

Seminar User-Interface Design und Multimedia

Datenformate für Multimedia-Files

Steffen Kamp

Achterstraße 33, 52062 Aachen
E-mail: steffen.kamp@post.rwth-aachen.de

Zusammenfassung: Die Vielzahl an verfügbaren Datenformaten zur Kodierung von multimedialen Inhalten erfordert eine Einteilung in verschiedene Klassen. Einige Datenformate aus den verschiedenen Bereichen zur Kodierung von Standbildern sowie Audio- und Videodaten werden kurz vorgestellt.

Schlüsselwörter: Multimedia, Datenformate, Dateiformate, Video, Audio, Datenkompression, Streaming

1 Klassifizierung von Multimedia Datenformaten

Die Vielfalt an Datenformaten und Kompressionsverfahren macht es schwierig, sich einen umfassenden Überblick über dieses Themengebiet zu verschaffen. Um nicht alle Formate einzeln betrachten zu müssen, ist es sinnvoll, sie in verschiedene anwendungsspezifische Klassen einzuteilen.

Zunächst kann man Multimediadatenformate in Formate zur Beschreibung von unbewegten und bewegten Medien unterteilen. Unbewegte Formate beschreiben ausschließlich Standbilder, bewegte Formate können Videodaten, Audiodaten oder auch beides enthalten.

Für Standbilder kommen die pixelbasierte Methode, bei der ein Bild in horizontaler und vertikaler Richtung in einzelne Bildpunkte (*Picture Element*, Pixel) aufgeteilt wird, und die vektorbasierte Methode, die Bilder anhand geometrischer Strukturen darstellt, in Frage.

Daten mit bewegtem Inhalt lassen sich wiederum in zwei Untergruppen aufspalten: die der linearen und die der nichtlinearen Daten. Beispiele für lineare Daten wären ein Film oder ein Musikstück mit strenger zeitlicher Abfolge der Bilder oder Töne. Der Anwender hat dabei lediglich die Möglichkeit, in der Zeit vor- oder zurückzuspringen, einen sinnvollen Gesamteindruck erhält man jedoch nur bei kontinuierlicher Wiedergabe der gespeicherten Daten. Im Gegensatz dazu können nichtlineare Daten mehrere alternative oder unabhängige Handlungsstränge und Ereignisse besitzen, die vom Anwender abgerufen werden. Teilweise hat der Anwender bei nichtlinearen Daten sogar Einflußmöglichkeiten auf den Inhalt.

Für jede dieser Klassen gibt es eine ganze Reihe von Datenformaten. Deren Unterschiede begründen sich zumeist in unterschiedlichen Einsatzgebieten und Anforderungen. So benötigt man für die Erstellung von „traditionellen“ Medien, wie zum Beispiel Filmen oder Zeitschriften, komplexe und häufig verlustlose Datenformate. Für internetbasierte Anwendungen beispielsweise steht hohe Kompression im Vordergrund, um eine akzeptable Transferdauer über meist relativ langsame Telefonverbindungen zu gewährleisten (siehe Tab. 1).

Profi/Autor	Endanwender (über Internet)
möglichst verlustlos	hohe Komprimierung
unsicht-/unhörbare Zusatzinformationen	eingeschränkte Zusatzinformationen
TIFF	JPEG/JFIF
AIFF	MP3
DV	MPEG

Tabella 1: Nach Einsatzgebiet unterschiedliche Eigenschaften von Dateiformaten und einige Beispiele.

2 Einige Datenformate im Detail

2.1 Formate für unbewegte Bilder

2.1.1 Das Tagged Image File Format (TIFF)

Das *Tagged Image File Format* (TIFF) wurde erstmals 1986 von der *Aldus Corporation* (seit der Fusion 1994 *Adobe Systems Incorporated*) vorgestellt. Das TIF-Format war das Ergebnis mehrerer Treffen mit einer Reihe von Software- und Scannerherstellern und sollte eine gemeinsame Basis für einen reibungslosen Datenaustausch zwischen Scannern und Software, sowie zwischen verschiedenen Anwendungen bilden. Die TIFF-Spezifikation liegt derzeit in Revision 6.0 – der vierten öffentlichen Version – vom 3. Juni 1992 vor [1].

TIFF ist ein außerordentlich flexibles Datenformat zur Beschreibung von Pixelbildern, sogenannten *Raster Image*-Daten. Eine Datei kann theoretisch eine unbegrenzte Anzahl von Elementen, sowohl Bilder als auch Zusatzinformationen, enthalten. Ermöglicht wird dies durch ein oder mehrere Inhaltsverzeichnisse, die auf die einzelnen Datenblöcke verweisen. Der Inhalt eines spezifischen Blocks wird durch die ersten zwei Byte des jeweiligen Blocks bestimmt – dem sogenannten *Tag*, dem das TIF-Format seinen Namen verdankt.

Diese Struktur ermöglicht es einzelnen Implementoren, eigene Daten mit zu sichern, indem sie einen bislang nicht spezifizierten Tag verwenden. Idealerweise sollte dieser Tag bei Adobe registriert werden, um Überschneidungen mit anderen Herstel-

lern zu vermeiden. Weiterhin ist das Format dadurch einfach für zukünftige Anforderungen erweiterbar, bei gleichzeitig guter Rückwärtskompatibilität zu älteren Implementationen.

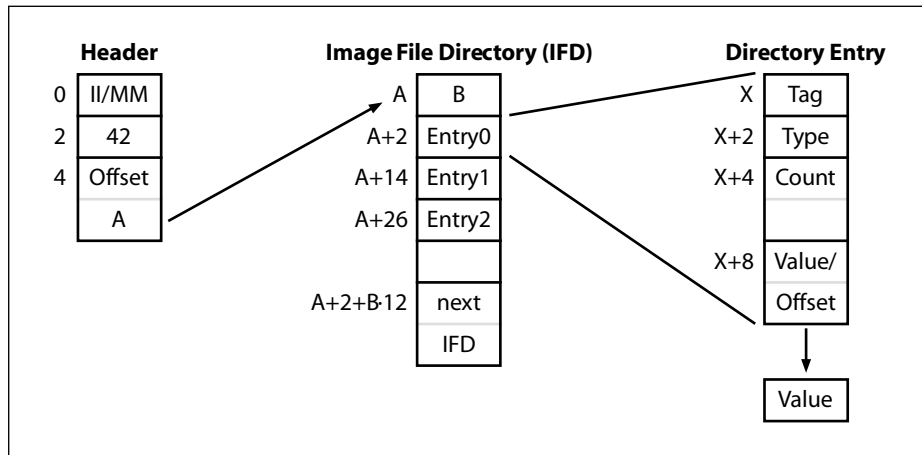


Abbildung 1: Struktur von TIFF-Dateien.

2.1.2 Kompression mittels JPEG (Joint Photographic Experts Group)

Die *Working Group 8 (WG8)* der *International Standards Organisation (ISO)* arbeitet seit 1982 an Standards zur Kompression und Dekompression von Standbildern. Nach dem Vergleich von zehn Kodierungsverfahren im Jahre 1987 und einer vertieften Untersuchung dreier dieser Verfahren wurde das am besten geeignete Verfahren ausgewählt. Dieses wurde unter Berücksichtigung der anderen Verfahren weiterentwickelt und mündete schließlich in der Verabschiedung des JPEG-Kodierverfahrens, das seit 1992 ein *ISO International Standard (IS)* ist [2].

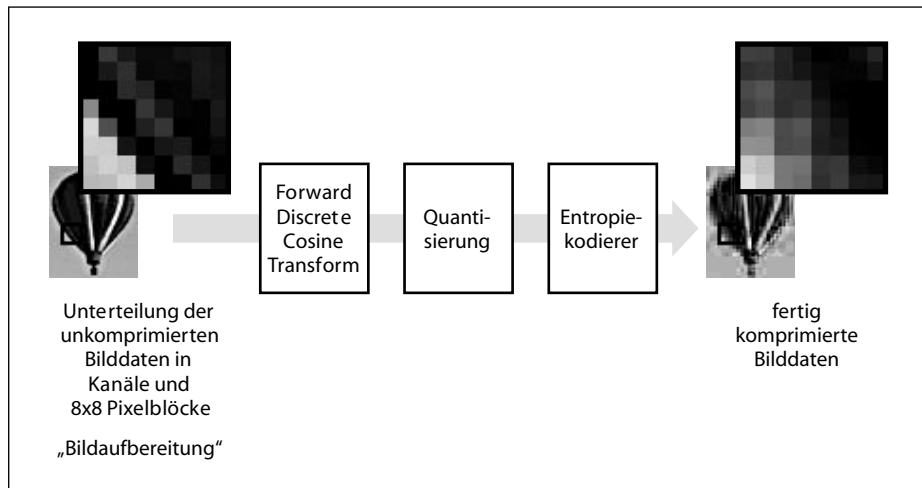


Abbildung 2: Die Schritte der JPEG-Komprimierung.

Um eine weite Verbreitung von JPEG zu gewährleisten, wurden mehrere Anforderungen an das Verfahren gestellt, von denen einige wichtige hier aufgezählt sind [3]:

- JPEG soll nahezu beliebige Bildgrößen mit jedem *Höhe-zu-Breite*-Verhältnis beschreiben können.
- Die Komplexität des Verfahrens soll eine Software-Lösung auf verfügbaren Standardprozessoren ermöglichen und mit entsprechender Spezialhardware drastisch reduzierbar sein.
- Die Dekodierung des Bildes soll in *sequentieller* (Zeile für Zeile) und *progressiver* (das Bild wird als Ganzes immer weiter verfeinert) Folge möglich sein.
- Es soll sowohl verlustbehaftete als auch verlustfreie Komprimierung der Bilder möglich sein.

Das verlustbehaftete JPEG-Kompressionsverfahren teilt sich in vier grundlegende Abschnitte in ebendieser Reihenfolge auf: Bildaufbereitung, Bildverarbeitung, Quantisierung und Entropiekodierung (siehe Tab. 2). Zur Dekompression müssen diese Schritte in umgekehrter Reihenfolge ausgeführt werden.

Bildaufbereitung

Jede der maximal 255 Bildebenen (meist die Farbkomponenten) eines Bildes wird bei der Bildaufbereitung in Blöcke zu je 8×8 Pixeln aufgeteilt, deren Bearbeitung in den folgenden Schritten entweder streng linear (*non-interleaved data ordering*) oder in verschiedenen – auch ebenenübergreifenden – verzahnten Reihenfolgen (*Interleaving*) stattfinden kann.

Bildverarbeitung

Bei der Bildverarbeitung wird jeweils ein 8×8 -Block mit Hilfe der *Diskreten Cosinus Transformation* (DCT) in seine horizontalen und vertikalen Frequenzanteile zerlegt.

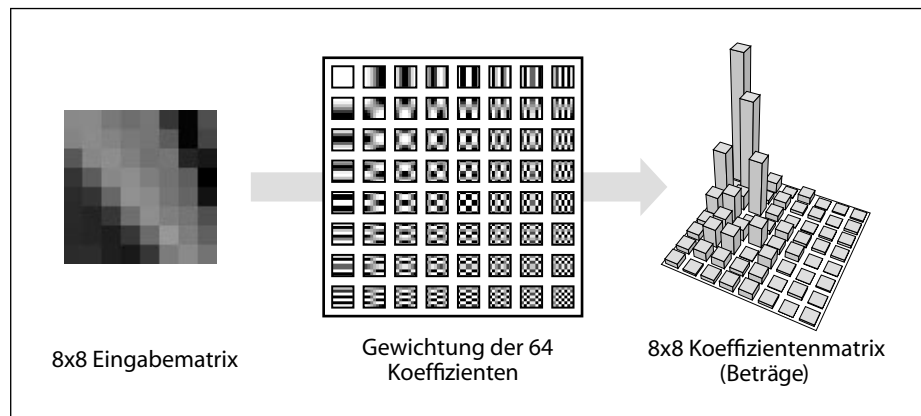


Abbildung 3: Cosinus-Transformation eines Bildblocks.

Das Ergebnis dieser FDCT (Forward-DCT) sind insgesamt 64 Koeffizienten, die sich wiederum in einer 8×8 -Matrix darstellen lassen. Dabei entspricht der erste Koeffizient (links oben angenommen) dem Grundfarbwert des Blocks. Als Gleichanteil des

Blocks wird er in Anlehnung an Gleich- und Wechselstrom auch als DC-Koeffizient bezeichnet. Die weiteren 63 Koeffizienten nennt man entsprechend AC-Koeffizienten. Die Koeffizienten in der Matrixdarstellung stellen dabei von links nach rechts und von oben nach unten höhere Frequenzen dar.

Quantisierung

Da jeweils 64 Eingangspixel auch 64 Ausgangskoeffizienten liefern, wäre die erreichbare Kompression nicht sonderlich hoch. Der Vorteil der Koeffizientendarstellung ergibt sich dadurch, daß Bilder in der Praxis nur selten hohe Frequenzanteile besitzen. Dadurch haben viele der Koeffizienten nur noch sehr niedrige Werte. Im dritten Schritt, der *Quantisierung*, werden die Einträge der Koeffizientenmatrix zunächst mit jeweils einem Eintrag einer Quantisierungsmatrix mit 64 Einträgen dividiert. Die Matrix bestimmt somit gewissermaßen die Relevanz der Koeffizienten: Unwichtige Koeffizienten (meist die mit hohen Frequenzanteilen) erhalten niedrige Werte, die später mit geringer Bit-Tiefe gespeichert, oder beim anschließenden Runden auf ganzzahlige Werte direkt gleich null gesetzt werden können. Zum Dekodieren eines bereits kodierten Bildes müssen die Koeffizienten wieder mit Einträgen aus derselben Quantisierungsmatrix multipliziert werden. Pro Bild können bis zu vier verschiedene Matrizen in beliebiger Reihenfolge verwendet werden. Die Wahl der Quantisierungsmatrizen ist nicht im JPEG-Standard festgelegt. JPEG ist damit implementationsabhängig, so daß Qualitätsunterschiede zwischen verschiedenen Implementierungen auftreten.

Entropiekodierung

Im letzten Schritt, der *Entropiekodierung*, werden DC- und AC-Koeffizienten unterschiedlich behandelt. Da sich der Grundfarbton zwischen benachbarten Blöcken meist nur wenig ändert, wird ein Koeffizient von dem des vorherigen Blockes subtrahiert und die Differenz weiter verwendet. Dadurch entstehen für die DC-Koeffizienten sehr kleine Werte. Die AC-Koeffizienten werden in diagonalen Zick-Zack-Reihenfolge (siehe Abb. 4) bearbeitet. Da hohe Frequenzanteile meist vernachlässigbar sind, wird durch die Zick-Zack-Reihenfolge erreicht, daß möglichst viele Null-Koeffizienten hintereinanderliegen.

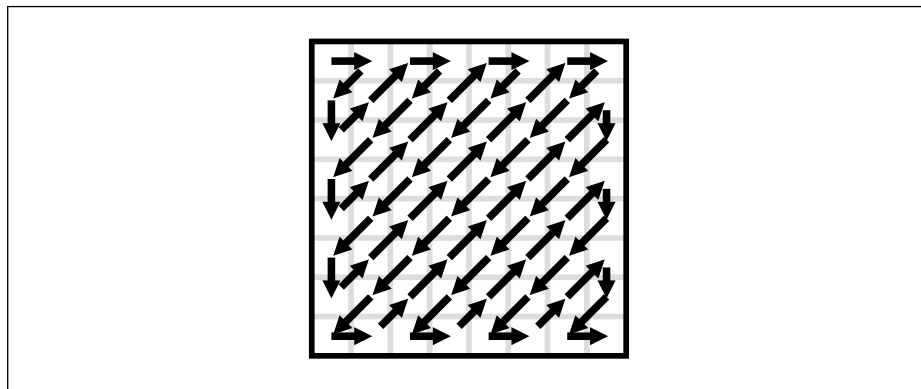


Abbildung 4: Koeffizientenbearbeitung in Zick-Zack-Reihenfolge.

Für die Speicherung der Koeffizienten können jeweils verschiedene Bit-Tiefen verwendet werden. Die Speicherung erfolgt gemäß *ISO-Intermediate-Symbol-Sequence-Format*, das im wesentlichen alternierend die drei folgenden Angaben enthält:

- Anzahl der Koeffizienten gleich Null,
- Anzahl der Bits die für die Darstellung des folgenden Koeffizienten genutzt werden,
- Wert des Koeffizienten mit der angegebenen Bit-Anzahl.

Die endgültigen Daten erhält man durch anschließende Huffman-Kodierung.

Ein Nachteil der vorgestellten Kodierung ist ihre beinahe ausschließliche Eignung für fotografische Bilder. Glatte Farbverläufe, wie sie bei computergenerierten Bildern vorkommen, können nur ungenügend dargestellt werden. Bei hoher Kompression kommt noch die Einteilung in 8×8 -Blöcke zum Tragen. Hier sind die Blockgrenzen häufig als störende Kanten im Bild sichtbar.

2.2 Formate für bewegte lineare Daten

2.2.1 Videokompression mit MPEG (Moving Picture Experts Group)

Zur gemeinsamen Kodierung von Bewegtbildern und Audiodaten hat sich in vielen Bereichen das MPEG-1-Format durchgesetzt, das seit 1993 auch ein ISO International Standard ist. Die Datenrate die für MPEG-Daten angestrebt wird, liegt bei 1,2 Mbit/s und entspricht damit der Datenrate, die CD-Spieler bei einfacher Geschwindigkeit erreichen.

Visuelle Daten werden nicht in dem später für die Darstellung nötigen RGB-Farbraum, sondern im $Y C_r C_b$ -Farbraum dargestellt. Die Y-Komponente enthält dabei die Luminanz (Helligkeit) des Bildes, die beiden C-Komponenten die Chrominanz (Farbigkeit). Da für die Bildwahrnehmung im menschliche Auge die Luminanz eine wichtigere Rolle als die Chrominanz spielt, wird die Luminanzkomponente mit entsprechend höherer Datenrate kodiert. Man erhält dadurch eine – im Gegensatz zu einer RGB-Kodierung mit gleicher Datenrate in allen drei Kanälen – deutlich bessere subjektive Qualität.

Um die Qualität darüberhinaus zu erhöhen, ist MPEG als vorzugsweise asymmetrische Kompression konzipiert. Asymmetrisch bedeutet in diesem Zusammenhang, daß für die Kodierung ein wesentlich höherer Aufwand benötigt wird als für die Dekodierung.

Der Videostrom wird bei MPEG aus vier verschiedenen Einzelbildtypen, sogenannten *Frames*, zusammengesetzt. Diese Frametypen können in beinahe beliebiger Reihenfolge liegen. Die Reihenfolge wird vom Kodierer festgelegt.

I-Bilder (Intra Coded Pictures)

I-Bilder kodieren gesamte Einzelbilder, ohne sich auf andere Bilder des Datenstroms zu beziehen. Die Daten eines I-Bildes stellen also ein Einzelbild komplett dar. Die Kompressionstechnik ist stark an die verlustbehaftete JPEG-Kompression angelehnt.

P-Bilder (Predictive Coded Pictures)

Bei P-Bildern macht man sich die Tatsache zunutze, daß sich zwischen zwei Bildern einzelne Bildbereiche häufig nicht, oder nur wenig ändern, sondern nur verschieben.

Um geringe Unterschiede zwischen Bildbereichen zu sichern, wird nur die Differenz der beiden Bereiche gesichert. Die für die Differenz benötigte Datenmenge ist deutlich geringer als für die komplette Neukodierung des Bereichs. Außerdem muß der Bewegungsvektor des Bereichs gesichert werden.

P-Bilder beziehen sich bei dieser Kodierung stets auf das vorhergehende P- oder I-Bild, deshalb muß das jeweils letzte P- oder I-Bild im Dekodierer gespeichert werden.

Weiterhin können Bildbereiche ebenso wie in I-Bildern komplett kodiert werden. Die Auswahl, welche Bestandteile neu und welche „vorhersehend“ (predictive) kodiert werden, ist dem Kodierer überlassen.

B-Bilder (Bidirectionally Predictive Coded Pictures)

Bei B-Bildern wird zusätzlich noch das nachfolgende P- oder I-Bild in die Berechnung mit einbezogen. Dadurch lassen sich Bestandteile, die beispielsweise durch die Bewegung eines Vordergrundobjekts im vorherigen Bild noch nicht verfügbar und erst im nachfolgenden Bild sichtbar sind, effizient kodieren.

D-Bilder (DC Coded Pictures)

D-Bilder sind, wie I-Bilder, als ganzes kodiert und beziehen sich nicht auf andere Bilder. Jedoch wird hier nur der DC-Koeffizient (siehe JPEG-Kompression) der Diskreten Cosinus-Transformation gespeichert. Diese Bilder werden für die normale Wiedergabe nicht benötigt und nur bei schnellem Vor- oder Rücklauf dekodiert um wenigstens eine grobe Vorstellung des Inhalts zu vermitteln. Da heutige Dekodierer meist genügend schnell sind, um I-Bilder in hoher Geschwindigkeit darzustellen, und I-Bilder im Datenstrom meist in regelmäßigen Abständen auftreten, werden D-Bilder in der Praxis kaum verwendet.

Da nur bei I-Bildern gewährleistet ist, daß das komplette Bild dekodierbar ist, müssen für einzelne P- oder B-Bilder alle Bilder seit dem letzten I-Bild mitdekodiert werden. Somit ist es nicht sinnvoll, nur ein einziges I-Bild am Anfang des Datenstroms zu platzieren und alle Folgebilder als P- oder B-Bild zu kodieren. Für einen Zugriff auf ein beliebiges Bild müßten so alle Bilder seit dem ersten I-Bild dekodiert werden, was ohne weitere Zwischenspeicherung einen viel zu hohen Rechenaufwand erfordern würde. Für praktische Anwendungen hat sich eine Kodierung in der Folge **IBBPBBPBB IBBPBBPBB**... als sinnvoll herausgestellt.

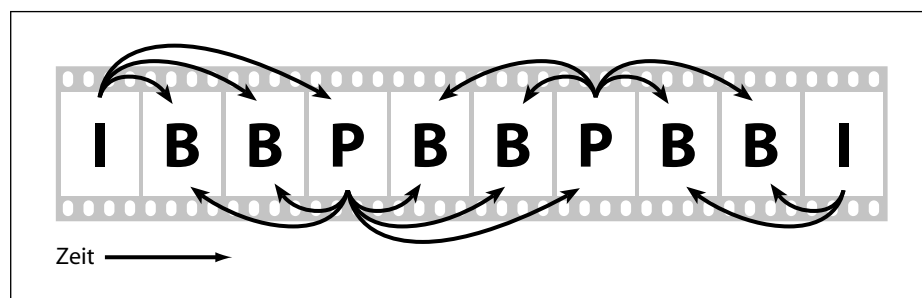


Abbildung 5: Typische MPEG-Framereihenfolge und mögliche Abhängigkeiten zwischen einzelnen Bildern.

2.2.2 Audiokompression mit MPEG Layer-3 (MP3)

Für die Kodierung des Audiostroms in MPEG-1 Daten stehen drei verschiedene Techniken mit unterschiedlichem Kodierungs- und Dekodierungsaufwand zur Verfügung. Diese sind in sogenannten *Layern* (Ebenen) definiert. Dekodierer eines höheren Layers müssen dabei immer auch Datenströme niedrigerer Ebenen dekodieren können.

Die derzeit wohl beliebteste Kodierung ist die der Ebene drei, die unter der etwas irreführenden Bezeichnung *MP3* für reine Musikkodierung verwendet wird. *MP3* steht dabei nicht für MPEG-3 (diesen Standard gibt es nämlich gar nicht), sondern für *MPEG Layer-3*.

Das MP3-Format wurde maßgeblich vom Fraunhofer Institut für Integrierte Schaltungen in Kooperation mit der Universität Erlangen entwickelt [4]. Die hohe Kompression bei gleichzeitig guter Qualität wird durch die Anwendung eines psychoakustischen Modells erreicht, das in aufwendigen Versuchen mit Testpersonen ermittelt wurde. Man macht sich dabei das Phänomen zunutze, daß Audiosignale vom menschlichen Ohr nicht wahrgenommen werden, wenn sie von anderen, meist lauterem Signalen überdeckt werden (*Maskierungseffekt*).

Tonqualität	Bandbreite	Modus	Bitrate	Kompression
Telefon	2,5 kHz	mono	8 kbps	96:1
besser als KW	4,5 kHz	mono	16 kbps	48:1
besser als AM	7,5 kHz	mono	32 kbps	24:1
ähnlich zu FM	11 kHz	stereo	56...64 kbps	26...24:1
nahe-CD	15 kHz	stereo	96 kbps	16:1
CD	>15 kHz	stereo	112...128 kbps	14...12:1

Tabelle 2: Typische MP3-Performance (Quelle: Fraunhofer IIS-A)

Für die Kodierung wird das Tonsignal zunächst in Blöcke zu mehreren *Samples* (einzelne Abtastwerte des Signals) aufgeteilt und in 32 Frequenzbereiche aufgespalten. Die Blöcke werden mittels Diskreter Cosinus-Transformation, Quantisierung und Lauflängenkodierung komprimiert. Wichtige, hörbare Frequenzbereiche werden mit relativ hoher Datenrate kodiert. Für andere Frequenzbereiche wird der *Signal-to-Mask-Ratio* ermittelt und die Datenrate im jeweiligen Frequenzbereich durch Quantisierung so weit verringert, daß das entstehende Quantisierungsrauschen (Fehler im Signal) gerade noch unter der Maskierungsschwelle liegt.

Joint Stereo

Um Stereosignale effizient kodieren zu können bietet MPEG Layer-3 als Besonderheit das *Joint Stereo*-Format. Man nutzt dabei aus, daß bei Stereosignalen meist nur geringe Unterschiede zwischen den beiden Tonkanälen bestehen [4]. Bei Joint Stereo wird die Differenz zwischen den beiden Kanälen ermittelt und mit etwa 30 % der verfügbaren Bandbreite kodiert. Das eigentliche Signal kann mit den verbleibenden 70 % ko-

diert werden. Diese Technik ermöglicht ein qualitativ insgesamt höherwertiges Signal bei gewissen Verlusten in der Kanaltrennung.

2.2.3 Echtzeit Streaming von Video- und Audiodaten via RTP und RTSP

Für die Echtzeit-Datenübertragung von audiovisuellen Daten (*Streaming*) gibt es zwei grundsätzliche Übertragungsmöglichkeiten: *Unicast* (one-to-one) und *Multicast* (one-to-many) [5].

Im Unicast-Verfahren schickt der sogenannte Medienserver jeweils einen Datenstrom direkt an einen einzelnen Empfänger. Im Multicast-Verfahren wird nur ein Datenstrom in jeden Ast eines Netzwerks übertragen und kann von mehreren Endnutzern empfangen werden. Somit muß der Medienserver nicht mehrere Datenströme verschicken – die insgesamt benötigte Bandbreite wird drastisch verringert. Für dieses Verfahren muß jeder Router im Netzwerk wissen, ob sich in den angeschlossenen Teilästen Empfänger des Datenstroms befinden. Da dieses Verfahren nicht von allen Routern unterstützt wird, ist es möglich daß Empfänger die Multicast-Pakete nicht erhalten. Um dennoch den Datenstrom zu empfangen, können Reflektoren eingesetzt werden, die den Datenstrom per Multicast vor dem betreffenden Router erhalten und diesen als einzelnen Strom (im Unicast-Verfahren) an den Endempfänger schicken.

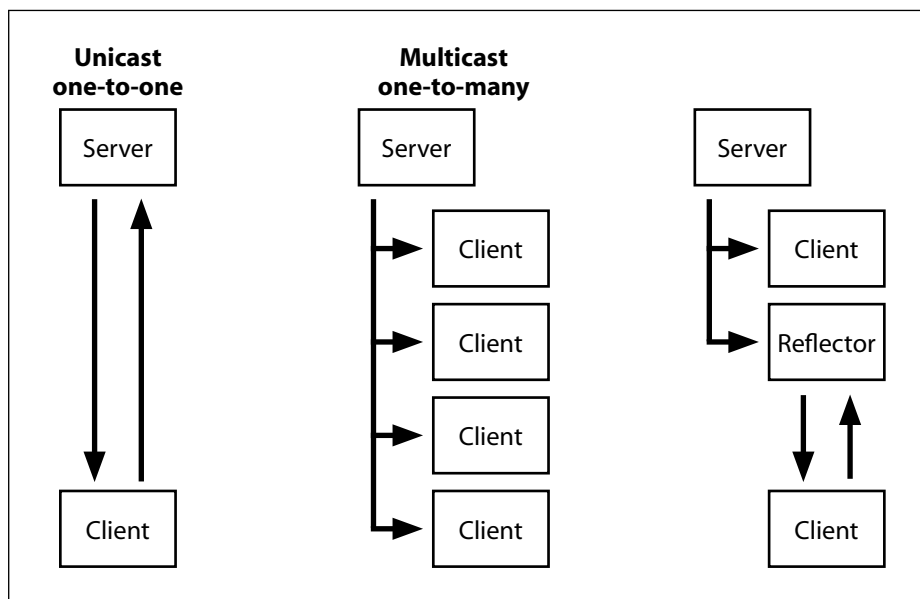


Abbildung 6: Datenstromübertragung per Unicast, Multicast und Reflektor.

Real Time Streaming Protocol (RTSP)

Zur Kontrolle und Steuerung des Datenstroms wird vielfach das *Real Time Streaming Protocol* eingesetzt [6]. Die Syntax von RTSP ist stark an das *Hypertext Transfer Protocol* (HTTP) angelehnt. Wichtige Unterschiede zu HTTP bestehen darin, daß im RTSP auch der Server Anfragen an den Client senden kann – beispielsweise um Statusinformationen wie Paketverluste abzurufen. Weiterhin wird bei RTSP der Status der Kommunikationssitzung gespeichert, so daß der Server den Client bei jeder Anfrage

eindeutig identifizieren kann und die Anfrage in Kontext zu vorherigen Anfragen stellen kann.

Operationen, die vom RTSP unterstützt werden, sind das Abrufen von Streams vom Server, die „Einladung“ eines Servers zu einer bestehenden Konferenz (entweder zur Wiedergabe oder zur Aufnahme eines Stroms), das Ankündigen von zusätzlichen verfügbaren und verfügbar werdenden Daten (beispielsweise ein Tonkanal in anderer Sprache), sowie der generelle Informationsaustausch zwischen Client und Server.

Das RTSP überträgt die Audio- und Video-Daten jedoch nicht selbst – dies muß mittels eines anderen Protokolls, beispielsweise dem *Real Time Transport Protocol* (RTP) geschehen.

Real Time Transport Protocol (RTP)

Das Real Time Transport Protocol spezifiziert die Übertragung von Medienstreams über Netzwerke [7]. Das verwendete Datenformat ist dabei nicht festgelegt und wird mittels sogenanntem *Payload Type Identifier* angekündigt. Damit kann das RTP auch für proprietäre Datenformate (QuickTime, RealVideo) verwendet werden. Die RTP-Pakete werden sequentiell nummeriert, so daß sie beim Empfänger gegebenenfalls in die korrekte Reihenfolge sortiert werden können.

Das RTP erzielt durch den Einsatz von *Mixern* und *Translatoren* eine hohe Flexibilität, was Datenquellen und -formate betrifft. Mixer können einzelne Datenströme aus unterschiedlichen Quellen kombinieren und als gemeinsamen Strom weiterversenden. Somit können Audio- und Videodaten von unterschiedlichen Servern stammen, die selbst nichts voneinander wissen. Translatoren können das Datenformat oder das Übertragungsprotokoll in ein anderes übersetzen. So könnte ein eingehender MPEG-Datenstrom in ein anderes Format übersetzt werden, so daß ein Empfänger, der MPEG nicht dekodieren kann, die Daten dennoch interpretieren kann.

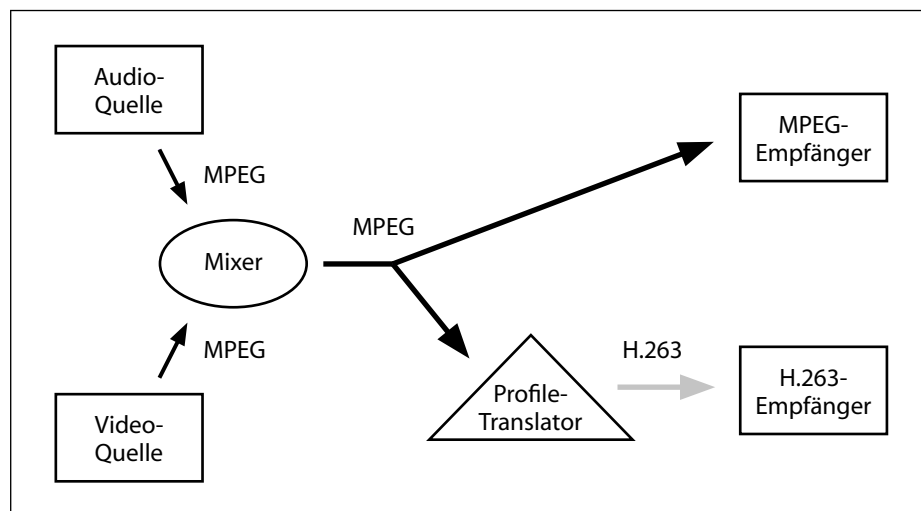


Abbildung 7: Manipulation von RTP-Datenströmen durch Mixer und Translatoren.

2.3 Formate für nichtlineare audiovisuelle Daten

2.3.1 MPEG-4

Die Entwicklung von MPEG-4 als Kodierungsstandard für audiovisuelle Programme mit extrem niedrigen Bitraten begann 1993 und liegt seit 1999 als ISO internationaler Standard vor. Eine erste Erweiterung in Form von MPEG-4 Version 2 ist seit 2000 internationaler Standard. Derzeit wird noch an Erweiterungen in einigen Bereichen gearbeitet.

Audiovisuelle Szenen sind bei MPEG-4 in hierarchischer Struktur aus sogenannten *Media Objects* zusammengesetzt. Ein Media Object beschreibt dabei Objektprimitive wie Standbilder, Videoobjekte und Audioobjekte. MPEG-4 spezifiziert dabei Objektprimitive für natürliche und synthetische Daten, sowie für zwei- und dreidimensionale Repräsentationen. Einzelne Media Objects sind so kodiert, daß sie auch ohne umgebende Objekte behandelt werden können; die wirkliche Szene entsteht aber durch die Komposition der Objekte.

MPEG-4 bietet standardisierte Wege, um eine Szene aus Objekten zusammenzustellen, beispielsweise um

- Objekte beliebig in einem gegebenen Koordinatensystem zu plazieren;
- Transformationen auszuführen, um die geometrische oder akustische Erscheinung zu modifizieren;
- Objekte zu gruppieren;
- gestreamte Daten an ein Objekt zu binden um seine Attribute zu ändern;
- interaktiv den Blickpunkt zu ändern.

Im Gegensatz zu den Vorgängern MPEG-1 und MPEG-2 legt MPEG-4 einen deutlichen Schwerpunkt auf Interaktionsmöglichkeiten. So kann es beispielsweise möglich sein, Objekte in der Szene umherzubewegen, Ereignisse auszulösen, die Auswirkungen auf die Szene haben, oder die Sprache von Tonquellen zu wählen.

MPEG-4 verabschiedet sich damit von der streng linearen Welt seiner Vorgänger und bietet so eine deutlich höhere Flexibilität für eine Vielzahl von Einsatzbereichen.

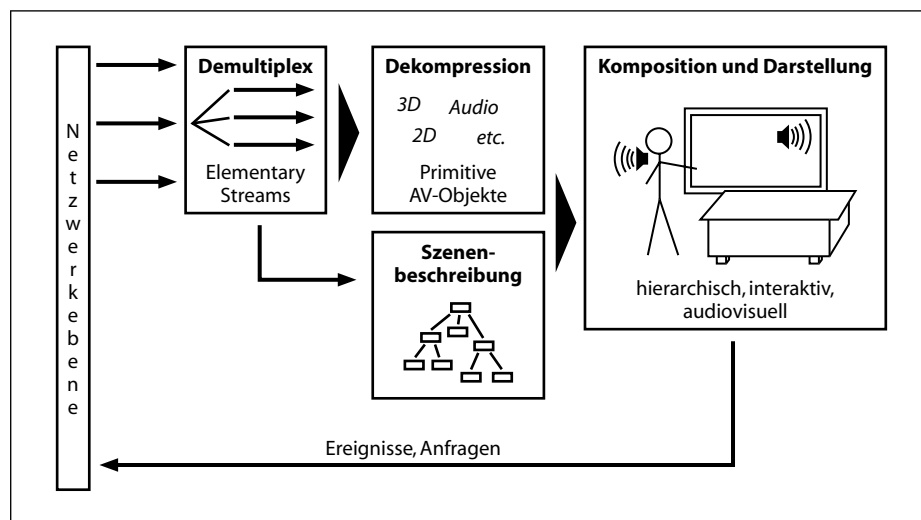


Abbildung 8: Schematische Darstellung von MPEG-4 auf Empfängerseite.

2.3.2 Dynamische Vektorgrafiken im WWW: Scalable Vector Graphics (SVG)

Scalable Vector Graphics (SVG) ist ein noch in Entwicklung befindlicher Standard zur Beschreibung zweidimensionaler Grafiken. Bilder können dabei aus Vektorgrafiken, Pixelgrafiken (GIF, JPEG, PNG) und Text bestehen.

Die Beschreibung der Vektorgrafiken und der Hierarchie geschieht mittels *Extensible Markup Language* (XML). Die textbasierte Syntax von XML ist jedoch nicht sonderlich effizient, was die Datenrate betrifft. Deshalb können SVG-Szenen mittels *gzip* komprimiert werden. Jeder SVG-Interpreter muß in der Lage sein, so komprimierte Daten darzustellen. Vorteil der XML-Beschreibung ist die nahtlose Integration in HTML- und XHTML-Webseiten.

SVG-Objekte integrieren sich in die strukturierte Beschreibung von Webseiten nach dem *Document Object Model* (DOM); so ist es beispielsweise möglich, SVG-Objekte durch die Skriptsprache JavaScript zu modifizieren. Dies ermöglicht sowohl animierte Effekte als auch interaktive Elemente.

Weiterhin lassen sich Textelementen Stile mit demselben *Cascading Style Sheet* (CSS) zuweisen, das auch für Textstile auf der enthaltenden Webseite genutzt wird. Damit kann man durch Änderung des Style Sheets schnell das Aussehen von Webseite und Grafiken modifizieren. Dies ist mit bisherigen Grafikformaten nicht möglich.

Durch die Möglichkeit, auf Schriften, Bilder und andere Elemente verweisen zu können und diese nicht in den Code einbinden zu müssen, können mehrere Grafiken auf dieselben Ressourcen zurückgreifen und sparen somit Speicherplatz und Übertragungsdauer.

2.3.3 QuickTime Virtual Reality (QTVR)

Das *QuickTime Virtual Reality* Format ermöglicht die Darstellung von dreidimensionalen Objekten (Außenansicht) oder Panoramas (Innenansicht). Im Vergleich zu aufwendiger Virtueller Realität, bei der man tatsächlich den Eindruck erhält, Teil der Szene zu sein, bietet QuickTime VR nur eine gewöhnliche zweidimensionale Darstellung auf Bildschirmen. Der Eindruck einer virtuellen Realität wird dadurch erzeugt, daß der Betrachter das Objekt um eine zentrale Achse drehen kann oder – bei Panoramas – sich selbst um die eigene Achse drehen kann, um Räume oder Landschaften im Rundumblick betrachten zu können.

Zielgruppe von QTVR ist damit weniger Forschung und Industrie, sondern der Heimanwender. Da die Objekte und Panoramas nicht durch Vektoren sondern durch aneinandergefügte Fotografien repräsentiert werden, wird zwar ein guter Bildeindruck erzeugt, jedoch ist es prinzipiell nicht möglich, sich frei in den Räumen zu bewegen: Der Beobachtungsstandpunkt ist festgelegt. Um dennoch eine gewisse Interaktion zu ermöglichen, können Bildbereiche als *Hotspots* definiert werden. Hotspots sind Verweise auf externe Webseiten oder andere Medien, oder weitere Panoramas, die in der selben Datei gespeichert sind. Damit ist eine einfache Bewegungsmöglichkeit gegeben, jedoch springt man nur von einem Standpunkt zum nächsten – ein fließender Übergang ist nicht möglich.

Um aus Fotografien einen Panoramaeindruck zu erzeugen, existieren bei QTVR zwei verschiedene Methoden. Bei der ersten Methode werden die Fotografien auf einen angenommenen Zylinder projiziert, in dessen Mitte sich der Beobachter befindet. Da hierbei Zylinderdeckel und Boden nicht mit erfaßt werden, ist der Blick nach oben und unten beschränkt. Um eine vollständige Rundumsicht zu ermöglichen, gibt es in

der neuen Version 5 kubische Panoramas. Hierbei werden sechs Fotografien auf die Innenseiten eines Würfels projiziert. Dank entsprechender Korrekturalgorithmen sind die Würfelkanten im fertigen Panorama nicht zu sehen.

3 Zukünftige Entwicklungen

Die Entwicklung von Multimedia-Datenformaten steht eigentlich erst an ihrem Anfang. Einige der vorgestellten Formate sind erst in den letzten Jahren entstanden, einige befinden sich gar noch in Entwicklung. In allen Bereichen sind dabei ähnliche Tendenzen zu erkennen. Selbstverständlich ist es ein andauerndes Ziel, die Dateigröße bei gleichbleibender Qualität weiter zu verringern. Im Bereich der unbewegten Bilder sind derzeit zum Beispiel einige konkurrierende *Wavelet*-Verfahren verfügbar oder in Entwicklung.

Eine ganz entscheidende Tendenz, wie sie beispielsweise beim MPEG-4-Format deutlich wird, ist das Zusammenführen verschiedener Inhalte in einem einzigen Format. Im Zusammenhang mit der Darstellung interaktiver Inhalte bietet sich damit die Möglichkeit, ganze Benutzerschnittstellen realisieren zu können – ein wichtiger Gesichtspunkt für Kleingeräte wie Mobiltelefone oder sogenannte *Personal Digital Assistants* (PDA), und Geräte für den Consumer-Bereich wie zum Beispiel Set-Top Boxen für Digitales Fernsehen. Auch Flexibilität spielt eine große Rolle, so daß viele Formate Wert auf einfache Erweiterbarkeit legen, um neue Technologien möglichst problemlos integrieren zu können.

4 Literaturverzeichnis

1. Adobe Systems Incorporated. *TIFF 6.0 Specification*. <http://partners.adobe.com/asn/developer/PDFS/TN/TIFF6.pdf>, 1992.
2. International Standards Organization. *Information technology – digital compression and coding of continuous-tone still images*. International Standard ISO/IEC IS 10918, 1993.
3. R. Steinmetz. *Multimedia-Technologie*, 3. Auflage. Springer-Verlag, Berlin, 2000.
4. Fraunhofer Institut für Integrierte Schaltungen. *MPEG Audio Layer-3*. <http://www.iis.fhg.de/amm/techinf/layer3/index.html>, 2001.
5. Apple Inc. *QuickTime Streaming*. <http://developer.apple.com/techpubs/quicktime/qtdevdocs/PDF/QTStreaming.pdf>, 2001.
6. H. Schulzrinne, A. Rao, R. Lanphier. *Real Time Streaming Protocol (RTSP)*. Request for Comments (RFC): 2326, April 1998.
7. Audio-Video Transport Working Group, H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson. *RTP: A Transport Protocol for Real-Time Applications*. Request for Comments (RFC): 1889, Januar 1996.
8. International Standards Organization, Working Group 11 (MPEG). *Overview of the MPEG-4 Standard*. <http://www.cselt.it/mpeg/standards/mpeg-4/mpeg-4.htm>, 2000.
9. World Wide Web Consortium (W3C). *Scalable Vector Graphics (SVG) 1.0 Specification*, W3C Candidate Recommendation. <http://www.w3.org/TR/2000/CR-SVG-20001102/CR-SVG-20001102.zip>, 2000.