

Seminar „User-Interface Design und Multimedia“
Wintersemester 2000/01

Datenformate für Multimedia-Files



Steffen Kamp
steffen.kamp@post.rwth-aachen.de

Datenformate für Multimedia-Files

1 Übersicht und Zielsetzungen von Multimediadatenformaten

2 Formate im Detail

2.1 Unbewegte Bilder

2.2 Lineare Formate

2.3 Nichtlineare Formate

3 Zukünftige Entwicklungen

1

Übersicht und Zielsetzungen von Multimediadatenformaten

Formatvielfalt

Datenformate		Video Kompressoren	Audio Kompressoren
3DMF	MPEG 4	Animation	X-bit integer
AI	PCD	Apple BMP	X-bit floating point
AIFF	PCX	Apple Video	ALaw
AU	PDF	Cinepak	AU
AVI	Photoshop	Component video	G.721 ADPCM
BMP	PICS	DV	G.723 ADPCM
DV	PICT	H.261	GSM
DXF	PNG	H.263	IMA
EPSF	PostScript	Intel Indeo Video	MACE
FlashPix	QuickTime Image File	Microsoft RLE	MPEG Audio
GIF	QuickTime Movie	Microsoft Video	MS ADPCM
IFF	QuickTime VR	Motion JPEG	QDesign Music
JPEG2000	Scitex CT	Photo JPEG	Qualcomm PureVoice
JPEG/JFIF	SGI	Planar RGB	µLaw
LWF	SVG	Sorenson Video	
MacPaint	Targa		
Macromedia Flash	TIFF		
MIDI	VRML		
MOD	WAVE		
MPEG 1	WMF		
MPEG 1, Layer 3	X11-Bitmap		
MPEG 2			

→ **Klassifizierung nötig**

Klassifizierung von Multimediaformaten

Unbewegte Bilder

Zeitlicher Ablauf

Klassifizierung von Multimediaformaten

Unbewegte Bilder

- └ Pixelbasiert
- └ Vektorbasiert

Zeitlicher Ablauf

Klassifizierung von Multimediaformaten

Unbewegte Bilder

- Pixelbasiert
- Vektorbasiert

Zeitlicher Ablauf

- **linear**
- **nichtlinear (interaktiv)**

Klassifizierung von Multimediaformaten

Unbewegte Bilder

- Pixelbasiert
- Vektorbasiert

Zeitlicher Ablauf


- **linear**
 - audio
 - video
- **nichtlinear (interaktiv)**
 - 3D
 - Flash/SVG
 - Metaformate

Zielsetzungen von Datenformaten

Abhängig von

- Einsatzgebiet
- Distributionsweg

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



Formate im Detail

2.1 Unbewegte Bilder

Tagged Image File Format – TIFF

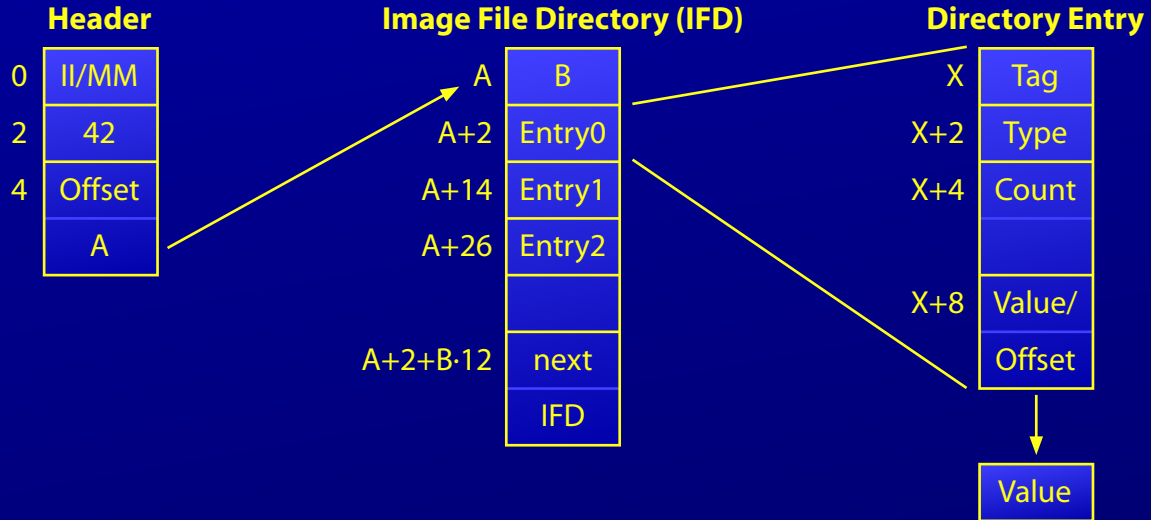
Fähigkeiten

- Vielfältige Farbmodi und Farbräume
- Verlustlose Komprimierung
- Erweiterbar für zukünftige Anforderungen
- Einbinden unbegrenzter eigener (nicht im Standard definierter) Informationen möglich

Einsatzgebiete

- Professionelle Publikationserstellung
- Verlustlose Datenübertragung (z. B. von Scannern)

Struktur von TIFF-Dateien



Joint Photographic Experts Group – JPEG

Anforderungen

- Einsetzbar für jedes *Höhe zu Breite* Verhältnis
- Komplexität des Verfahrens muß Software-Lösung auf Standardprozessoren ermöglichen
- Verlustlose und verlustbehaftete Komprimierung
- *Sequentielles* und *progressives* Dekodieren

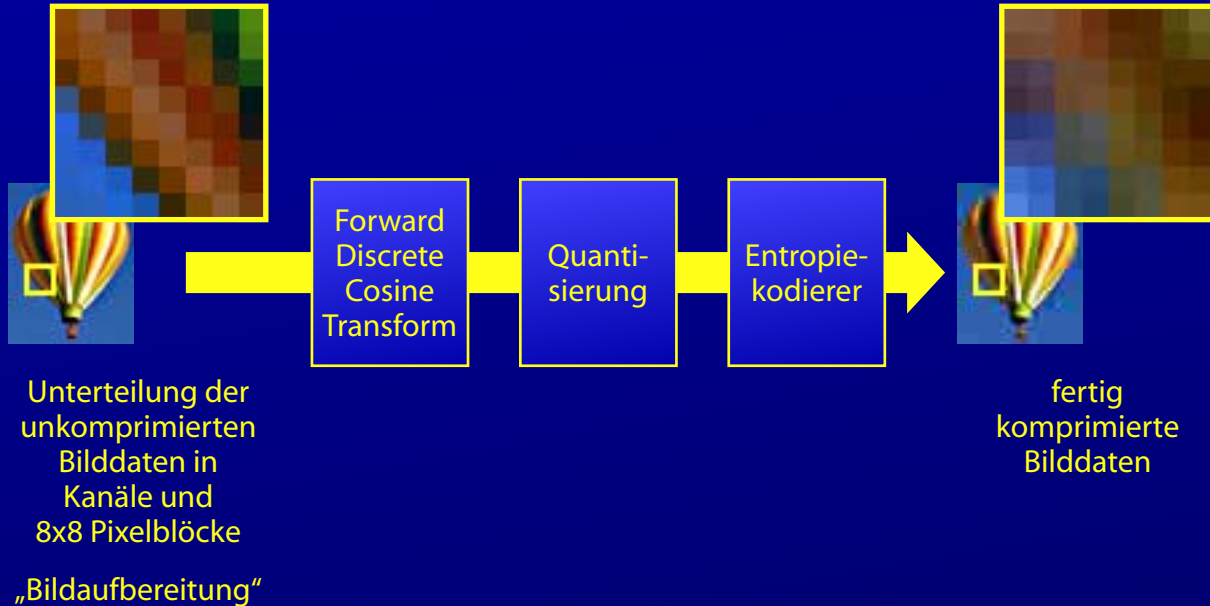
Grenzen

- Nur für Fotografien gut geeignet
- Schlechte Qualität bei mittleren bis hohen Kompressionsraten
- Kein Schutz gegen Übertragungsfehler

Einsatzgebiete

- Internet
- Fotografien (Digitalkameras, Bildarchive)

Schritte der Baseline JPEG-Kodierung

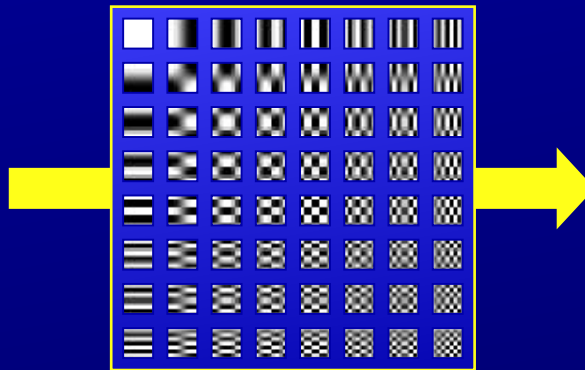


Diskrete Cosinus-Transformation

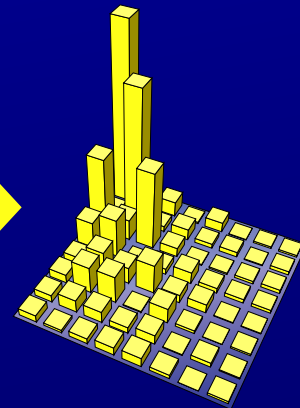
- Angelehnt an die diskrete Fourier-Transformation
- Zerlegung der Bilddateneinheiten in 2D-Frequenzanteile
- Jeder Koeffizient beschreibt Teil des 8x8 Kanalpixelblocks



8x8 Eingabematrix



Gewichtung der 64
Koeffizienten



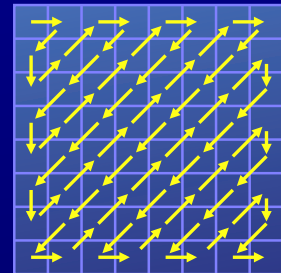
8x8 Koeffizientenmatrix
(Beträge)

Quantisierung

- Division der 64 Koeffizienten mit entsprechendem Eintrag einer Quantisierungsliste mit 64 Einträgen
- Runden liefert endgültige Koeffizienten

Entropiekodierung

- „Zick-Zack“-Bearbeitungsreihenfolge führt zu längeren Folgen ähnlicher Datenbytes
- Huffman- oder arithmetische Kodierung



Zick-Zack-Reihenfolge
nach steigender Frequenz



Formate im Detail

2.2 Lineare Formate

Moving Picture Experts Group – MPEG-1

Ziele und Anforderungen

- Gemeinsame Kodierung von Video und Audiodaten
- Angestrebte Datenrate von etwa 1,2 Mbit/s – entsprechend der CD-Technologie
- Vorzugsweise Asymmetrische Kompression
- Drei-Komponenten Bilder (Y, C_r, C_b -Farbraum)
- Unterschiedliche Datenrate der Farbkomponenten

MPEG Frametypen

Vier verschiedene Bildarten:

I-Bilder – Intra Coded Picture

- Ohne Informationen anderer Einzelbilder – „Standbild“
- JPEG-ähnliche Kompression
- Bilden Ankerpunkte für wahlfreien Zugriff

P-Bilder – Predictive Coded Picture

- Teile können wie in I-Bildern kodiert werden
- Teile können sich auf vorhergehendes I-Bild beziehungsweise P-Bild beziehen
 - Speicherung von
 1. Bewegungsvektor
 2. Differenzen mittels DCT

MPEG Frametypen (Fortsetzung)

B-Bilder – Bidirectionally Predictive Coded Picture

- Möglichkeiten wie P-Bilder
- Einbeziehung des *nachfolgenden* I-Bildes beziehungsweise P-Bildes möglich

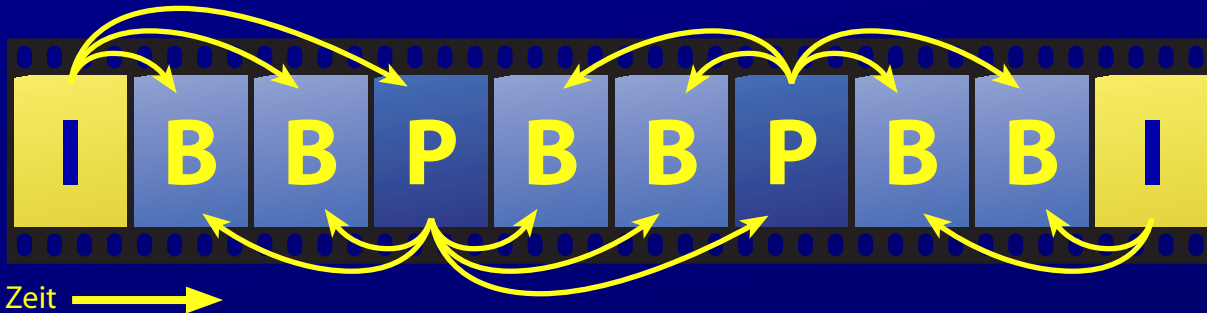
D-Bilder – DC Coded Picture

- Nur DC-Koeffizienten werden gespeichert
- Anzeige bei schnellem Vorlauf
- Überflüssig bei geeigneter periodischer Speicherung von I-Bildern

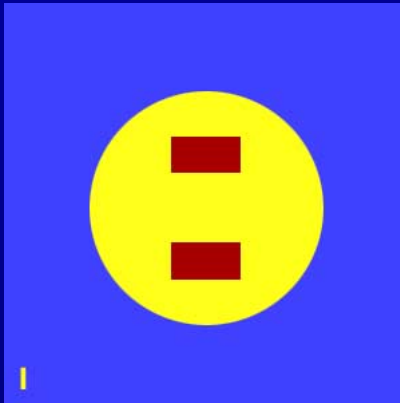
MPEG Frametypen (Fortsetzung)

- Reihenfolge der Frametypen bleibt dem Kodierer überlassen
- Forderung nach idealem wahlfreiem Zugriff würde durch ausschließliche I-Bild-Kodierung erreicht
- Forderung nach hoher Kompression würde durch Kodierung mit möglichst vielen B-Bildern erreicht
- Für die Praxis hat sich eine Folge von IBBPBBPBB IBBPBBPBB... bewährt

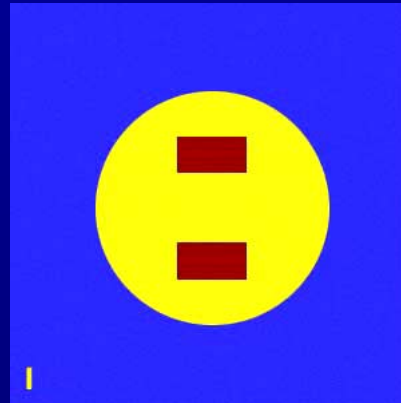
Mögliche Bildabhängigkeiten:



MPEG Demonstration



Originalfilm



MPEG kodierter Film

MPEG Layer 3 – MP3

- Wahrnehmungsbasierte Kompression
- Gute Qualität bei Kompressionsraten von 1:10
- Effektive Kodierung von Stereo-Signalen (Joint Stereo)

Kompressionsverfahren

- Aufspaltung des Signals in 32 Frequenzbereiche
- Psychoakustisches Modell ermittelt Signal-to-Mask-Ratio
- Kodierung mittels DCT
- Quantisierung
- Huffman-Kodierung

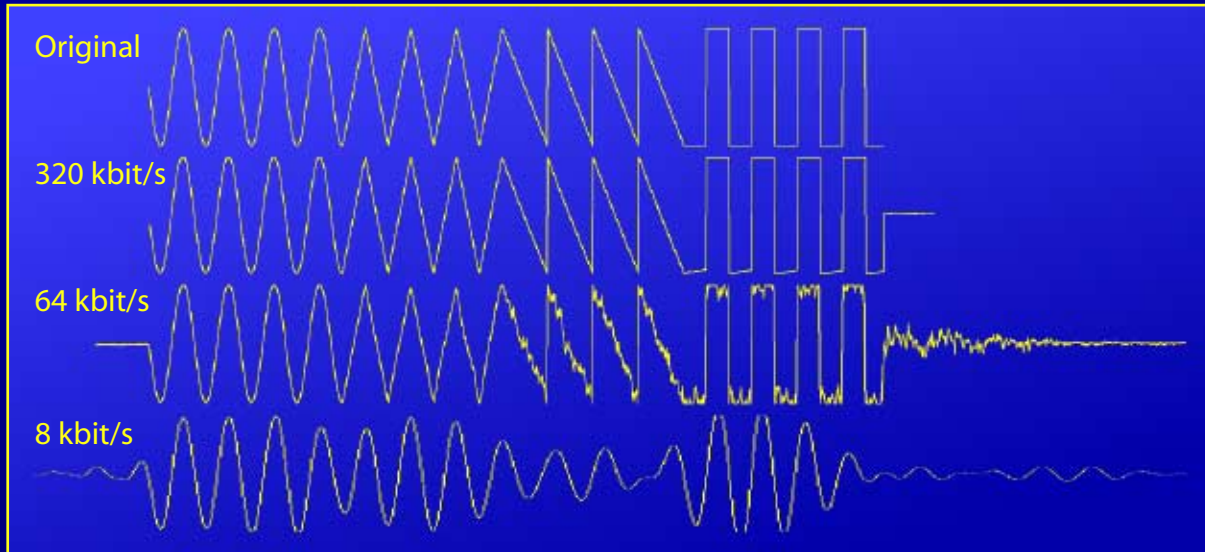
Typische MP3 Performance

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

Quelle: Fraunhofer IIS-A

MP3 Kodierungsqualität

- Sinus, Dreieck, Sägezahn und Rechteck bei 440 Hz
- Kodierung in Mono



MP3 Hörbeispiele

- Alle Dateien mit *Joint Stereo* kodiert

5 Years
(Björk)

64 kbit/s

96 kbit/s

128 kbit/s

Cosmic Girl
(Jamiroquai)

64 kbit/s

96 kbit/s

J. S. Bach
Goldberg
Variationen
(Glenn Gould)

40 kbit/s

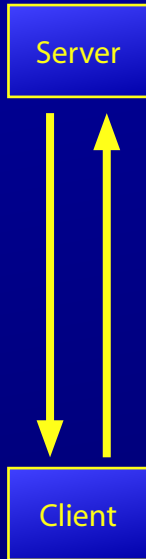
64 kbit/s

96 kbit/s

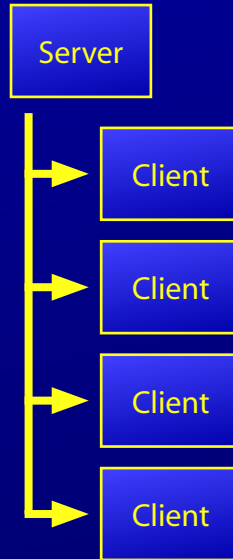
Echtzeit-Streaming via RTP/RTSP

Unicast und Multicast

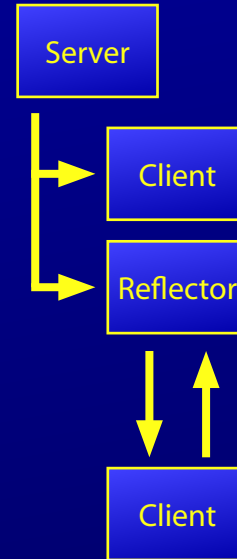
**Unicast
one-to-one**



**Multicast
one-to-many**



**Multicast
über Reflektor**



Real-Time Streaming Protocol – RTSP

- Dient Client als „Fernbedienung“ für den Medien-Server
- Liefert Audio-/Videodaten nicht selbst
- Syntax und Vorgehen ähnlich zu HTTP mit Ausnahmen:
 - Server speichert Status der Kommunikation mit dem Client
 - Anfragen von Client und Server möglich

Unterstützte Operationen

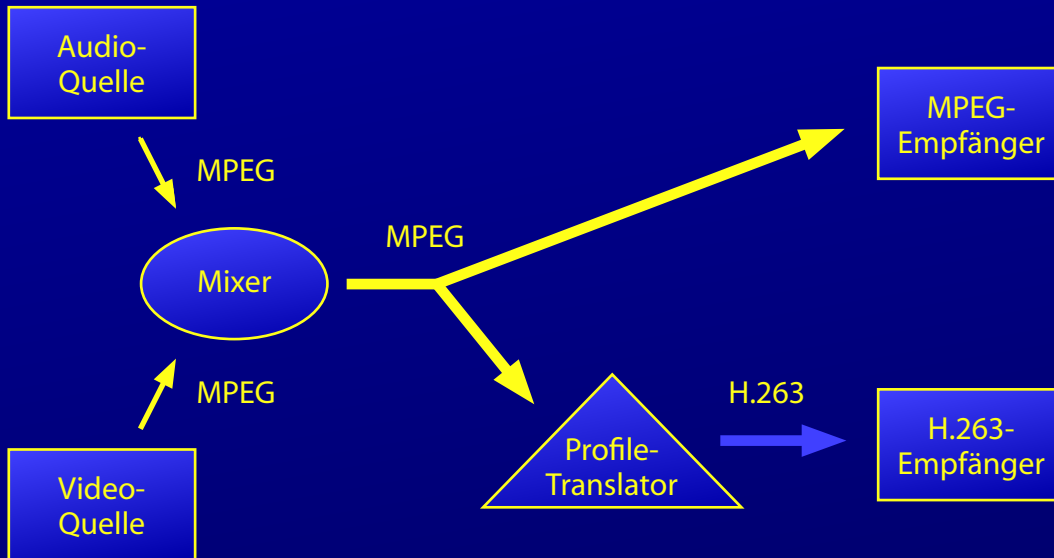
- Abrufen von Streams vom Medien-Server
Methoden: DESCRIBE, SETUP, PLAY, PAUSE, TEARDOWN
- „Einladung“ eines Servers zu bestehender Konferenz
Methoden: RECORD, PLAY
- Ankündigen von zusätzlichen Medien
Methode: ANNOUNCE
- Informationsaustausch
Methoden: ANNOUNCE, GET_PARAMETER, SET_PARAMETER, REDIRECT, OPTIONS

Real-Time Transport Protocol – RTP

- Protokoll kann an applikationsspezifische Bedürfnisse angepaßt werden (Header Extensions)
- Beschreibung der transportierten Daten mittels *Payload Type Identifier*
- Pakete werden fortlaufend nummeriert und mit Zeitinformationen versehen
- Rückkopplungsinformation via RTCP – *Real-Time Control Protocol*
- Heterogenitätsunterstützung durch Übersetzer (*Translator*) und *Mixer*

RTP – Translator und Mixer

- Mixer können Streams mehrerer Quellen zu einem mischen
- Translatoren übersetzen beispielsweise Übertragungsprotokolle oder Datenformate





Formate im Detail

2.3 Nichtlineare Formate

MPEG-4

Ziele

- Audiovisuelle Kodierung mit niedrigen Bitraten
- Flexibilität
- Kein fester Standard

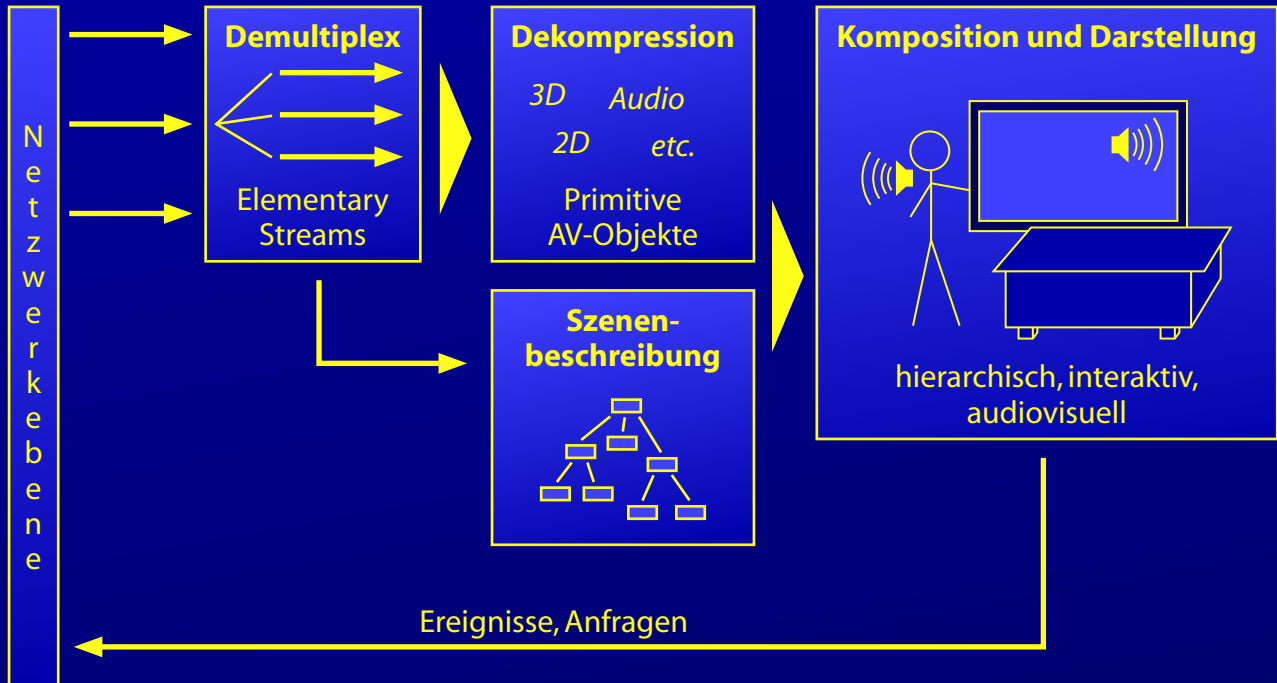
Was bietet MPEG-4?

- Repräsentation von Audio- und Video-Daten als *Media Objects*
- Gruppierung von Media Objects zu audiovisuellen Szenen
- Transport der Daten über Netzwerke
- Interaktion auf Benutzerseite

Media Objects

- Natürliche und synthetische, 2- oder 3-dimensionale Daten
- Ganze Reihe standardisierter Datentypen
- Eigene Datentypen möglich
- Hierarchische Struktur

MPEG-4 auf Empfängerseite



Scalable Vector Graphics – SVG

- XML-Beschreibung von zweidimensionalen Grafiken, dadurch „Menschenlesbar“
- Komprimierung durch *gzip* möglich
- Drei Grafik-Objekttypen: Vektorzeichnungen, Bilder und Text
- Grafiken sind dynamisch und interaktiv
 1. durch eingebetteten Code
 2. Document Object Model (DOM) ist skriptbar (JavaScript)
- Unterstützung vieler Standards (XML, XSL, HTML4, XHTML, DOM, CSS2, SMIL, WAI, Unicode)
- Verwendung von Schriften, Pixelbildern und Style Sheets entweder eingebunden oder extern referenziert

SVG Codebeispiel

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 20000303 Stylable//EN" "http://
www.w3.org/TR/2000/03/WD-SVG-20000303/DTD/svg-20000303-stylable.dtd">

<svg>
  <g transform="translate(160 150)">
    <circle id="circle1" style="fill:#ffff1a;opacity:.8"
      cx="10" cy="0" r="50"/>
    <circle id="circle2" style="fill:#4040ff;opacity:.7"
      cx="20" cy="30" r="40"/>
    <circle id="circle3" style="fill:#990000;opacity:.5"
      cx="0" cy="10" r="30"/>
    <circle id="circle4" style="fill:#4040ff;opacity:.7"
      cx="10" cy="-5" r="20"/>
  </g>
</svg>
```



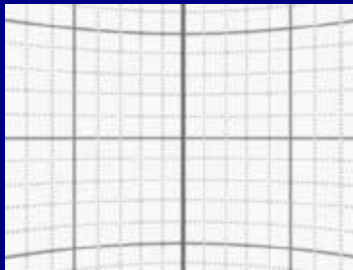
Darstellung
des obigen Codes

QuickTime Virtual Reality – QTVR

- Keine VR im eigentlichen Sinn
- Darstellung durch Pixelbilder
- Navigation intern und extern durch *Hotspots*

Zwei Filmtypen

- Objekte (Außenansicht)
- Panoramas (Innenansicht)
 - Zylindrisch
 - Kubisch



Zylindrisches Panorama



Kubisches Panorama

QTVR Demonstration



Zylindrisches Panorama mit Hotspots



Kubisches Panorama



Rotationsobjekt

3 Zukünftige Entwicklungen

Neue Kompressionstechniken

- Höhere Qualität bei stärkerer Kompression
- Wavelet-Verfahren

Neue Anforderungen

- Vielseitige „Containerformate“
- Einfach erweiterbar
- Interaktivität (Video on Demand)

Q&A