des Endes der Rahmen-Daten für den Rahmen „F“ enthalten sind
PACK-Offset am Start 225	der Offset zwischen der Grenze des Starts der Rahmen-Daten für den Rahmen „F“ und dem Start des Pack-Paketes, das die Grenze des Starts für den Rahmen enthält
PACK, verbleibend am Beginn 227	der Abstand zwischen der Grenze des Starts für den Rahmen „F“ und dem Ende des Pack-Paketes, das die Grenze des Starts des Rahmens „F“ enthält
PACK-Offset am Ende 229	der Offset zwischen der Grenze des Endes für den Rahmen „F“ am Anfang des Pack-Paketes, das die Grenze des Endes für den Rahmen „F“ enthält
PACK, verbleibend am Ende 231	der Abstand zwischen der Grenze des Endes für den Rahmen „F“ und dem Ende des Pack-Paketes, das die Grenze des Endes des Rahmens „F“ enthält
Bildgröße 233	der Abstand (in Byte) zwischen der Grenze des Beginns für den Rahmen „F“ und der Grenze des Endes für den Rahmen „F“
Bild-Start-Position 235	der Abstand zwischen dem Beginn der MPEG 1-Datei und der Grenze des Beginns für den Rahmen „F“
Bild-Ende-Position 237	die Position der Grenze des Endes für den Rahmen „F“ in Bezug auf den Beginn der MPEG 1-Datei
Rahmen-Typ 239	die Technik, die verwendet worden ist, um die Daten zu kodieren, die den Rahmen „F“ darstellen
Zeitwert 241	die Zeit in Bezug auf den Beginn des Films, in der der Rahmen „F“ während einer normalen Wiedergabe der MPEG-Datei 134 angezeigt würde
Zeitsteuerungs-Puffer-Information 243	kennzeichnet, wie voll der Puffer des Dekoders ist (wird an den Dekoder gesendet, um zu ermitteln, wann die Information aus dem Puffer heraus transportiert werden soll, um neu ankommende Information zu empfangen)

[0057] Die Tag-Information weist Daten auf, die den Zustand der relevanten Zustandsmaschinen am Beginn von Videorahmen anzeigen. Jedoch unterscheiden sich die Zustandsmaschinen, die mit anderen digitalen audio-visuellen Formaten angewendet werden, von denen, wie oben beschrieben, ebenso wie sich die Zustandsmaschinen, die beim MPEG 1-Format angewendet werden, von denen bei MPEG 2 unterscheiden. Folglich wird die spezifische Tag-Information, die für jeden Rahmen des Videos gespeichert ist, basierend auf dem digitalen Audio-Videoformat der Datei variieren, der sie zugeordnet ist.

Der MDS

[0058] Der MDS **112** weist einen MDS-Server **110** und einen oder mehrere Nichtflüchtig-Speicher-Einrichtungen, wie beispielsweise Platten **114**, auf. Bei dem dargestellten Ausführungsbeispiel wird die MPEG-Datei **134** über viele Platten **114** hinweg gespeichert, um die Fehlertoleranz des Systems zu erhöhen. Betrachtet wird beispielsweise ein Mehrfach-Plattensystem **300**, wie in **Fig. 3** dargestellt. Das System **300** weist N+1 Platten-

laufwerke auf. Eine MPEG-Datei wird auf N der N+1 Platten gespeichert. Die MPEG-Datei wird in Abschnitte **350**, **352**, **354** und **356** aufgeteilt. Jeder Abschnitt wird in N Blöcke aufgeteilt, wobei N die Anzahl der Platten ist, die verwendet werden, um die MPEG-Datei zu speichern. Jede Platte speichert einen Block eines gegebenen Abschnitts.

[0059] Bei dem dargestellten Beispiel weist der erste Abschnitt **350** der MPEG-Datei Blöcke **310**, **312** und **314** auf, die auf Platten **302**, **304** bzw. **306** gespeichert sind. Der zweite Abschnitt **352** weist Blöcke **316**, **318** und **320** auf, die auf Platten **302**, **304** bzw. **306** gespeichert sind. Der dritte Abschnitt **354** weist Blöcke **322**, **324** und **326** auf, die auf Platten **302**, **304** bzw. **306** gespeichert sind. Der vierte Abschnitt **356** weist Blöcke **328**, **330** und **332** auf, die auf Platten **302**, **304** bzw. **306** gespeichert sind.

[0060] Die Platte **308**, die nicht verwendet wird, um die MPEG-Datei zu speichern, wird verwendet, um Prüf-Bits zu speichern. Jeder Satz von Prüf-Bits entspricht einem Abschnitt der MPEG-Datei und wird basierend auf den verschiedenen Blöcken aufgebaut, die zu dem entsprechenden Abschnitt gehören. Beispielsweise entsprechen die Prüfbits **334** dem Abschnitt **350** und werden mittels Durchführen einer Exklusiv-ODER (XOR)-Operation auf all den Blöcken im ersten Abschnitt **350** erzeugt. Auf die gleiche Weise sind die Prüf-Bits **336**, **338** und **340** die Ergebnisse einer Exklusiv-ODER-Operation, die auf all den Blöcken im Abschnitt **352**, **354** bzw. **356** durchgeführt worden ist.

[0061] Das System **300** weist eine höhere Fehlertoleranz als ein Einzel-Plattensystem in der Hinsicht auf, dass, wenn eine Platte im System aufhört, korrekt zu arbeiten, die Inhalte der kaputten Platte basierend auf den Inhalten der verbliebenen Platten rekonstruiert werden können. Wenn beispielsweise die Platte **304** aufhört zu funktionieren, können die Inhalte des Blocks **312** basierend auf den verbliebenen Blöcken im Abschnitt **350** und den Prüf-Bits **334**, die dem Abschnitt **350** zugeordnet sind, rekonstruiert werden.

[0062] Auf die gleiche Weise kann der Block **318** basierend auf den verbliebenen Blöcken im Abschnitt **352** und den Prüf-Bits **336**, die dem Abschnitt **352** zugeordnet sind, aufgebaut werden. Diese Fehlererkennung und -Korrekturtechnik ist allgemein bekannt als „redundantes Feld preisgünstiger Platten“ (Redundant Array of Inexpensive Disk – RAID).

[0063] Während einer Echtzeit-Wiedergabe unter Verwenden von RAID liest eine Videopumpe **120** die MPEG-Datei auf einer Abschnitt-für-Abschnitt-Basis und verarbeitet sie, so dass all die Information verfügbar ist, um irgendwelche Daten, die von einer Platte fehlerhaft gelesen worden sind, zu rekonstruieren. Techniken zum Durchführen von RAID in Echtzeit sind beschrieben im US-Patent Nr. US-A-5,623,595, betitelt mit „Method and Apparatus for Transparent, Real Time Reconstruction of Corrupted Data In A Redundant Array Data Storage System“.

[0064] Während normaler Wiedergabe-Operationen ist genügend Zeit, um die Platten-Zugriffe durchzuführen, die erforderlich sind, um einen gesamten Abschnitt zu lesen, während die Daten vom vorherigen Abschnitt im MPEG-Datenstrom übertragen werden. Jedoch werden bei den Operationen des schnellen Vorspulens und des schnellen Zurückspulens weniger als all die Daten in jedem Abschnitt im MPEG-Datenstrom gesendet. Da weniger Daten gesendet werden, nimmt die Übertragungszeit der Daten weniger Zeit in Anspruch. Folglich ist weniger Zeit verfügbar, um den folgenden Abschnitt zu lesen und zu verarbeiten.

[0065] Beispielsweise wird angenommen, dass lediglich ein Rahmen X des Abschnitts **350** zur Anzeige während einer Operation des schnellen Vorspulens ausgewählt war. Während der Zeit, die in Anspruch genommen wird, das Segment für den Rahmen X zu übertragen, werden die Daten für den nächsten ausgewählten Rahmen Y gelesen und verarbeitet. Es wird angenommen, dass der nächste Rahmen Y in Abschnitt **352** positioniert ist. Wenn die MPEG-Datei auf einer Abschnitts-Basis gelesen und verarbeitet wird (erforderlich für RAID), dann werden all die Blöcke im Abschnitt **352** während der Übertragung des einzelnen Rahmens X gelesen und verarbeitet. Selbst wenn es möglich ist, all die Blöcke im Abschnitt **352** in der zugewiesenen Zeit zu lesen und zu verarbeiten, kann es nicht wünschenswert sein, dies wegen der Ressourcen, die beim Durchführen der erforderlichen Plattenzugriffe verbraucht würden, so durchzuführen.

[0066] Im Licht des Vorangegangenen nutzt die Videopumpe **120** nicht RAID während der Operationen des schnellen Vorspulens und des schnellen Zurückspulens. Stattdessen liest, verarbeitet und überträgt die Videopumpe **120** nur die Daten, die in den Befehlen gekennzeichnet sind, die sie vom Stream-Server **118** empfängt. Daher würden bei dem oben gegebenen Beispiel lediglich die Rahmen-Daten für den Rahmen Y während der Übertragung des Segments für den Rahmen X gelesen und verarbeitet. Unter Umgehung von RAID während der Operationen des schnellen Vorspulens und des schnellen Zurückspulens bleibt die Platten-Band-

breite auf der gleichen Stufe oder unter der Stufe, die während des normalen Wiedergabebetriebs verwendet wird.

[0067] Da RAID nicht während der Operationen des Echtzeit-Vorspulens und des Echtzeit-Zurückspulens verwendet wird, können fehlerhafte Daten während dieser Operationen nicht rekonstruiert werden. Folglich, wenn die Videopumpe **120** erfasst, dass die Daten für einen selektierten Rahmen zerstört oder nicht verfügbar sind, verwirft die Videopumpe **120** das gesamte Segment, das dem problematischen Rahmen zugeordnet ist. Deshalb werden die Präfix- und Suffix-Daten für den Rahmen ebenfalls nicht gesendet, wenn die dem Rahmen zugeordneten Daten nicht gesendet werden können. Jedoch werden einige Auffüll-Pakete, die mit den Präfix- oder Suffix-Daten zu senden sind, weiterhin gesendet.

[0068] Mittels des Sendens von Daten in Gesamt-„Segmenten“ wird die Konformität mit dem digitalen Audio-Visuell-Format beibehalten. Bei einem Ausführungsbeispiel wird die Videopumpe **120** Auffüll-Pakete senden, um die Leitung aufzufüllen, um die korrekte Darstellungsrate beizubehalten. Bei dem bevorzugten Ausführungsbeispiel ist dieses Verhalten vom Client auswählbar.

Data-Striping

[0069] Die oben beschriebenen RAID-Techniken verbessern sowohl den Durchsatz (da alle Daten von allen Platten in einem Feld parallel gelesen werden) als auch die Zuverlässigkeit (infolge der Fehlerkorrektur). Um den Durchsatz weiter zu erhöhen, kann RAID in Verbindung mit Data-Striping verwendet werden. Unter Verwenden von Data-Striping werden Segmente von logisch hintereinander liegenden Daten auf viele physikalische Geräte (oder Sätze von physikalischen Geräten) in einer Round-Robin-Weise geschrieben. Die Menge gespeicherter Daten auf jedem Speicherelement in der Round-Robin-Sequenz wird als ein „Stripe“ bezeichnet. Wenn jedes Speicherelement in der Round-Robin-Sequenz ein Feld von RAID-Platten ist, wird jedes Datensegment als RAID-Stripe bezeichnet.

[0070] [Fig. 3B](#) stellt ein System dar, bei dem Data-Striping in Verbindung mit RAID verwendet wird. Das System in [Fig. 3B](#) ist gleich dem von [Fig. 3A](#) mit der Ausnahme, dass jede der Platten in [Fig. 3A](#) durch einen Folge von M Platten ersetzt ist. Daher ist die Platte **302** durch Platten **302-1** bis **302-M** ersetzt worden. Die Segment-Abschnitte, die auf Platten **302** gespeichert worden sind, sind auf Platten **302-1** bis **302-M** in einer sequenziellen Round-Robin-Weise gespeichert. Beispielsweise wird angenommen, dass die MPEG-Datei in 50 Segmente aufgeteilt ist und dass die Platte **302** durch 25 Platten ersetzt worden ist. Unter diesen Umständen würde die Platte **302-1** den ersten Abschnitt der Segmente 1 und 26 speichern. Die Platte **302-2** würde den ersten Abschnitt der Segmente 2 und 27 speichern usw.

[0071] Durch das Data-Striping wird der Durchsatz erhöht, da verschiedene Prozesse von verschiedenen Platten-Feldern parallel lesen können. Beispielsweise kann eine Datenpumpe das erste Segment einer MPEG-Datei aus dem RAID-Feld lesen, das die Platten Disk_1,1 bis Disk_1,N+1 aufweist, während eine andere Datenpumpe konkurrenzlos das zweite Segment der gleichen MPEG-Datei aus dem RAID-Feld liest, das die Platten Disk_2,1 bis Disk_2,N+1 aufweist.

[0072] Aus Durchsatz-Leistungs-Gründen erfolgt das Lesen und das Schreiben in diskreten Blöcken, typischerweise in Platten-RAID-Stripes. Bei einem typischen digitalen Video-Zuführsystem ist jede Zugriffseinheit 256 kByte oder 2 Megabit, und der Inhalt ist ein 2 MB/s-MPEG. Folglich entspricht jedes RAID-Stripe ungefähr einer Sekunde eines Videos, obwohl diese leicht im Bereich von ungefähr 0,2 bis 10 Sekunden pro Stripe abhängig von der Inhalts-Bitrate und der Serverkonfiguration variieren kann.

Der Client

[0073] Das audio-visuelle Informations-Zuführsystem **100** enthält einen oder mehrere Clients, wie beispielsweise den Client **122**. Der Client **122** repräsentiert im Allgemeinen Geräte, die eingerichtet sind, audio-visuelle Information zu dekodieren, die in einem Strom digitaler audio-visueller Daten enthalten ist. Beispielsweise kann der Client **122** eine Set-Top-Wandlerbox sein, die mit einer Ausgabe-Anzeige, wie beispielsweise einem Fernsehgerät, gekoppelt ist. Der Client **122** weist einen Dekoder **126** zum Dekodieren eines digitalen Datenstroms und eine Steuereinheit **124** zum Übertragen von Information zu dem Stream-Server **118** auf.

[0074] Der Stream-Server **118** ist befähigt, Informationen von Client **122** mittels eines Steuernetzwerks **140** zu empfangen. Das Steuernetzwerk **140** kann irgendein Netzwerk sein, das die Kommunikation zwischen zwei oder mehr Geräten ermöglicht. Beispielsweise kann das Steuernetzwerk **140** ein Netzwerk hoher Bandbreite,

eine X.25-Schaltung oder eine Electronic Industry Association (EIA) **232** (RS-232) serielle Leitung sein.

[0075] Der Client **122** kommuniziert mit dem Stream-Server **118** und der Datenbank **116** mittels des Steuernetzwerks **140**. Beispielsweise kann der Client **122** eine Anfrage an die Datenbank **116** senden, dabei eine Information abfragend, was zurzeit zum Anschauen verfügbar ist. Die Datenbank **116** antwortet auf das Senden der angeforderten Information zurück an den Client **122**. Der Nutzer des Clients **122** kann dann eine Ansicht einer bestimmten audio-visuellen Arbeit auswählen, beginnend an einer bestimmten Position und mit einer bestimmten Geschwindigkeit. Der Client **122** überträgt Anforderungen, um die Übertragung von audio-visuellen Datenströmen und von Steuerinformation zu initiieren, um im Stream-Server **118** die Wiedergabe digitaler audio-visueller Übertragungen mittels des Netzwerks **140** zu bewirken.

Die Videopumpe und der Stream-Server

[0076] Die Videopumpe **120** ist mit dem Stream-Server **118** gekoppelt und empfängt Befehle von dem Stream-Server **118**. Die Videopumpe **120** ist mit den Platten **114** derart gekoppelt, dass die Videopumpe **120** Daten speichert und von den Platten **114** abfragt.

[0077] Zusätzlich zum Kommunizieren mit dem Stream-Server **118** empfängt der Client **122** Informationen von der Videopumpe **120** mittels eines Netzwerks **150** hoher Bandbreite. Das Netzwerk **150** hoher Bandbreite kann irgendein Typ einer schaltungsartigen Netzwerkverbindung sein, die befähigt ist, große Datenmengen zu übertragen. Eine schaltungsartige Netzwerkverbindung ist derart konfiguriert, dass das Ziel der Daten durch das darunter liegende Netzwerk garantiert wird, und nicht durch das Übertragungsprotokoll. Beispielsweise kann das Netzwerk **150** hoher Bandbreite eine asynchrone Übertragungsmodus (ATM)-Schaltung oder ein physikalischer Typ einer Leitung sein, wie beispielsweise eine T1- oder E1-Leitung. Zusätzlich können bei dem Netzwerk **150** hoher Bandbreite ein faseroptisches Kabel, Twisted Pair-Leitungen, Koaxialkabel oder ein kabelloses Kommunikationssystem, wie beispielsweise ein Mikrowellen-Kommunikationssystem, verwendet werden.

[0078] Das Netzwerk **150** kann alternativ ein Netzwerk relativ geringer Bandbreite oder eine Kombination von Kommunikationsmedien hoher und geringer Bandbreite sein. Beispielsweise kann ein Abschnitt des Netzwerks **150** eine ATM-Schaltung relativ hoher Bandbreite aufweisen, während netzabwärts ein Gerät relativ niedriger Bandbreite, wie beispielsweise ein 28.8K-Modem, verwendet wird, um vom Netzwerk die Videoinformation dem Client **122** zuzuführen.

[0079] Das audio-visuelle Informations-Zuführsystem **100** erlaubt es einem Server, wie beispielsweise der Videopumpe **120**, große Datenmengen von den Platten **114** mittels des Netzwerks **150** hoher Bandbreite an den Client **122** mit minimalem Overhead zu übertragen. Zusätzlich erlaubt das audio-visuelle Informations-Zuführsystem **100** dem Client **122**, Anforderungen zu dem Stream-Server **118** unter Verwenden eines Standard-Netzwerkprotokolls mittels des Steuernetzwerks **140** zu übertragen. Bei einem bevorzugten Ausführungsbeispiel ist das darunter liegende Protokoll für das Netzwerk **150** hoher Bandbreite und für das Steuernetzwerk **140** das gleiche. Der Stream-Server **118** kann aus einem einzigen Computersystem oder aus einer Mehrzahl von Computergeräten bestehen, die als Server konfiguriert sind. Auf die gleiche Weise kann die Videopumpe **120** aus einer einzigen Server-Einrichtung bestehen oder kann eine Mehrzahl solcher Server aufweisen.

[0080] Um einen digitalen audio-visuellen Datenstrom aus einer bestimmten digitalen audio-visuellen Datei zu empfangen, überträgt der Client **122** eine Anforderung an den Stream-Server **118**. Als Antwort auf die Anforderung überträgt der Stream-Server **118** Befehle an die Videopumpe **120**, um die Videopumpe **120** dazu zu bringen, den angeforderten digitalen audio-visuellen Datenstrom an den Client zu übertragen, der den digitalen audio-visuellen Datenstrom angefordert hat. Für Live-Zuführungen wird der Videoserver **106** den Videostrom in die Videodatei zur gleichen Zeit speichern, zu der die Videopumpe **120** einen Videostrom von der Datei **134** an den Client **122** sendet.

[0081] Die von dem Stream-Server **118** an die Videopumpe **120** übertragenen Befehle weisen Steuerinformation auf, die für die Client-Anforderung spezifisch ist. Beispielsweise identifiziert die Steuerinformation die gewünschte digitale audio-visuelle Datei, den Start-Offset der gewünschten Daten innerhalb der digitalen audio-visuellen Datei und die Adresse des Client. Um einen gültigen digitalen audio-visuellen Strom an dem spezifizierten Offset zu erzeugen, sendet der Stream-Server **118** ferner „Präfix-Daten“ an die Videopumpe **120** und fordert die Videopumpe **120** an, die Präfix-Daten an den Client zu senden. Wie weiter unten ausführlicher beschrieben wird, sind Präfix-Daten Daten, mit denen der Client vorbereitet wird, digitale audio-visuelle Daten von der spezifizierten Position in der digitalen audio-visuellen Datei zu empfangen.

[0082] Die Videopumpe **120** beginnt nach dem Empfangen der Befehle und der Steuerinformation von dem Stream-Server **118**, digitale audio-visuelle Daten von der spezifizierten Position in der spezifizierten digitalen audio-visuellen Datei auf den Platten **114** abzufragen. Zum Zweck der Erläuterung soll angenommen werden, dass das audio-visuelle Informations-Zuführsystem **100** audio-visuelle Information gemäß einem oder mehrerer der MPEG-Formate liefert. Folglich wird die Videopumpe **120** die audio-visuellen Daten von einer MPEG-Datei **134** auf den Platten **114** abfragen.

[0083] Die Videopumpe **120** überträgt die Präfix-Daten an den Client und überträgt dann ruckfrei die MPEG-Daten zu dem Client, die von den Platten **114** abgefragt worden sind, beginnend an der spezifizierten Position. Die Präfix-Daten weisen einen Paket-Header auf, mittels dessen, wenn gefolgt von den MPEG-Daten, die an einer spezifizierten Position positioniert sind, ein MPEG-konformes Übertragungspaket erzeugt wird. Die Daten, die dem ersten Paket folgen, werden sequenziell von der MPEG-Datei **134** abgefragt und bilden deshalb eine Folge MPEG-konformer Pakete. Die Videopumpe **120** überträgt diese Pakete an den anfordernden Client mittels des Netzwerks **150** hoher Bandbreite.

[0084] Der anfordernde Client empfängt den MPEG-Datenstrom, beginnend mit den Präfix-Daten. Der Client dekodiert den MPEG-Datenstrom, um die audio-visuelle Sequenz wiederzugeben, die in dem MPEG-Datenstrom dargestellt ist.

Verhindern eines verfrühten Lesens

[0085] Wenn der Client **122** einen MPEG-Strom zur gleichen Zeit abspielt, zu der der MPEG-Strom vom Koder **101** erzeugt wird, sollten Vorkehrungen getroffen werden, um sicherzustellen, dass der Client **122** nicht stehen bleibt (da er das Ende der gültigen Inhalts-Daten erreicht hat) oder dass er ungültige Daten abspielt (weil er hinter das Ende der aktuell verfügbaren Inhalts-Daten gelesen hat). Wenn die Videopumpe **120** von den Platten **114** einen Stripe verfrüht liest, wird die Videopumpe **120** ungültige Daten an den Client **122** senden, was zu einer Anzeige von nicht erzieltem Inhalt oder von Müll (unsäuberem Inhalt) führt. Solch ein verfrühtes Lesen tritt beispielsweise auf, wenn ein Nutzer die Anzeige eines Abschnitts des Videostroms anfordert, der noch nicht auf den Platten **114** gespeichert ist. Um dies zu verhindern, wird eine Ende-der-Datei-Information für die MPEG-Datei **134** verwaltet, um das aktuelle Ende der Datei **134** zu kennzeichnen. Wenn mehrere Inhalts-Daten zu der Datei **134** hinzugefügt worden sind, wird die Ende-der-Datei-Information so aktualisiert, dass auf die neuen Daten zugegriffen werden kann.

[0086] Ein Ansatz, verfrühtes Lesen zu verhindern, ist die wiederholte Aktualisierung einer Inhalts-Tabelle auf den Platten **114** mit einem neuen Ende-der-Datei-Wert, und dass die Videopumpe **120** diesen Wert prüft, bevor die Stripes von den Platten **114** gelesen werden. Der MDS-Server **110** aktualisiert das Ende der Datei, um zu kennzeichnen, dass die Inhalts-Datei **134** neuen Inhalt aufweist, nur, nachdem geprüft worden ist, dass der neue Inhalt erfolgreich auf den Platten **114** gespeichert ist. Unglücklicherweise führt diese Technik zu einer Schwankung in der Latenzzeit der Aktualisierungen, die schwer vorherzusagen ist, es sei denn, es wird garantiert, dass die Ende-der-Datei-Information in einem dynamischen Speicher gehalten wird.

[0087] Ein anderer Ansatz, um ein verfrühtes Lesen zu verhindern, ist für den MDS-Server **110**, die neue Ende-der-Datei-Information aktiv an alle Prozesse zu übertragen, die den Inhalt lesen. Deshalb speichert der MDS-Server **110** die Inhalts-Daten in die Datei **134** auf den Platten **114**, wartet auf eine Verifikation, dass der Inhalt gespeichert ist, und überträgt dann Nachrichten, die die Existenz des neuen gespeicherten Inhalts anzeigen, an alle Prozesse, die die Inhalts-Daten lesen (beispielsweise Videopumpe **120**). Der MDS-Server **110** kann solche Ende-der-Datei-Benachrichtigungs-Nachrichten periodisch (beispielsweise alle 5 Sekunden) oder nach dem erfolgreichen Speichern einer vorbestimmten Menge von neuen Inhalts-Daten (beispielsweise jeweils nach 1 Megabyte) erstellen. Unglücklicherweise werden die Benachrichtigungs-Zeiten ebenfalls infolge der Variationen der Ankunftszeiten des Inhalts schwanken, die eine Funktion des Koders **101** und des Netzwerks zwischen dem Koder **101** und dem Videoserver **106** sind.

[0088] Gemäß einem Ausführungsbeispiel wird die Tag-Information verwendet, um das aktuelle Ende der Datei zu kennzeichnen. Insbesondere aktualisiert der Video-Server **106** effektiv die Ende-der-Datei-Information der Datei **134** mittels Sendens von Tag-Information vom Tag-Puffer **108** zur Speicherung mittels des MDS **112**. Sobald die Tag-Information, die einem bestimmten Abschnitt des Inhalts entspricht, vom Video-Server **106** übertragen worden ist, ist die Videopumpe **120** frei darin, eine Suche nach diesem bestimmten Abschnitt des Videos durchzuführen. Bis die Tag-Information, die einem bestimmten Abschnitt des Videos entspricht, freigegeben ist, darf die Videopumpe **120** keine Suche nach dem entsprechenden Abschnitt des Videos durchführen. Da die neueste Tag-Information das aktuelle Ende der Datei kennzeichnet, können neu angeschlossene Nut-

zer leicht den Inhalt suchen, der der neuesten Tag-Information zugeordnet ist, und das Abspielen der Zuführung mit einer Echtzeit-Rate starten.

Minimale Tag-Verzögerungszeit

[0089] Um den Client **122** daran zu hindern, stehenzubleiben oder ungültige Daten abzuspielen, wird die Übertragung von Tag-Daten vom Tag-Puffer **108** zum MDS **112** verzögert. Bevorzugt ist die Verzögerungsdauer lang genug, um sicherzustellen, dass auf die zugeordneten Inhalts-Daten nicht verfrüht zugegriffen wird. Andererseits erhöht die Verzögerung der Tag-Daten mehr als notwendig die Latenzzeit zwischen dem Zeitpunkt, zu dem der Inhalt kodiert ist, und dem Zeitpunkt, zu dem die Nutzer den Inhalt suchen oder abtasten können. Folglich ist es wünschenswert, eine minimale Tag-Verzögerungs-Zeitdauer zu ermitteln und die Puffer-Tag-Daten in dem Tag-Puffer **108** für eine minimale Tag-Verzögerungs-Zeitdauer zu Puffern. Die minimale Tag-Verzögerungs-Zeitdauer für ein Tag-Daten-Item wird mittels der maximalen Latenzzeit ermittelt, die mit dem Zuführen der entsprechenden Inhalts-Daten von dem Koder **101** der Videopumpe **120** verbunden ist.

[0090] Der Video-Server **106** weist einen Netzwerk-Puffer **152** und einen Schreib-Puffer **154** auf. Typischerweise wird der Videoserver **106** die Inhalts-Daten vom Kanal **128** in einen Netzwerk-Puffer **152** zur gleichen Zeit lesen, zu der der Video-Server **106** die Inhalts-Daten vom Schreib-Puffer **154** auf die Platten **114** schreibt. Bei Ausführungsbeispielen, bei denen Raid-Speichertechniken verwendet werden, werden Inhalts-Daten im Inneren des Video-Servers **106** in Einheiten empfangen und gepuffert, die einem Raid-Stripe entsprechen.

[0091] Die Videopumpe **120** weist eine Vorauslese-Einheit **146** und einen Puffer **144** auf. Die Videopumpe **120** liest die Inhalts-Daten asynchron von den Platten **114**. Um die Inhalts-Daten zu lesen, fordert die Vorauslese-Einheit **146** die Übertragung eines bestimmten Abschnitts von Inhalts-Daten an, und die Platten **114** antworten entweder mittels Sendens der angeforderten Inhalts-Daten oder mittels Anzeigens, dass die angeforderten Daten nicht gesendet werden können. Einige Latenzzeit tritt zwischen der Zeit auf, zu der die Vorauslese-Einheit **146** die Daten anfordert, und der Zeit, zu der die Daten von der Videopumpe **120** empfangen werden.

[0092] Wenn die Inhalts-Daten von der Datei **134** bei der Videopumpe **120** ankommen, speichert die Videopumpe **120** die Inhalts-Daten von der Datei **134** in den Puffer **144**. Sobald die Bandbreite auf dem Netzwerk **150** verfügbar wird, überträgt die Videopumpe **120** die Inhalts-Daten vom Puffer **144** über das Netzwerk **150** zu dem Client **122**. Wie bei dem Video-Server **106** werden Inhalts-Daten im Voraus gelesen und in der Videopumpe **120** in Einheiten gepuffert, die einem Raid-Stripe entsprechen, wenn Raid-Speichertechniken verwendet werden.

[0093] Wie oben erläutert, kopiert die Videopumpe **120** typischerweise Daten von einem Raid-Stripe in Netzwerk-Puffer und liest den folgenden Stripe im Voraus. Auf ähnliche Weise schreibt der Videoserver **106** einen Inhalt-Raid-Stripe in den Datenspeicher und empfängt Daten vom Netzwerk in einen zweiten Speicherpuffer. Folglich sind typischerweise vier Raid-Stripes „in der Übertragung“, so dass die Latenz zwischen dem Zeitpunkt, zu dem irgendwelche Inhalts-Daten erzeugt werden, und dem Zeitpunkt, zu dem sie verfügbar sind, um abgespielt zu werden, ungefähr die Zeit ist, die notwendig ist, um vier Raid-Stripes, gefüllt mit Daten, zuzuführen.

[0094] Raid-Stripes betragen gewöhnlicherweise 128 KBit oder 256 KBit pro Platte. Die kombinierte Gesamtgröße aller Platten in einem Raid-Stripe beträgt deshalb 1 bis 2 Megabit. Bei typischen MPEG-Dateien wird jeder Raid-Stripe ungefähr einer Sekunde des Videos entsprechen. Folglich führt dies mit vier Raid-Stripes auf dem Wege zu einer minimalen Latenzzeit von ungefähr 4 Sekunden.

[0095] Die Auswirkung auf Tag-Daten ist die, dass ein Tag von dem Video-Server **106** zur Verwendung durch andere Geräte freigegeben werden kann, wenn der entsprechende Inhalt verfügbar ist, abgespielt zu werden (d. h., der Inhalt für zwei Sekunden wurde erfolgreich auf der Platte gespeichert). Daher werden bei einem Video-Zuführsystem, bei dem die Inhalts-Zuführung vier Sekunden Latenzzeit benötigt, die Tag-Daten, die im Tag-Puffer **108** verblieben sind, nicht eher als vier Sekunden nach der Erzeugung des entsprechenden Inhalts übertragen.

[0096] Gemäß einem Ausführungsbeispiel werden die Schwankung und das Stehenbleiben beide durch das Übertragen eines Stapels von Tag-Daten vom Tag-Puffer **108** zum MDS **112** alle 12 Sekunden verhindert. Der Tag-Daten-Stapel, der in einem Intervall von jeweils 12 Sekunden übertragen wird, weist alle Tag-Informationen im Tag-Puffer **108** auf, die zumindest 12 Sekunden alt sind. Die Tag-Daten, die weniger als 12 Sekunden

alt sind, bleiben im Tag-Puffer **108** erhalten und werden zum MDS **112** in einem Stapel am Ende des nächsten 12-Sekunden-Intervalls übertragen. Der MDS-Server **110** sendet die Tag-Daten an verschiedene Geräte (beispielsweise die Videopumpe **120**), die die Videodatei **134** lesen, und speichert dann die Tag-Information auf den Platten **114**.

Digitale Kanäle

[0097] Videodateien, die für spezifische audio-visuelle Arbeiten, wie beispielsweise Sportereignisse, erzeugt worden sind, weisen eine endliche Länge auf. Folglich verbrauchen deren entsprechende Inhalts-Dateien ebenfalls eine endliche Menge an Speicher, was es praktikabel macht, die gesamte Inhalts-Datei für eine spätere Ansicht zu speichern. Jedoch ist ein herkömmlicher Fernseh-„Kanal“ aus einer nicht endenden Sequenz von audio-visuellen Arbeiten zusammengesetzt. Das fortwährende Verbleiben des gesamten Inhalts des digitalen Kanals würde den kontinuierlichen Speicherverbrauch auf einen unakzeptabel hohen Wert treiben. Andererseits ist es wünschenswert, Nutzern zu ermöglichen, sich Programme anzusehen, die sie noch nicht befähigt waren, sich zu der Zeit anzuschauen, zu der die Programme ursprünglich gesendet worden sind. Beispielsweise wäre es für einen Zuschauer wünschenswert, einen Zugriff auf die letzten 24 Stunden des Programms zu haben, das über einen digitalen Kanal ausgesendet worden ist. Gemäß einem für das Verständnis der Erfindung wichtigen Beispiel wird das herkömmliche Anschauen für eine Endlos-Zuführung durch die Verwendung eines kontinuierlichen endlichen Puffers bereitgestellt, wobei ältere Daten „ablaufen“ und mit neuen Daten überschrieben werden.

Inhalts-Ablauf

[0098] Um einen kontinuierlichen Daten-Puffer zu haben, beispielsweise die letzten 24 Stunden der Lebenszeit, Fernsehen für Frauen, muss älterer Inhalt mit den entsprechenden Tags gelöscht werden. Es können verschiedene Ansätze verwendet werden, um solch einen kontinuierlichen Puffer zu implementieren.

[0099] In Bezug auf die Inhalts-Daten ist der einfachste Ansatz, um einen kontinuierlichen Puffer zu implementieren, eine einzelne Datei zu erzeugen, die groß genug ist, Filmmaterial mit einer Länge von 24 Stunden zu halten. Die Datei wird dann als Ring-Puffer behandelt. Insbesondere würde der MDS-Server **110** nach der Erzeugung der initialen 24 Stunden-Datei den Beginn der Datei als den aktuellen „Einstiegspunkt“ setzen. Der MDS-Server **110** würde dann die neuen Inhalts-Daten über die alten Daten am Einstiegspunkt speichern und den Einstiegspunkt an das Ende der neuen Daten bewegen. Wenn der Einstiegspunkt auf das Ende der Datei trifft, wird er umlaufend erneut auf den Beginn der Datei gesetzt.

[0100] Unglücklicherweise macht es dieser Einzel-Datei-Ringpuffer-Ansatz schwierig, die Zeittdauer der Datei zu vergrößern oder zu verkleinern. Beispielsweise wird angenommen, dass sich der Einstiegspunkt in der Mitte der Datei befindet, und eine Entscheidung wird getätigt, die Datei zu vergrößern, so dass sie 48 Stunden fasst. Unter diesen Umständen könnte der MDS-Server **110** nicht beginnen, die Zeit zu erhöhen, für die weitere 12 Stunden abgedeckt sind, wenn der Einstiegspunkt das Ende der Datei erreicht haben würde. Unter Verwenden des Einzel-Ringpuffer-Ansatzes ist es ebenso schwierig zu erfassen, wenn ein Client pausiert hat und sich der „Horizont“ über ihre Position bewegt hat, so dass, wenn sie fortsetzen, der Inhalt, den sie sahen, überschrieben ist.

[0101] [Fig. 4](#) stellt einen alternativen, flexibleren Ansatz zum Puffern einer vorbestimmten Menge einer Endlos-Videozufuhr dar. Bezugnehmend auf [Fig. 4](#) werden die Inhalts-Daten in einer Gruppe kleinerer Dateien **402-414** gespeichert. Jede der kleineren Dateien speichert einen Teil (Sub)-Satz der gepufferten Inhalts-Daten. Bei dem dargestellten Ausführungsbeispiel speichert jede der Dateien **402-412** zwei Stunden Nutzinhalt. Die Datei **414** speichert zurzeit eine Stunde Inhalt. Der aktuelle Einstiegspunkt befindet sich am Ende der Datei **414**. Erreicht die Datei **414** zwei Stunden Inhalt, wird die Datei **414** geschlossen, und eine neue Inhalts-Datei wird erzeugt. Da Inhalts-Dateien altern, werden die älteren Inhalts-Dateien gelöscht, um Plattspeicher für neue Dateien freizumachen. Während der Wiedergabe werden die Dateien von der Videopumpe nahtlos zusammengefügt, wie die Inhalts-Daten an den Client gesendet werden.

[0102] Wenn die Pufferungstechnik, die in [Fig. 4](#) dargestellt ist, verwendet wird, kann eine nachsichtige Ablauf-Strategie eingestellt werden. Insbesondere kann eine Strategie aufgestellt werden, dass eine Datei nicht gelöscht wird, bis alle Clients die Datei abgeschlossen haben (die Datei und irgendwelche Dateien, die der Datei vorausgehen). Beispielsweise wird angenommen, dass es Nutzern ermöglicht wird, auf die letzten 12 Stunden einer Zuführung zuzugreifen. Ist die Datei **414** abgeschlossen, enthalten die Dateien **404-414** die letzten 12 Stunden, so dass die Datei **402** nicht länger erforderlich ist. Jedoch kann sich ein Client die Inhalte der Datei

402 zur Zeit anschauen. Folglich wird die Datei **402** nicht unmittelbar gelöscht. Stattdessen werden neue Clients daran gehindert, auf die Datei **402** zuzugreifen, aber dem Client, der zurzeit auf die Datei **402** zugreift, wird ermöglicht, das Abspielen der Datei **402** zu beenden. Wenn der letzte Client das Abspielen der Datei **402** beendet hat, wird die Datei **402** gelöscht.

[0103] Um einen Abschluss auf die Anzahl bestehender Dateien zu setzen, kann für Clients eine zeitliche Grenze aufgestellt werden, um das Abspielen alter Dateien zu beenden. Beispielsweise, wenn die Datei **414** abgeschlossen ist, werden nicht nur neue Clients daran gehindert, auf die Datei **402** zuzugreifen, sondern den Clients, die zurzeit auf die Datei **402** zugreifen, wird zwei Stunden gegeben, um das Abspielen der Datei **402** zu beenden. Am Ende der zwei Stunden wird dann die Datei **402** gelöscht, um Plattspeicher freizumachen, unabhängig davon, ob irgendeiner der Clients immer noch die Datei **402** liest.

Tag-Ablauf

[0104] Wird eine Inhalts-Datei (beispielsweise Datei **402**) gelöscht, werden die Tags, die der gelöschten Datei entsprechen, als „abgelaufen“ betrachtet, und können deshalb gelöscht werden. Idealerweise sind Tags in einem Format gespeichert, wie beispielsweise einer Datenbank-Tabelle, was es ermöglicht, alte Tags leicht zu löschen ebenso wie neue hinzuzufügen. Unglücklicherweise kann der Overhead, der mit dem Speichern und Abfragen von Tags von einer Datenbank-Tabelle verbunden ist, zu teuer sein, um praktikabel unter Live-Zuführ-Bedingungen zu sein. Für einen einfachen und schnellen Zugriff werden Tags deshalb typischerweise in einer flachen Datei (Flat File) gespeichert.

[0105] Bezugnehmend auf [Fig. 5](#) ist dort eine flache Tag-Datei **500** dargestellt. Die flache Tag-Datei **500** weist einen Header **502** auf, gefolgt von einem Satz von Tags **504**. Der Header **502** kennzeichnet Informationen über den Inhalt der Tag-Datei **500** inklusive des Satzes von Inhalts-Dateien, denen die Tags innerhalb der Tag-Datei **500** entsprechen.

[0106] Kommen neue Tags hinzu, werden die Tags an die Tag-Datei **500** angehängt. Da die Tag-Datei **500** einer kontinuierlichen Zuführung zugeordnet ist, wird die Tag-Datei undefinierbar groß, wenn kein Mechanismus zum Löschen abgelaufener Tags vorgesehen ist. Jedoch sollte die Tag-Datei **500** selbst gültig bleiben, selbst nach dem Ablauf einiger Tags (beispielsweise der Tags **510**) innerhalb der Tag-Datei **500**, da Clients auf die Tags **512** innerhalb der Tag-Datei **500** zugreifen können und diese nutzen können, die noch nicht abgelaufen sind. Daher kann der Ablauf-Mechanismus nicht einfach die abgelaufenen Tags **510** aus der Tag-Datei **500** löschen.

[0107] Stattdessen wird eine temporäre Tag-Datei **514** mittels Erstellens eines neuen Headers **506** und Anhangens des neuen Headers **506** als Kopie der nicht abgelaufenen Tags **512** von der alten Tag-Datei **500** aufgebaut, als dass die abgelaufenen Tags innerhalb der Tag-Datei **500** direkt aus ihr gelöscht werden. Der neue Header **506** weist die gleiche Information auf wie der alte Header **502**, abgesehen davon, dass die Daten innerhalb des Headers **502** kennzeichnen, dass die Tag-Datei **500** Tags für die gelöschte Inhalts-Datei aufweist, während die Daten innerhalb des Headers **506** dies nicht tun.

[0108] Während die neue Tag-Datei **514** erzeugt wird, werden neue Tag-Daten sowohl an die neue Tag-Datei **514** als auch an die alte Tag-Datei **500** angehängt. Nachdem die neue Tag-Datei **514** erzeugt worden ist, werden neue Tag-Daten an die neue Tag-Datei **514** angehängt, statt dass sie an die alte Tag-Datei **500** angehängt werden. Um sicherzustellen, dass die neuen Tag-Daten nach den Tag-Daten **512** angehängt werden, ist im Voraus ein Speicherbereich für die zu kopierenden Tags **512** in der neuen Tag-Datei **514** vorgesehen, und die neuen Tags werden nach dem vorgegebenen Speicherbereich angehängt, während die bestehenden Tags **512** in den vorgegebenen Speicherbereich kopiert werden.

[0109] Wenn alle der nicht abgelaufenen Tags **512** in die neue Tag-Datei **514** kopiert worden sind, wird die alte Tag-Datei **500** geschlossen, und die neue Tag-Datei **514** wird auf den Namen der alten Tag-Datei **500** umbenannt. Nachdem die neue Tag-Datei **514** umbenannt worden ist, werden die Tag-Datei-Lesevorrichtungen (beispielsweise der Stream-Server **118**), die die alte Tag-Datei **500** verwendet haben, basierend auf der Information zurückgesetzt, die im Header der neuen Tag-Datei **514** enthalten ist. Gemäß einem Ausführungsbeispiel (dem „Push-Modell“) werden Nachrichten an die Tag-Datei-Lesevorrichtungen gesendet, um sie ausdrücklich zu informieren, dass die Tag-Datei geändert worden ist, und dass sie sich selbst basierend auf der Header-Information in der neuen Tag-Datei **514** aktualisieren sollen.

[0110] Gemäß einem alternativen „Pull-Modell“-Ausführungsbeispiel werden die Tag-Datei-Lesevorrichtun-

gen nicht ausdrücklich informiert. Stattdessen sind diese so konfiguriert, dass sie basierend auf der Header-Information der neuen Tag-Datei selbst lesen und sich aktualisieren, wann immer sie bei einem Versuch, ein Tag zu lesen, fehlschlagen. Der Pull-Modell-Ansatz hat den Vorteil, dass er die Übertragung von Nachrichten verhindert, die unter vielen Umständen nicht notwendig sind.

[0111] Wenn Tags, die einem bestimmten Inhaltsegment zugeordnet sind, gelöscht werden, können Clients fortsetzen, sich das Inhalts-Segment anzuschauen. Jedoch sind die Clients nicht befähigt, einen nicht-sequentiellen Zugriff auf Operationen durchzuführen, die die gelöschte Tag-Information erfordern, wie beispielsweise schnelles Vorspulen und Zurückspulen.

Zuweisung von Datum und Uhrzeit

[0112] Die Tag-Information weist eine Zeitstempel-Information für jedes der Rahmen in den entsprechenden Inhalts-Daten auf. Zum Zwecke des Dekodierens stellt die Zeitstempel-Information typischerweise die Zeit in Bezug zum Beginn einer Zuführung (d. h. die „Darstellungszeit“) dar und wird auf den Byte-Offset in der Inhalts-Datei des Rahmens abgebildet, der dieser Darstellungszeit entspricht. Jedoch können für solche kontinuierlichen Zuführungen solche relativen Zeitwerte nicht bedeutsam sein. Beispielsweise würde ein Nutzer wollen, eine Wiedergabe, beginnend am 21. Januar 1997, 16:30:23 Uhr, anzufordern, statt dass er 5 345 789,76 Sekunden nach der Zeit startet, zu der eine Station begonnen hat zu senden.

[0113] Gemäß einem Ausführungsbeispiel der Erfindung werden absolute Zeitwerte mittels Speicherns eines absoluten Zeitwertes unterstützt, der dem relativen Zeitwert „0“ entspricht. Deshalb wird der absolute Zeitwert, der „0“ zugeordnet ist, wenn ein Client die Wiedergabe von einer absoluten Zeit bestimmt, vom bestimmten absoluten Zeitwert subtrahiert, um einen relativen Zeitwert zu erhalten. Der relative Zeitwert wird dann mittels des Stream-Servers **118** verwendet, um die geeignete Tag-Information zu identifizieren, und die Tag-Information wird vom Stream-Server **118** verwendet, eine Videopumpe **120** dazu zu bringen, das Senden von Inhalt von der geeigneten Position in der Inhalts-Datei **134** an zu senden.

[0114] Typischerweise stellen die Transportformate digitaler Videos eine feste Anzahl von Bits (beispielsweise 33 Bit) bereit, um Zeitstempel darzustellen. Für kontinuierliche Zuführ-Umgebungen werden die relativen Zeitstempelwerte unvermeidlich Zahlen erreichen, die nicht mit der Bit-Anzahl darstellbar sind, die im Transportformat verfügbar ist. Wenn dies geschieht, „laufen“ die Zeitstempelwerte „um“ und beginnen wieder bei 0.

[0115] Um dem Umlauf-Problem zu begegnen, wird ein Umlaufwert mit höherer Genauigkeit (beispielsweise 64 Bit) verwaltet. Beim Durchführen einer Suche oder eines anderen nichtsequentiellen Zugriffs verwendet der Stream-Server **118** die Zeitstempelwerte mit höherer Genauigkeit. Beim Übertragen von Inhalt an einen Client sendet die Videopumpe **120** die Zeitstempel mit geringerer Genauigkeit.

[0116] Die Video-Kodier- und Zuführ-Techniken, wie an dieser Stelle beschrieben, ermächtigen Nutzer mit Funktionssteuerungen, die sich vorher ausschließlich in der Domäne von Programm-Anbietern befanden. Beispielsweise bestimmen zurzeit Programm-Anbieter, welche Spiele der SuperBowl Zuschauern wiedergegeben werden, die Geschwindigkeit, mit der sie wiedergegeben werden, und wie oft sie wiedergegeben werden.

[0117] Jedoch haben Zuschauer meist stark unterschiedliche Meinungen darüber, wie die Leistungen des mehrfachen Zuschauens zu bewerten sind. Beispielsweise können zwei Zuschauer die Genauigkeit eines bestimmten Anrufs diskutieren. Jedoch kann der Programm-Anbieter das Spiel nicht beachten, das bewirkte, das der Anruf bedeutsam genug war, das Spiel zu wiederholen. Unter Verwenden der an dieser Stelle bereitgestellten Techniken können Zuschauer selbst entscheiden, welche Spiele unmittelbar wiederholt werden sollen, mit welcher Geschwindigkeit sie wiederholt werden sollen, und wie oft sie wiederholt werden sollen.

[0118] In der vorangegangenen Beschreibung wurde die Erfindung unter Bezugnahme auf deren spezifische Ausführungsbeispiele beschrieben. Es ist jedoch offensichtlich, dass verschiedene Modifikationen und Änderungen daran vorgenommen werden können. Die Beschreibung und die Zeichnungen sind demgemäß eher in einem erläuternden als in einem einschränkenden Sinne zu beachten.

Patentansprüche

1. Was beansprucht wird, ist:
1. Verfahren zum Bereitstellen eines nicht-sequentiellen Zugriffs auf Videos von einer kontinuierlichen Zufuhr, wobei das Verfahren aufweist das Empfangen eines digitalen Datenstroms, der mittels Kodierens der kontinu-

ierlichen Zufuhr in ein digitales Videoformat erzeugt worden ist, bei einem Server (106), wobei das Verfahren aufweist:

Erzeugen von Tag-Information bei dem Server (106), die Information über Rahmen kennzeichnet, die in dem digitalen Datenstrom enthalten sind, wobei die Tag-Information Zeitstempel aufweist, die die Zeitsteuerung von Rahmen relativ zu einem Anfang des digitalen Datenstroms kennzeichnen; und wobei die Tag-Information ferner Information über eine oder mehrere Zustandsmaschinen aufweist, die verwendet werden, um den digitalen Datenstrom zu dekodieren; wobei das Verfahren gekennzeichnet ist durch:

Speichern eines Anfangszeitwerts bei dem Server (106), der eine Absolutzeit anzeigt, die mit dem Anfang des digitalen Datenstroms korrespondiert;

Empfangen einer Anforderung von einem Client (122) für die Wiedergabe, die bei einer vorgegebenen Absolutzeit beginnt;

Subtrahieren des Anfangszeitwerts von der vorgegebenen Absolutzeit, um eine Relativzeit festzulegen; Verwenden der Tag-Information, um eine Stelle in dem digitalen Datenstrom zu identifizieren, die zu der Relativzeit korrespondiert; und

Übertragen des digitalen Datenstroms an den Client (122), wobei an der Stelle in dem digitalen Datenstrom begonnen wird, die zu der Relativzeit korrespondiert.

2. Verfahren nach Anspruch 1, wobei:

der Server (106) einen ersten Präzisionszeitstempel verwendet, wenn eine Suche oder ein anderer nichtsequentieller Zugriff durchgeführt wird; und der Server (106) einen zweiten Präzisionszeitstempel verwendet, wenn Inhalt an einen Client (122) übertragen wird, wobei die erste Genauigkeit höher als die zweite Genauigkeit ist.

3. Verfahren nach Anspruch 1, wobei die Information über eine oder mehrere Zustandsmaschinen Information über eine Programm-Elementarstrom-Zustandsmaschine oder eine Pack-Zustandsmaschine aufweist.

4. Verfahren nach Anspruch 1, wobei die Information über eine oder mehrere Zustandsmaschinen Information über eine Video-Zustandsmaschine aufweist.

5. Verfahren nach Anspruch 1, wobei die Information über eine oder mehrere Zustandsmaschinen Information über eine Transportschicht-Zustandsmaschine oder eine System-Zustandsmaschine aufweist.

6. Verfahren nach Anspruch 1, wobei das Übertragen des digitalen Datenstroms verzögert wird zum Verhindern, dass der Client (122) stehen bleibt oder ungültige Daten abspielt.

7. Verfahren nach Anspruch 6, ferner aufweisend, bevor der digitale Datenstrom übertragen wird, Festlegen einer minimalen Tag-Zeitverzögerungszeitdauer, die zum Verzögern des Übertragens des digitalen Datenstroms verwendet wird.

8. Verfahren nach Anspruch 7, wobei die minimale Tag-Verzögerungszeitdauer mittels einer maximalen Latenzzeit festgelegt wird, die mit dem Zuführen des digitalen Datenstroms von einem entsprechenden Koder (101) an eine entsprechende Videopumpe (120) verbunden ist.

9. Verfahren nach Anspruch 1, wobei die Tag-Information ferner einen Dateityp-Identifizierer (202), einen Längen-Kennzeichner (204), einen Bit-Raten-Kennzeichner (206), einen Spieldauer-Kennzeichner (208), einen Rahmennummer-Kennzeichner (210), Stromzugriffsinformation (212), eine Anfangszeit-Offset (213) und eine Vielzahl von Tag-Datei-Einträgen (214) aufweist.

10. Verfahren nach Anspruch 9, wobei jeder Tag-Datei-Eintrag (214) der Vielzahl von Tag-Datei-Einträgen (214) die Information über eine oder mehrere Zustandsmaschinen aufweist.

11. Verfahren nach Anspruch 10, wobei die Information über eine oder mehrere Zustandsmaschinen Information über eine Programm-Elementarstrom-Zustandsmaschine, Information über eine Video-Zustandsmaschine und Information über eine Transportschicht-Zustandsmaschine aufweist.

12. Verfahren nach Anspruch 11, wobei die Information über eine Programm-Elementarstrom-Zustandsmaschine ferner einen Programm-Elementarstrom-Offset bei einem Bildstart (217) und einen Programm-Elementarstrom-Offset bei einem Bildende (219) aufweist.

13. Verfahren nach Anspruch 11, wobei die Information über eine Video-Zustandsmaschine ferner eine Bildgröße (220), eine Bildstartposition (226), einen Zeitwert (228), einen Rahmen-Typ (232) und eine Zeitsteuerungs-Puffer-Information (238) aufweist.

14. Verfahren nach Anspruch 11, wobei die Information über eine Transportschicht-Zustandsmaschine ferner einen Start-Offset (234), eine Anzahl von Nicht-Video-Paketen (222), eine Anzahl von Auffüllpaketen (224), einen End-Offset (236), einen Strom-Stetigkeits-Zähler (215) und einen Unterbrechungs-Bitschalter (230) aufweist.

15. Verfahren nach Anspruch 10, wobei die Information über eine oder mehrere Zustandsmaschinen Information über eine Paketierungs-Zustandsmaschine, Information über eine Video-Zustandsmaschine und Information über eine System-Zustandsmaschine aufweist.

16. Verfahren nach Anspruch 15, wobei die Information über eine Paketierungs-Zustandsmaschine ferner einen Pack-Offset bei einem Bildanfang (225), einen Pack-Rest bei dem Bildanfang (227), einen Pack-Offset bei einem Bildende (229) und einen Pack-Rest bei dem Bildende (231) aufweist.

17. Verfahren nach Anspruch 15, wobei die Information über eine Video-Zustandsmaschine ferner eine Bildgröße (233), eine Bildanfangsposition (235), eine Bildendposition (237), einen Zeitwert (241), einen Rahmen-Typ (239) und eine Zeitsteuerungs-Puffer-Information (243) aufweist.

18. Verfahren nach Anspruch 15, wobei die Information über eine System-Zustandsmaschine ferner eine Menge von Nicht-Video-Daten (221) und eine Menge von Auffüll-Daten (223) aufweist.

Es folgen 8 Blatt Zeichnungen

Anhängende Zeichnungen

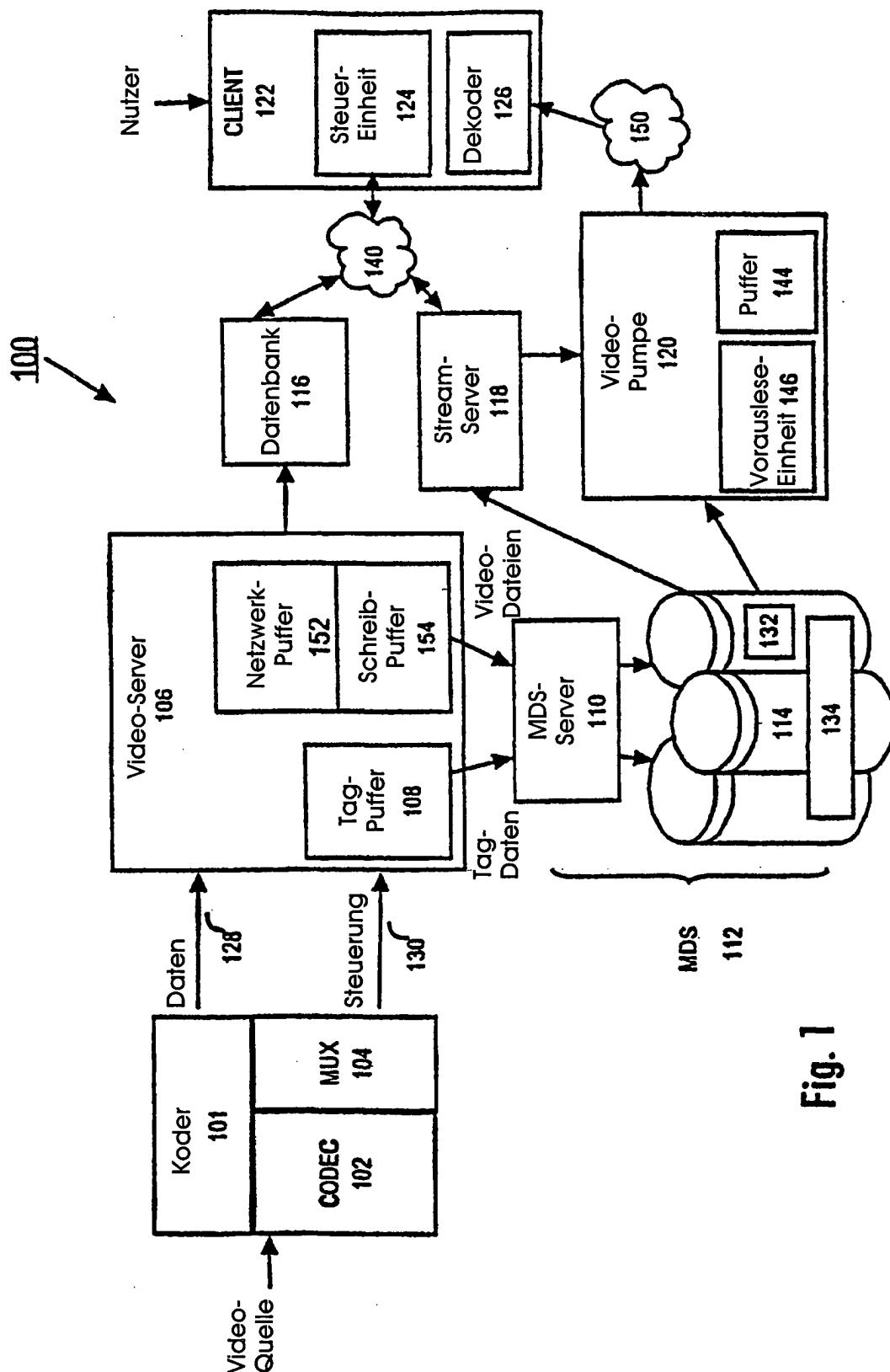


Fig. 1

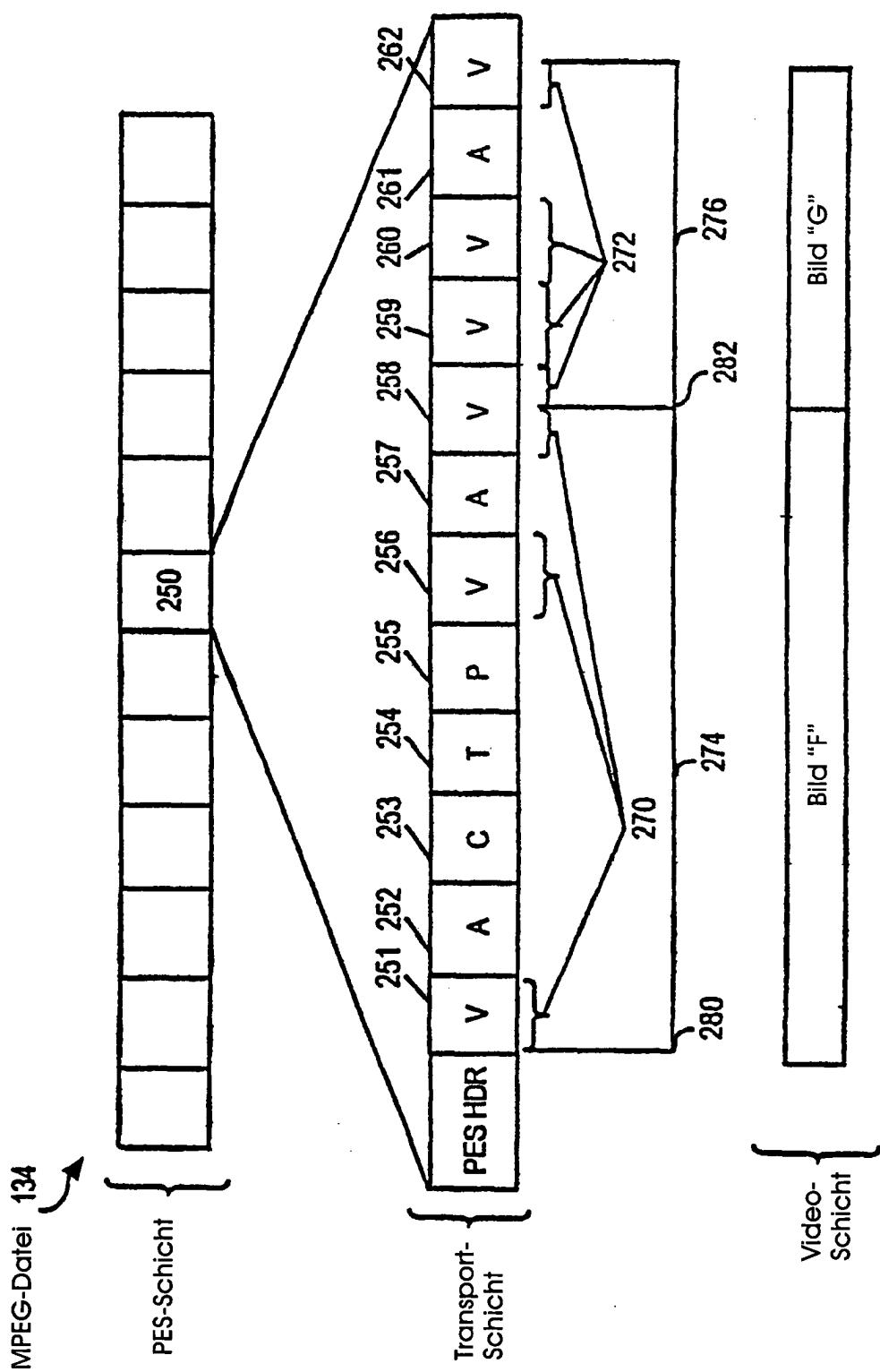


Fig. 2A

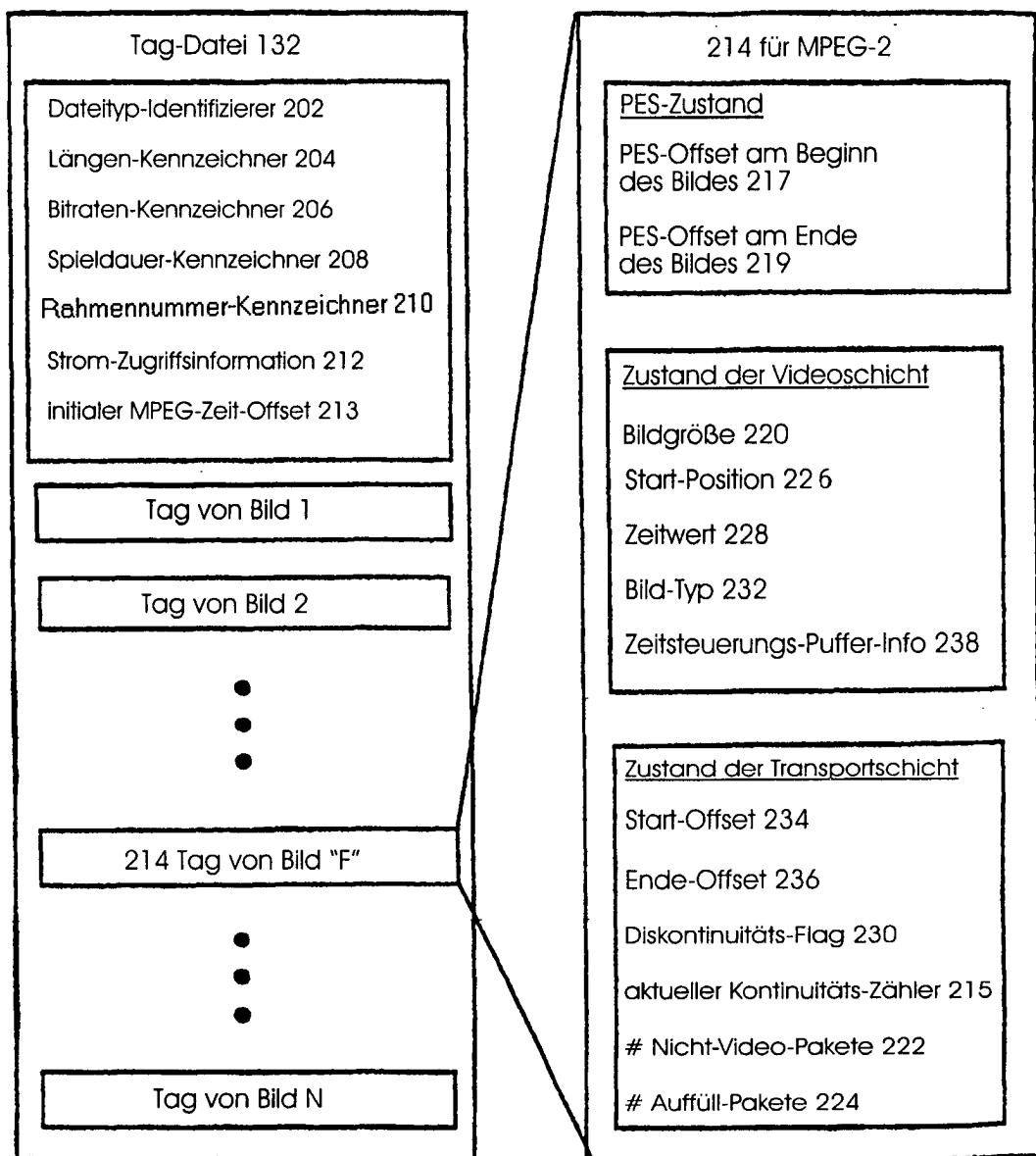


Fig. 2B

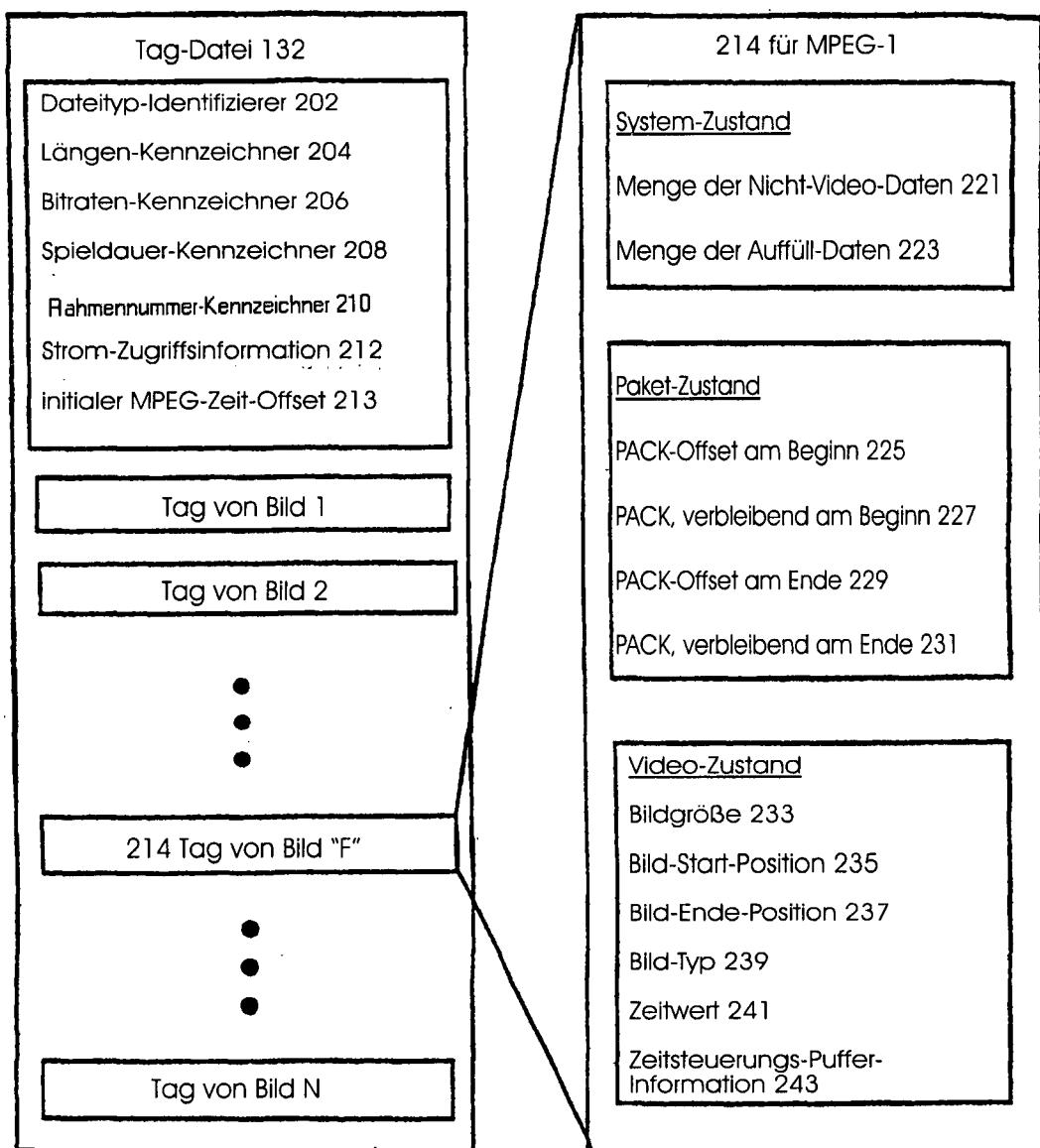


Fig. 2C

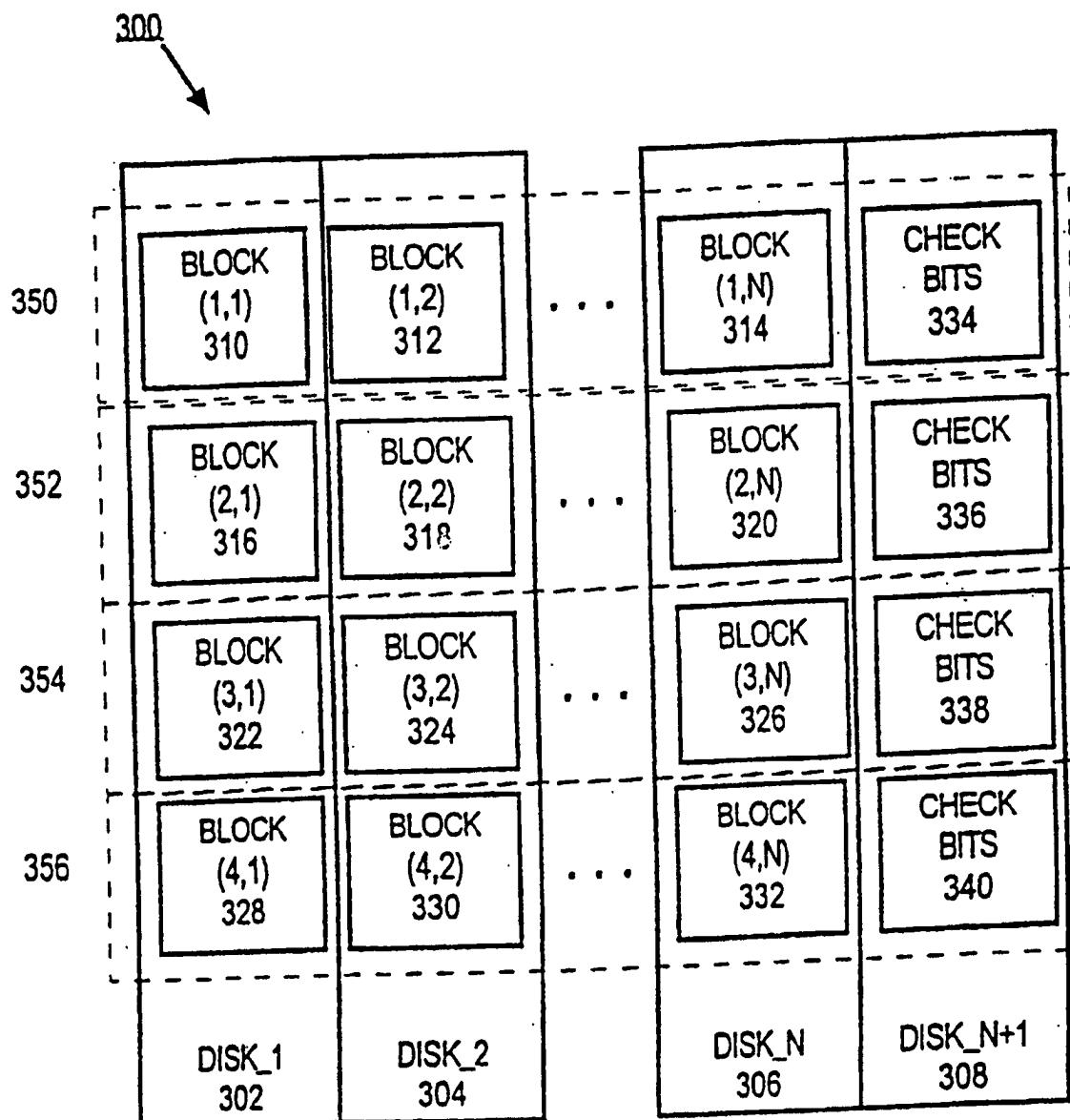


Fig. 3A

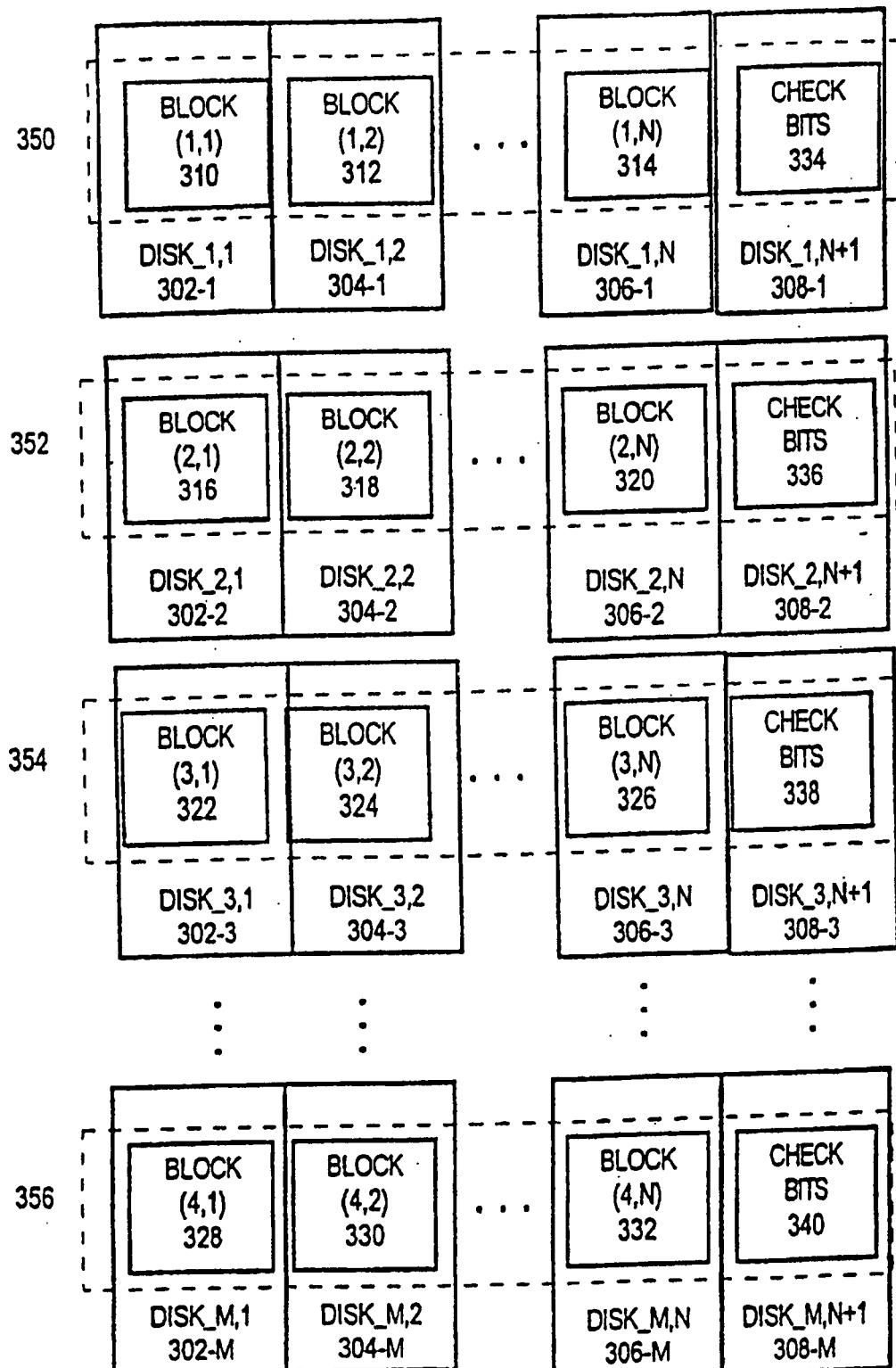


Fig. 3B

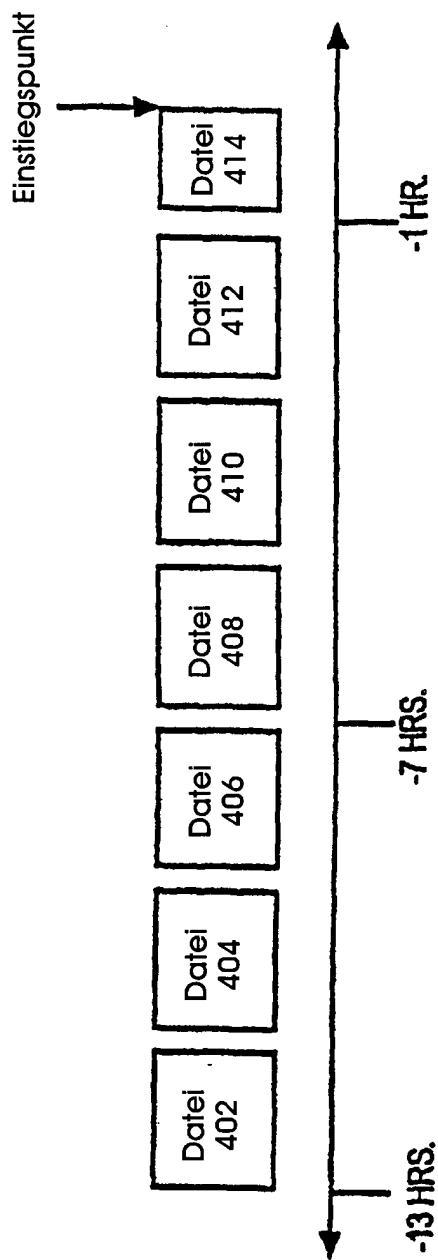


Fig. 4

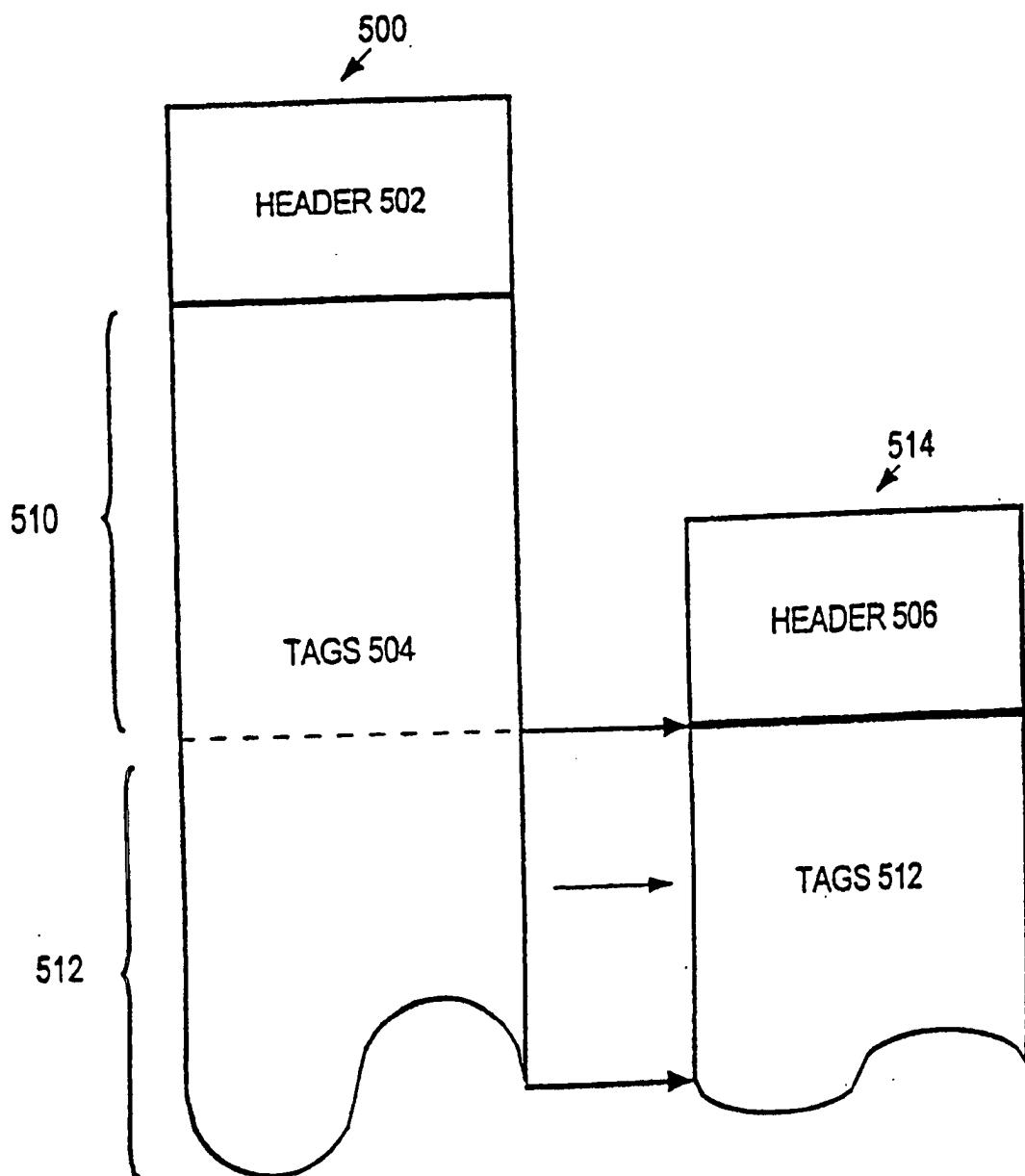


Fig. 5