Multimedia-Systems: Compression

Prof. Dr.-Ing. Ralf Steinmetz
Dr. L.Wolf, Dr. S.Fischer

TU Darmstadt - Darmstadt University of Technology,
Dept. of Electrical Engineering and Information Technology, Dept. of Computer Science
KOM - Industrial Process and System Communications, Tel.+49 6151 166151,
Merckstr. 25, D-64283 Darmstadt, Germany, Ralf.Steinmetz@KOM.tu-darmstadt.de Fax. +49 6151 166152

GMD - German National Research Center for Information Technology
IPSI - Integrated Publication and Information Systems Institute, Tel.+49 6151 869869
Dolivostr. 15, D-64293 Darmstadt, Germany, Ralf.Steinmetz@darmstadt.gmd.de Fax. +49 6151 869870
Scope

Usage

Applications
- Learning & Teaching
- Design
- User Interfaces

Services

Content Processing | Documents | Security | ... | Synchronization | Group Communications

Systems

Databases | Programming
- Media-Server | Operating Systems | Communications
- Opt. Memories | Quality of Service | Networks

Basics

Computer Architectures | Compression
- Image & Graphics
- Animation
- Video
- Audio
Contents

1. Motivation
2. Requirements - General
3. Fundamentals - Categories
4. Source Coding
5. Entropy Coding:
6. Hybrid Coding: Basic Encoding Steps
7. JPEG
8. H.261 and related ITU Standards
9. MPEG-1
10. MPEG-2
11. MPEG-4
12. Wavelets
13. Fractal Image Compression
14. Basic Audio and Speech Coding Schemes
15. Conclusion
1. Motivation

Digital video in computing means for

- **Text:**
  - 1 page with 80 char/line and 64 lines/page and 2 Byte/Char
  - \(80 \times 64 \times 2 \times 8 = 80\) kBit/page

- **Image:**
  - 24 Bit/Pixel, 512 x 512 Pixel/image
  - \(512 \times 512 \times 24 = 6\) MBit/Image

- **Audio:**
  - CD-quality, samplerate 44,1 kHz, 16 Bit/sample
  - Mono: \(44,1 \times 16 = 706\) kBit/s
  - Stereo: \(1.412\) MBit/s

- **Video:**
  - full frames with 1024 x 1024 Pixel/frame, 24 Bit/Pixel, 30 frames/s
  - \(1024 \times 1024 \times 24 \times 30 = 720\) MBit/s
  - more realistic
  - \(360 \times 240\) Pixel/frame = 60 MBit/s

Hence compression is *necessary*
2. Requirements - General

- High quality
- Low delay
- Low complexity (e.g., ease of decoding)
- Efficient implementation (e.g., memory req.)
- Intrinsic scalability

Diagram:
- Compression
- Low delay
- Intrinsic scalability
- High quality
- Low complexity (e.g., ease of decoding)
- Efficient implementation (e.g., memory req.)
Requirements

Dialogue and retrieval mode requirements:
• Independence of frame size and video frame rate
• Synchronization of audio, video, and other media

Dialogue mode requirements:
• Compression and decompression in real-time (e.g. 25 frames/s)
• End-to-end delay < 150ms

Retrieval mode requirements:
• Fast forward and backward data retrieval
• Random access within 1/2 s

Software and/or hardware-assisted implementation requirements
3. Fundamentals - Categories

- **Entropy Coding**
  - ignoring semantics of the data
  - lossless

- **Source Coding**
  - based on semantic of the data
  - often lossy

- **Channel Coding**
  - adaptation to communication channel
  - introduction of redundancy

- **Hybrid Coding**
  - entropy and source coding
## Categories and Techniques

<table>
<thead>
<tr>
<th>Entropy Coding</th>
<th>Run-Length Coding</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Huffman Coding</td>
</tr>
<tr>
<td></td>
<td>Arithmetic Coding</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source Coding</th>
<th>Prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DPCM</td>
</tr>
<tr>
<td></td>
<td>DM</td>
</tr>
</tbody>
</table>

| Transformation | FFT         |
|               | DCT         |

| Layered Coding | Bit Position |
|               | Subsampling  |
|               | Sub-Band Coding |

<table>
<thead>
<tr>
<th>Vector Quantization</th>
<th></th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Hybrid Coding</th>
<th>JPEG</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MPEG</td>
</tr>
</tbody>
</table>

| H.261, H.263 | proprietary: Quicktime, ... |
4. Source Coding

**DPCM**

DPCM = Differential Pulse-Code Modulation

Assumptions:
- Consecutive samples or frames have similar values
- Prediction is possible due to existing correlation

Fundamental Steps:
- Incoming sample or frame (pixel or block) is predicted by means of previously processed data
- Difference between incoming data and prediction is determined
- Difference is quantized

Challenge: optimal predictor

Further predictive coding technique:
- Delta modulation (DM): 1 bit as difference signal
Source Coding: Transformation

Assumptions:
• Data in the transformed domain is easier to compress
• Related processing is feasible

Example:

FFT: Fast Fourier Transformation
DCT: Discrete Cosine Transformation
Source Coding: Sub-Band

Assumption:
• Some frequency ranges are more important than others

Example:

Application:
• vocoder for speech communication
• MPEG audio
5. Entropy Coding:

Run-Length

Assumption:
• Long sequences of identical symbols

Example:
Entropy Coding: Huffman

Assumption:
• Some symbols occur more often than others
• E.g., character frequencies of the English language

Fundamental principle:
• Frequently occurring symbols are coded with shorter bit strings

Example:
• Characters to be encoded: A, B, C, D, E
• probability to occur: \( p(A) = 0.3 \), \( p(B) = 0.3 \), \( p(C) = 0.1 \), \( p(D) = 0.15 \), \( p(E) = 0.15 \)

![Coding Tree](attachment:image.png)
Entropy Coding: Huffman

Table and example of application to data stream

<table>
<thead>
<tr>
<th>symbol</th>
<th>code</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>11</td>
</tr>
<tr>
<td>B</td>
<td>10</td>
</tr>
<tr>
<td>C</td>
<td>011</td>
</tr>
<tr>
<td>D</td>
<td>010</td>
</tr>
<tr>
<td>E</td>
<td>00</td>
</tr>
</tbody>
</table>

symbol code

<table>
<thead>
<tr>
<th>B</th>
<th>A</th>
<th>C</th>
<th>D</th>
<th>A</th>
<th>B</th>
<th>E</th>
<th>B</th>
<th>A</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>11</td>
<td>011</td>
<td>010</td>
<td>11</td>
<td>10</td>
<td>00</td>
<td>10</td>
<td>11</td>
<td>00</td>
</tr>
</tbody>
</table>

Other types of Entropy encoding

- Arithmetic Encoding (1)
6. Hybrid Coding: Basic Encoding Steps

- **Video:**
  - Lossy
- **Audio:**
  - Lossless (sometimes lossless)

**Source Data**

**Data Preparation**
- e.g. resolution, frame rate

**Data Processing**
- e.g. DCT, sub-band coding

**Quantization**
- e.g. linear, DC, AC values

**Entropy Encoding**
- e.g. runlength, Huffman

**Compressed Data**
7. JPEG

“JPEG”: Joint Photographic Expert Group

International Standard:
• For digital compression and coding of continuous-tone still images:
  • Gray-scale
  • Color
• Since 1992

Joint effort of:
• ISO/IEC JTC1/SC2/WG10
• Commission Q.16 of CCITT SGVIII

Compression rate of 1:10 yields reasonable results
JPEG

Very general compression scheme

  Independence of:
  • Image resolution
  • Image and pixel aspect ratio
  • Color representation
  • Image complexity and statistical characteristics

  Well-defined interchange format of encoded data

Implementation in:
  • Software only
  • Software and hardware

  “MOTION JPEG” for video compression
  • Sequence of JPEG-encoded images
JPEG - Compression Steps

MCU: Minimum Coded Unit
FDCT: Forward Discrete Cosine Transformation
JPEG - Image Preparation

Planes:
- \( 1 \leq N \leq 255 \) components \( C_i \) (e.g., one plane per color)
- Different resolution of individual components possible

Pixel resolution:
- 8 or 12 bit per pixel in lossy modes
- 2 to 16 bit per pixel in lossless mode

Data units: samples in lossless mode, blocks with 8x8 pixels in other modes
JPEG - Image Preparation

Example 4:2:2 YUV, 4:1:1 YUV, and YUV9 Coding

- **Luminance (Y):**
  - brightness
  - sampling frequency 13.5 MHz

- **Chrominance (U, V):**
  - color differences
  - sampling frequency 6.75 MHz
JPEG - Image Preparation

Non-interleaved encoding:

Interleaved encoding:

Minimum Coded Unit (MCU):
• Combination of interleaved data units of different components
Baseline mode is mandatory for all JPEG implementations:

- Often restricted to certain resolution
- Often only three planes with predefined color set-up

Image preparation:

- Pixel resolution of p=8 bit
- 8 x 8 pixel blocks (data units)
JPEG - Baseline Mode: Image Processing

**Forward Discrete Cosine Transformation (FDCT):**

\[
S_{vu} = \frac{1}{4} C_u C_v \sum_{x=0}^{7} \sum_{y=0}^{7} s_{yx} \cos \left( \frac{(2x+1)u\pi}{16} \right) \cos \left( \frac{(2y+1)v\pi}{16} \right)
\]

*with:*

\[
c_u, c_v = \frac{1}{\sqrt{2}}, \text{ for } u, v = 0; \text{ else } c_u, c_v = 1
\]

Formula applied to each block for all \(0 \leq u, v \leq 7:*

- Blocks with 8x8 pixel result in 64 DCT coefficients:
  - 1 DC-coefficient \(S_{00}\): basic color of the block
  - 63 AC-coefficients: (likely) zero or near-by zero values

**Different significance of the coefficients:**

- **DC**: most important
- **AC**: less important
JPEG - Baseline Mode: Image Processing

FDCT transforms:
- blocks into blocks
- not pixels into pixels

Example:
- Calculation of $S_{00}$
JPEG - Baseline Mode: Quantization

Use of quantization tables for the DCT-coefficients:

- Map interval of real numbers to one integer number
- Allows to use different granularity for each coefficient
JPEG - Quantization Effect
JPEG - Baseline Mode: Entropy Encoding

DC-coefficients:

- **Compute the differences:**

  \[
  \text{DIFF} = \text{DC}_i - \text{DC}_{i-1}
  \]

- **Use differences instead of the DC}_i values**
63 AC coefficients:

- **Ordering in ‘zig-zag’ form**

  reason: coefficients in lower right corner are likely to be zero

- **Huffman coding of all coefficients:**
  - Transformation into a code
    where amount of bits depends on frequency of respective value

- **Subsequent runlength coding of zeros**
JPEG - 4 Modes of Compression

- lossy sequential DCT-based mode (baseline mode)
- expanded lossy DCT-based mode
- lossless mode
- hierarchical mode
JPEG - Extended Lossy DCT-Based Mode

Pixel resolution 8 to 12 bit

Sequential image display:
• Top to bottom
• Good for small images and fast processing

Progressive image display:
• Coarse to fine
• Good for large and complicated images
JPEG - Extended Lossy DCT-Based Mode

Principle:
- Coefficients stored in buffer after quantization
- Order of pixel/block processing changed

By spectral selection:
- Selection according to importance of DC, AC value
- All DC values of whole image first
- All AC values in order of importance subsequently

By successive approximation:
- Selection according to position of bits
- First the most significant bit of all blocks
- Then the second significant bit of all blocks
- Until the least significant bit of all blocks
JPEG - Lossless Mode

Image preparation:
- On pixel basis (2-16 bit/pixel)

Image processing:
- Selection of a predictor for each pixel

<table>
<thead>
<tr>
<th>code</th>
<th>prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>no prediction</td>
</tr>
<tr>
<td>1</td>
<td>x=A</td>
</tr>
<tr>
<td>2</td>
<td>x=B</td>
</tr>
<tr>
<td>3</td>
<td>x=C</td>
</tr>
<tr>
<td>4</td>
<td>x=A+B+C</td>
</tr>
<tr>
<td>5</td>
<td>x=A+((B-C)/2)</td>
</tr>
<tr>
<td>6</td>
<td>x=B+((A-C)/2)</td>
</tr>
<tr>
<td>7</td>
<td>x=(A+B)/2</td>
</tr>
</tbody>
</table>

Entropy coding:
- Same as lossy mode
- Code of chosen predictor and its difference to the actual value
JPEG - Hierarchical Mode

Coding of each image with several resolutions:

- Image scaling
- Differential encoding
- First, coded with lowest resolution - image A
- Coded with increasing horizontal & vertical resolution - image A’
- Difference between both images is computed - B = A - A’
- Iteration for higher resolutions

Features:

- Requires more storage and higher data rate
- Fast decoding process
- Used for scalable video
- Similar to Photo-CD
8. H.261 and related ITU Standards

Video codec for audiovisual services at p x 64kbit/s:
• CCITT standard from 1990
• For ISDN
• With p=1,..., 30

Technical issues:
• Real-time encoding/decoding
• Max. signal delay of 150ms
• Constant data rate
• Implementation in hardware (main goal) and software
H.261 - Image Preparation

Fixed source image format

- **Image components:**
  - Luminance signal (Y)
  - Two color difference signals (C_b, C_r)
- **Subsampling according to CCIR 601 (4:1:1)**

Quarter Common Intermediate Format (QCIF) resolution: Mandatory

- Y: 176 x 144 pixel
- At 29.97 frames/s appr. 9.115 Mbit/s (uncompressed)

Common Intermediate format (CIF) resolution: Optional

- Y: 352 x 288 pixel
- At 29.97 frames/s appr. 36.46 Mbit/s (uncompressed)

Layered structure:

- Block of 8 x 8 pixels
- Macroblock of: 4 Y blocks, 1 C_r block, 1 C_b block
- Group of blocks (GOBs) of 3 x 11 macroblocks
- Picture:
  - QCIF picture: 3 GOBs
  - CIF picture: 12 GOBs
H.261 - Image Compression

Intraframe coding:
- DCT as in JPEG baseline mode

Interframe coding, motion estimation:

- Search of similar macroblock in previous image
- Position of this macroblock defines motion vector
- Search range is up to the implementation:
  - i.e., motion vector may always be 0
H.261 - Image Compression

Interframe coding, further steps:

• **Results:**
  • Difference between similar macroblocks
  • Motion vector

• **Difference of macroblocks:**
  • DCT if value higher than a specific threshold
  • No further processing if value less than this threshold

• **Motion vector:**
  • Components are coded yielding code words of variable length

Quantization:

• **Linear**

• **Adaptation of step size** \( \neq \) **constant data rate**
Further ITU Video Schemes (H.263, H.3xx)

H.263
- extension to H.261
- max. bitrate: H.263 approx. 2.5 x H.261

Source Image Formats

<table>
<thead>
<tr>
<th>Format</th>
<th>Pixels</th>
<th>H.261 Encoder</th>
<th>H.261 Decoder</th>
<th>H.263 Encoder</th>
<th>H.263 Decoder</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQCIF</td>
<td>128 x 96</td>
<td>optional</td>
<td></td>
<td>required</td>
<td></td>
</tr>
<tr>
<td>QCIF</td>
<td>176 x 144</td>
<td>required</td>
<td></td>
<td>required</td>
<td></td>
</tr>
<tr>
<td>CIF</td>
<td>352 x 144</td>
<td>optional</td>
<td></td>
<td>optional</td>
<td></td>
</tr>
<tr>
<td>4CIF</td>
<td>704 x 576</td>
<td>not defined</td>
<td></td>
<td></td>
<td>optional</td>
</tr>
<tr>
<td>16CIF</td>
<td>1408 x 1152</td>
<td></td>
<td></td>
<td></td>
<td>optional</td>
</tr>
</tbody>
</table>
H.263

Differences of H.263 compared to H.261

- optimal PB-frames (2 combined pictures: 1 B- & 1 P-Frame)
- optional overlapped block motion compensation
- optional motion vector pointing outside image
- half pel motion compensation (instead of full pel)
- JPEG is the still picture mode
- no included error detection and correction
- ..
H.320, H.32x Family

H.320 specifies (as overview) videophone for ISDN

H.310
• adapt MPEG 2 for communication over B-ISDN (ATM)

H.321
• define videoconferencing terminal for B-ISDN (instead of N-ISDN)

H.322
• adapts H.320 for guaranteed QoS LANs (like ISO-Ethernet)

H.323
• videoconferencing over non-guaranteed LANs

H.324
• Terminal for low bit rate communication (over V.34 Modems)
9. MPEG-1

Motion Picture Expert Group (MPEG)
- ISO/IEC working group(s)
- ISO/IEC JTC1/SC29/WG11
- ISO IS 11172 since 3/93

Starting point: MPEG-1
- Audio/video at about 1.5 Mbit/s
- Based on experiences with JPEG and H.261

Follow-up standards
- MPEG-2
- MPEG-4
- MPEG-7
MPEG - Features

Consideration of other standards:
- JPEG
- H.261

Symmetric and asymmetric compression
Constant data rate, should be < 1856 kbit/s
Target rate about 1.5 Mbit/s
MPEG - Video: Preparation Step

Fixed image format

Color subsampling:
• $Y, C_r, C_b$
• 4:2:0

Resolution:
• Should be at most 768 x 576 pixel
• 8 bit/pixel in each layer (i.e., for $Y, C_r, C_b$)
• 14 pixel aspect ratios
• 8 frame rates

No user defined MCU like JPEG
No progressive mode like JPEG
MPEG - Video: Processing Step

4 types of frames:

I-frames (intra-coded frames):
• Like JPEG
• Real-time decoding demands

P-frames (predictive coded frames):
• Reference to previous I- or P-frames
• Motion vector
  • MPEG does not define how to determine the motion vector
  • difference of similar macroblocks is DCT coded
• DC and AC coefficients are runlength coded

B-frames (bi-directional predictive coded frames):
• Reference to previous and subsequent (I or P) frames
• Interpolation between macro blocks

D-frames (DC-coded frames):
• Only DC-coefficients are DCT coded
• For fast forward and rewind
MPEG - Video Coding

Sequence of I-, P-, and B-frames:

- **I-Frames** (Intracoded)
- **P-Frames** (Predictive Coded)
- **B-Frames** (Bidirectionally Coded)
- **(D-Frames (DC Coded))**

**Sequence:**
- Defined by application
- E.g., I B B P B B P B B I B B P B B P B B P B B ...
- Order of transmission is different: I P B B ...

References
- I-Frames  (Intracoded)
- P-Frames  (Predictive Coded)
- B-Frames  (Bidirectionally Coded)
- (D-Frames (DC Coded))
MPEG - Video: Implications

Random access
- at I-frames
- at P-frames: i.e. decode previous I-frame first
- at B-frame: i.e. decode I and P-frames first

Editing
- **decoded data**
  - loss of quality (encode -> decode -> encode -> ...)
  - application of all video editing functions
- **encoded data (previous to entropy encoding)**
  - preservation of quality
  - transition effects as function in the DCT domain
  - morphing, non-block conform overlay very difficult
- **encoded data**
  - preservation of quality
  - today: too complex, if possible, i.e. need for entropy decoding
MPEG - Audio Coding: Fundamentals

Masking threshold in the frequency domain
- narrowband random noise
- depends on frequency
Masking in Time Domain
• after and before the event
• depends on (to some extent) amplitude
MPEG - Audio Coding

Audio channel:
• Between 32 and 448 kbit/s
• In steps of 16 kbit/s

Definition of 3 layers of quality
• Layer 1: max. 448 Kbit/s
• Layer 2: max. 384 Kbit/s (most often used, also as MUSICAM in DAB)
• Layer 3: max. 320 Kbit/s
MPEG - Audio Coding

Compatible to encoding of CD-DA and DAT:

- **Sampling rates:**
  - 32 kHz, 44.1 kHz, 48 kHz
- **Sampling precision:**
  - 16 bit/sample

Audio channels:

- **Mono (single, 1 channel)**
- **Stereo (2 channels)**
  - dual channel mode (independent, e.g., bilingual)
  - optional: joint stereo (exploits redundancy and irrelevancy)

Application Example: DAB Digital Audio Broadcasting

- uses MPEG layer 2 (compression also known as “MUSICAM” = (Masking pattern adapted Universal Subband Integrated Coding And Multiplexing)

Delays

- max. of 30 ms encoding
- max. of 10 ms decoding
- based on VLSI
MPEG - Audio and Video Data Streams

Audio Data Stream Layers:
1. Frames
2. Audio access units
3. Slots

Video Data Stream Layers:
1. Video sequence layer
2. Group of pictures layer
3. Single picture layer
4. Slice layer
5. Macroblock layer
6. Block layer
10. MPEG-2

Follow-Up MPEG Standards

MPEG-2:
• Higher data rates for high-quality audio/video
• Multiple layers and profiles

MPEG-3
• Initially HDTV
• MPEG-2 scaled up to subsume MPEG-3

MPEG-4:
• Lower data rates for e.g. mobile communication
• Coding and additional functionalities based on image contents

MPEG-7:
• Content description
• Basis for search and retrieval
• See section on databases
MPEG-2

From MPEG-1 to MPEG-2

• **Improvement in quality**
  • from VCR to TV to HDTV

• **No CD-ROM based constraints**
  • higher data rates
    • MPEG-1: about 1.5 Mbit/s
    • MPEG-2: 2-100 Mbit/s

Evolution

• **1994: International Standard**
• Also later known as H.262
• **Prominent role for digital TV in DVB (digital video broadcasting)**
  • commercial MPEG-2 realizations available
MPEG-2 Video

Inclusion of interlaced video format
Increase resolution, more than CCIR 601

Defined as:
• 5 profiles (simple, main,..)
• 4 levels (with increasing resolution,...)

Other additional features
• DCT coefficients may be coded with a non-linear quantization function
MPEG-2 Video: Scaling

Motivation
• analog: continuous decrease in quality if errors occur
• digital: need for tolerance whenever error occur, i.e scaling

Option: Spatial scaling
• reduction of resolution
• approach
  • image sampled with half resolution, then MPEG algorithms applied, output processed with better FEC (base layer)
  • Image decoded, substracted from original, to difference MPEG algorithms applied, output processed with worse FEC (enhanced layer)

Option: Signal to Noise (SNR) scaling
• noise introduced by
  • quantization errors and visible block structures
• approach
  • Base layer: DCT output, more significant bits encoded with better FEC
  • Enhanced layer: DCT output, less significant bits encoded with worse FEC
## MPEG-2 Video Profiles und Levels

<table>
<thead>
<tr>
<th>LEVELS and PROFILES</th>
<th>Simple Profile</th>
<th>Main Profile</th>
<th>SNR Scalable Profile</th>
<th>Spatial Scalable Profile</th>
<th>High Profile</th>
</tr>
</thead>
<tbody>
<tr>
<td>No B-frames</td>
<td>B-frames</td>
<td>B-frames</td>
<td>B-frames</td>
<td>B-frames</td>
<td>B-frames</td>
</tr>
<tr>
<td>4:2:0</td>
<td>4:2:0</td>
<td>4:2:0</td>
<td>4:2:0</td>
<td>4:2:0 or 4:2:2</td>
<td></td>
</tr>
<tr>
<td>Not Scalable</td>
<td>Not Scalable</td>
<td>SNR Scalable</td>
<td>SNR Scalable or Spatial Scalable</td>
<td>SNR Scalable or Spatial Scalable</td>
<td></td>
</tr>
</tbody>
</table>

### High Level
- 1920 pixels/line
- 1152 lines
- ≤ 80 Mbit/s
- ≤ 100 Mbit/s

### High-1440 Level
- 1440 pixels/line
- 1152 lines
- ≤ 60 Mbit/s
- ≤ 80 Mbit/s

### Main Level
- 720 pixels/line
- 576 lines
- ≤ 15 Mbit/s
- ≤ 20 Mbit/s

### Low Level
- 352 pixels/line
- 288 lines
- ≤ 4 Mbit/s
MPEG-2 Audio

Up to

• **5 full bandwidth channels** (surround system)
  • left and right front
  • center (in front)
  • left and right back with \((x \text{ and } y = 0.71)\)

\[
\text{Left for Stereo} = \text{Left}_f + x\text{Center} + y\text{Left}_b \\
\text{Right for Stereo} = \text{Right}_f + x\text{Center} + y\text{Right}_b
\]

• **and 7 multilingual/commentary channels**

Improved quality at or below 64 kbit/s

Compatible to MPEG-1

• all MPEG-1 audio format can be processed by MPEG-2
• only 3 MPEG-2 audio codec will not provide backward compatibility
  (in the range between 256 - 448 Kbit/s)
MPEG-2 System

Steps
1. Audio and video combined to “Packetized Elementary Stream (PES)“
2. PES(es) combined to “Program Stream” or “Transport Stream”

Program stream:
- Error-free environment
- Packets of variable length
- One single stream with one timing reference

Transport stream:
- Designed for “noisy“ (lossy) media channels
- Multiplex of various programs with one or more time bases
- Packets of 188 byte length

Conversion between Program and Transport Streams possible
11. MPEG-4

Goals

MPEG-4 (ISO 14496) originally:
- Targeted at systems with very scarce resources
- To support applications like
  - Mobile communication
  - Videophone and E-mail
- Max. data rates and dimensions (roughly):
  - Between 4800 and 64000 bits/s
  - 176 columns x 144 lines x 10 frames/s

Further demand:
- To provide enhanced functionality
to allow for analysis and manipulation of image contents

MPEG-4: Schedule for Standardization
- 1993: Work started
- 1997: Committee Draft
- 1998: Final Committee Draft
- 1998: Draft International Standard
MPEG-4: Goals (cont.)

Hence: find standardized ways to

- Represent units of aural, visual or audiovisual content
  - "audio/visual objects" or AVOs

- object coding independent of other objects, surroundings and background
- natural and synthetic objects

- Compose these objects together
  - i.e. creation of compound objects that form audiovisual scenes

- Multiplex and synchronize the data associated with AVOs
  - for transportation over network channels providing a QoS (Quality-of-Service)

- Interact with the audiovisual scene generated at the decoder’s site
MPEG-4: Scope

Definition of

• „System Decoder Model“
  • specification for decoder implementations

• Description language
  • binary syntax of an AV object’s bitstream representation
  • scene description information

• Corresponding concepts, tools and algorithms, especially for
  • content-based compression of simple and compound audiovisual objects
  • manipulation of objects
  • transmission of objects
  • random access to objects
  • animation
  • scaling
  • error robustness
MPEG-4: Scope (cont.)

Targeted bit rates for video and audio:

- **VLBV core**
  - „Very Low Bit-rate Video“
  - 5 - 64 Kbit/s
  - image sequences with CIF resolution and up to 15 frames/s

- **Higher-quality video**
  - 64 Kbit/s - 4 Mbit/s
  - quality like digital TV

- **Natural audio coding**
  - 2 - 64 Kbit/s
MPEG-4: Video and Image Encoding

Encoding / decoding of
- Rectangular images and video
  - coding similar to MPEG-1/2
  - motion prediction
  - texture coding
- Images and video of arbitrary shape
  - as done in conventional approach
    - 8x8 DCT or shape-adaptive DCT
  - plus coding of shape and transparency information

Encoder
- Must generate timing information
  - speed of the encoder clock = time base
  - desired decoding times and/or expiration times
    - by using time stamps attached to the stream
- Can specify the minimum buffer resources needed for decoding
MPEG-4: Composition of Scenes

Scene description includes:

- Tree to define hierarchical relationships between objects

  "Rhubarb Rhubarb"

- Objects’ positions in space and time
  - by converting the objects’ local coordinate system into a global coordinate system

- Attribute value selection
  - e.g. pitch of sound, color, texture, animation parameters

Description based on some VRML concepts

- VRML = "Virtual Reality Modelling Language"

Interaction with scenes

- e.g. change viewing point, drag object, start/stop streams, select language
MPEG-4: Example of a Composition
MPEG-4: Scaling

Three approaches:

• **Spatial scalability**
  - decoder displays textures and visual objects at a reduced spatial resolution
    - by decoding only a subset of the total bit stream
  - 32 levels max. for textures and still images
  - 3 levels max. for video sequences

• **Temporal scalability**
  - decoder displays video at a reduced temporal resolution
    - by decoding only a subset of the total bit stream
  - 3 levels max.

• **Quality scalability**
  - bitstream is parsed into a number of bit stream layers of different bit-rates
    - either during transmission or in the decoder
  - subset of the layers still yields a meaningful signal

Spatial and temporal scaling both for

• **Conventional rectangular display and**
• **Objects with arbitrary shape**
MPEG-4: Synthetic Objects

Visual objects:
• Virtual parts of scenes
  • e.g. virtual background
• Animation
  • e.g. animated faces

Audio objects:
• „Text-to-speech“
  • speech generation from given text and prosodic parameters
  • face animation control
• „Score driven synthesis“
  • music generation from a score
  • more general than MIDI
• Special effects
MPEG-4: Layered Networking Architecture

**Display / Recording**

**Media**

**Access Units**

**Adaptation Layer**

**Elementary Streams**

**FlexMux Layer**

**Multiplexed Streams**

**TransMux Layer**

**Network or Local Storage**

**Coding / Decoding**

*e.g. video or audio frames or scene description commands*

**A/V object data**

+ stream type info, sync. info, QoS req.,...

**Flexible Multiplexing**

*e.g. multiple elementary streams with similar QoS requirements*

**Transport Multiplexing**

- only interface specified

- layer itself can be any network, e.g. RTP/UDP/IP, AAL5/ATM
MPEG-4: Layered Networking Architecture (cont.)

DMIF „Delivery Multimedia Integration Framework“
• Allows to establish multiple party sessions
  • interaction with
    • remote interactive peers
    • broadcast systems
    • storage systems
  • establishment of channels with specific QoSs and bandwidths
• Controls
  • FlexMux layer
  • TransMux layer
MPEG-4: Error Handling

Mobile communication:
- Low bit-rate (< 64 Kbps)
- Error-prone

MPEG-4 concepts for error handling:
- **Resynchronization**
  - enables receiver to „tune in“ again
  - based on markers within bitstream
- **Data recovery**
  - enables receiver to reconstruct lost data
  - encode data in an error-resilient manner
- **Error concealment**
  - enables receiver to bridge gaps in data
  - e.g. by repeating parts of old frames
12. Wavelets

Motivation

JPEG / DCT problems:
• DCT not applicable to whole image, but only to small blocks ⇒ block structure becomes visible at high compression ratios
• Scaling as add-on ⇒ additional effort
• DCT function is fixed ⇒ can not be adapted to source data

Improvements by using Wavelets:
• Transformation of the whole image ⇒ overcomes visible block structures and introduces inherent scaling
• Better identification of which data is relevant to human perception ⇒ higher compression ratio
Wavelets: Compression / Decompression

The same overall structure as for DCT-based algorithms

**But:** important differences in the transformation step
Wavelets: Fundamental Idea

Image is transformed into the frequency domain (as in JPEG)

But: based on Wavelet functions instead of cosine functions

cosine:  

Wavelet e.g.:  

Advantage: Wavelets are 0 outside a limited interval

⇒ Wavelet automatically relates only to a part of the image

⇒ Image needs not be splitted into blocks

Use Wavelet family: \( \{2^{-j/2}\Psi(2^{-j}x-k)\}, j,k \in \mathbb{Z}, \Psi \) being a Wavelet
Wavelets: Transformation Steps

"Discrete Wavelet Transformation" (Mallat, 1989)

Split image recursively by using high and low pass filters

read by line

read by column

L Low Pass
H High Pass

c_1

d_1

LH

L

H

d_1^1

d_1^2

d_1^3

L

H

\ldots

lower frequencies

transformed image with reduced size

higher frequencies

http://www.kom.e-technik.tu-darmstadt.de
http://www.ipsi.gmd.de
© Ralf Steinmetz

Wavelets: Transformation Steps

"Discrete Wavelet Transformation" (Mallat, 1989)

Split image recursively by using high and low pass filters

read by line

read by column

L Low Pass
H High Pass

c_1

d_1

LH

L

H

d_1^1

d_1^2

d_1^3

L

H

\ldots

lower frequencies

transformed image with reduced size

higher frequencies

http://www.kom.e-technik.tu-darmstadt.de
http://www.ipsi.gmd.de
© Ralf Steinmetz
Wavelets: Transformation Steps (cont.)

In each step i:

- **Three images** $d_{x}^{i}$ ($x=1,2,3$):
  - containing the high frequency parts of the image
  - representing "details" of the image
  - submitted to Wavelet transformation
    - or thrown away in case of scaling

- **One image** $c^{i}$:
  - containing the lower frequency parts of the image
  - representing the original image with less details / at a lower resolution
  - submitted to step $i+1$

Afterwards:

- **Quantization**
- **Entropy encoding**

as with DCT
Wavelets: DWT compared with DCT

Advantages of DWT over DCT:

• No block artefacts
• Inherent scaling
  • based on the $d_x^i$ for $i=1,2,3,...$
• Lower time complexity for the transformation
  • DCT: $O(n \cdot \log n)$,
  • DWT: $O(n)$ ($n=$number of values to be transformed)
• Higher flexibility: Wavelet function – can be freely chosen
Wavelets: Further Issues

Edge detection reduces high frequencies:
- First extract detected edges
- Then apply wavelets to such a filtered image

Application to video:

\[
\begin{align*}
\text{Image}_{n} & \quad \text{Compute differences} \quad \text{Wavelet compressor} \\
\text{I}_{n-2} & \quad \text{I}_{n-1} \\
\text{I}_{m} & \quad \text{...} \\
\text{I}_{n-1} - \text{I}_{n-2} & \quad \text{I}_{n} - \text{I}_{n-1} \\
\end{align*}
\]
13. Fractal Image Compression

Image Generation

Mandelbrot

- recursive construction of images
- infinite "granularity"
- self-similarities in images
- $Z_i = \text{RealConst.} \times Z_{i-1} + \text{ComplexConst}$
To apply self-similarity: Image Generation

Examples
• (from TUD + Univ. Bochum) for recursive construction of images

Sirpinky triangle
• to produce self-similar structures
• infinite steps applied to different source images lead to same result
• known as Sirpinski-triangle
• "Grenzwert" also known as attractor
To Find Self-Similarities

affine function allows for
• translation
• rotation
• scaling
• brightness adaptation

IFS:
Iterative Function System
• ideally completely self-similar
• example see right

PIFS:
Partitioned Iterative Function System
• real images are
  not completely self-similar
• $W_{\text{img}}$?
Theoretical Basis

Banach’s Fixed Point Theorem:
• Let F be a metrical space
• Let \( W: F \rightarrow F \) be a contractive mapping
  • i.e. there exists an \( s, 0<s<1 \), with \( |W(x)-W(y)| \leq s|x-y| \) for all \( x,y \in F \)
• Then \( W \) has exactly one fixed point \( x_f \)
  • i.e. \( W(x_f) = x_f \)
• \( x_f \) can be computed as \( x_f = \lim_{n \to \infty} W^n(x) \) with \( \text{any} \ x \in F \)

Application to image compression:
• Let \( \text{img} \) be the image to be compressed
• Regard the set of all possible images as a metrical space
  • metric e.g.: maximum difference between the pixels of two pictures
• Goal: construct \( W_{\text{img}} \) such that \( \text{img} \) is the fixed point of \( W_{\text{img}} \)
Fractal Image Compression and Decompression

Compression: Find appropriate $W_{\text{img}} \Rightarrow \text{difficult}$
Decompression: Apply $W_{\text{img}}$ iteratively to any image $\Rightarrow \text{easy}$

- Stop when error falls below some bound
- Error can be calculated by "Collage Theorem"
How to Find $W_{\text{img}}$?

Systematic search based on "Partitioned Iterative Function System (PIFS)"

- **Partition image into "range blocks" $R_i$**
  - 8*8 pixel blocks
  - non-overlapping

- **Consider all "domain blocks" $D_j$ of double size**
  - 16*16 pixel blocks
  - overlapping

- **Find for each $R_i$ the most similar $D_j$**
  - consider rotations ($0^\circ/90^\circ/180^\circ/270^\circ$) and mirroring
  - adapt brightness and contrast of $D_j$ to that of $R_i$
  - translation, rotation, mirroring, brightness adaptation
  - define a (partial) affine function

- **Combine partial functions to $W_{\text{img}}$**
Further Improvements

Quadtree partitioning:

- **Problem:**
  - fixed 8*8 blocks do not reflect image properties

- **Solution:**
  - flexible partition of image into larger or smaller squares
  - driven by image structure

Partitioning into rectangles and triangles
Advantages & Drawbacks

+ High quality at high compression rates
  - At least for images with self-similarities
  - Here: better than JPEG ("cross-over point" at about 1:10 to 1:30)

+ Zooming into image supported
  - detailed view possible, interpolation instead of "pixelization"

+ Scalability
  - decompression steps yield iteratively improving image

- Long compression times
  - asymmetric mechanisms
  - improving search techniques for range & domain block pairs

- blockwise artifacts with Information losses
  - $W_{\text{img}}$ is only approximative

- Not well applicable to images of non-fractal nature
  - E.g. texts, sharp lines & no quality guarantee possible

- Lower quality than JPEG at low compression rates
- Error (Fehlerfortpflanzung)
14. Basic Audio and Speech Coding Schemes

Background
• ITU driven activities

G.711: PCM
• with 64 kbps

G.722 differential PCM (DPCM)
• 48, 56, 64 kbps

G.723
• Multipulse-maximum Likelihood Quatizer (MP-MLQ): 6.3 kbps
• Algebraic Codebook Excitation Linear Prediction (ACELP) 5.3 kbps
• application: speech
Schemes for Speech Coding

G.728: Low Delay Code Excited Linear Prediction (LD-CELP)
- used in audio/video conferencing
- 16 kbps
- one-way end to end delay less than 2 msec (due to CODEC algorithm)
- complex algorithm
  - 16-18 MIPS in floating point required
  - appr. 40 MIPS whole encoding and decoding

AV.253
- still “under consideration” at ITU
- 32 kbps

IS-54
- VSELP
  - good for voice
  - bad for music
- 13 kbps (appr. 8 kbps voice + 5.05 kbps forward error correction FEC)
- driving force: Motorola (similar developments in Japan)
Speech Coding in Mobile Telephone Networks

RPE-LTP (GSM)
- Regular Pulse Excitation - Long-Term Predictor
- used in European GSM: speech
- 13 kbps

GSM Half-Rate Coders
- 5.6 - 6.25 kbps
- quality and characteristics similar to RPE-LPT
Vocoder: e.g. Inmarsat IMBE Coder

Improved Multiband Excitation Coder IMBE
• application: maritime satellite communications
• 4.15 kbps for voice (plus 2.25 kbps for channel coding)

Principle: Vocoder
• (IMBE voiced and unvoiced individually for each frequency band)
15. Conclusion

JPEG:
• Very general format with high compression ratio
• SW and HW for baseline mode available

H.261 / H.263:
• Established standard by telecom world
• Preferable hardware realization

MPEG family of standards:
• Video and audio compression for different data rates
• Asymmetric (focus) and symmetric

Proprietary systems: e.g. Quicktime Product
• Migration to the use of standards

Next steps: wavelets, fractals, models of objects