Multimedia Video Coding & Architectures (5LSE0), Module 09

Interframe Coding,
MPEG-1/2 Standards: Systems & Video

Peter H.N. de With
(p.h.n.de.with@tue.nl)

MPEG System / Standardization

* MPEG = Motion Picture Experts Group
  - joint standardization of ISO and IEC
  - cooperation with CCITT
* Objective (initial, MPEG-1)
  - definition of a generic standard for coding of digital video and associated audio and data for digital storage media (DSM)
* Objective (MPEG-2)
  - a generic standard for various applications, such as DSM, television broadcasting, and communication
* "Generic: wide range of bit rates, variable resolution and quality, flexible for different services

MPEG System / Applications – (1)

* Purpose of standardization
  - improvement of interoperability
  - common technology in SW and HW
  - lower manufacturing costs
  - convergence in AV applications
* Major applications of MPEG coding
  - BSS Broadcasting Satellite Service (home use)
  - CATV Cable Television Distribution
  - CDAD Cable Digital Audio Distribution
  - DAB Digital Audio Broadcasting
MPEG System / Applications – (2)

* Major applications of MPEG coding (continued)
  – DTDB Digital Terrestrial Television Broadcast
  – EC Electronic Cinema
  – ENG Electronic News Gathering (+ satellite SNG)
  – FSS Fixed Satellite Services (to headends)
  – HTT Home Television Theatre
  – IPC Interpersonal Communications (video phone, conf.)
  – ISM Interactive Storage Media
  – MMM Multi-Media Mailing
  – NDB Networked Database Services (via ATM; etc.)
  – SSM Serial Storage Media (digital VTR, etc.)

MPEG System / Milestones – (1)

* MPEG-1
  – Generic coding of moving pictures and associated audio at a throughput rate of up to 1.5 Mbit/s
  – Input is SIF format
  – Related standard H.261 for audiovisual services at px64 kbit/s
  – Final standard described in ISO-11172
  – Applications areas: CD-i, Video CD, video on PCs

* MPEG-2
  – More wide application area than MPEG-1
  – Extensions for interlaced video signals (TV, VCR, ...)
  – Bit rates up to 100 Mbit/s

MPEG System / Milestones – (2)

* MPEG-2 (continued)
  – final standard described in ISO 11383
  – different descriptions for Audio, Systems
  – draft of standard in November 1993, ratification in progress

* MPEG-4
  – advanced extensions of MPEG-2 with respect to block coding
  – New: model-based or object-oriented coding
  – very low bit rates (e.g. 10 kbit/s ... 100 kbit/s), draft in 1998-99

* MPEG-7
  – for archiving of video sequences
  – database management, standard 2001

MPEG System / Application bit rates

* MPEG-1 bit rates
  – Video decoder rates up to 1.856 Mbit/s (telecomm. channel 31 x 64 kbit/s = 1.984 Mbit/s - 0.128 Mbit/s audio
  – CD-i, Video CD have bit rates of 1.2 Mbit/s (audio about 200 kbit/s)

* MPEG-2 bit rates
  – 4-5 Mbit/s PAL TV quality
  – 6-9 Mbit/s CCIR-601 component video quality (approach studio quality)
  – 19 Mbit/s ATV standard for HDTV in the USA
  – 20-40 Mbit/s for HDTV contribution (studio to studio)
MPEG System / Structured data –(1)

* Structure of MPEG formatted data, applied codecs
* System
  – Multiplexing, packetizing of multiple compressed data streams
  – Synchronization and timing of individual data contributions
* Video coding using hybrid compression
  – Motion compensation in the temporal domain
  – DCT coding in the spatial domain (in the image)
* Audio coding
  – Subband coding at 64, 128, or 192 kbit/s
  – Audio subband masking of inaudible components

MPEG System / Structured data –(2)

* MPEG bit stream Structure
* Division in layers
  – Sequence layer
  – Group Of Pictures (GOP)
  – Picture layer
  – Slice layer
  – Macroblock layer
  – Block layer

MPEG System / Block diagram

* MPEG System block diagram
  – timing information of synchronous video and audio
  – timing of multiple MPEG-formatted data streams

MPEG System / Program stream

* MPEG Program stream
  – Combines one or more Packetized Elementary Streams (PES) into one single multiplexed stream.
  – All streams have a common time base.
* Requirements Program Stream
  – Define multiplex for audio and video streams
  – Control buffering of data (over- and underflow)
  – Enable start of decoding after random access
  – Supply timing information
  – Low overhead rate
MPEG System / Decoder program stream

- Typical Decoder Architecture for MPEG progr. stream
- Operate modes as
  - MULTIPLEX-WIDE: the program itself in the pack layer
  - STREAM-SPECIFIC: one of the elementary streams, in the PES packet layer

MPEG System / Timing system level –(1)

- Timing model
  - End-to-End delay from input of encoder to signal output of decoder is constant
  - All samples are presented only once
- Transport Mechanism
  - System Clock Reference (SCR) is sent to decoder
  - SCR specifies intended time at which SCR is entered in the system decoder

- All time units expressed in terms of common System Time Clock (STC)

MPEG System / Timing system level –(2)

- Accuracy
  - Units of 90 kHz or 11 microsec.
  - Repetition intervals no longer than 700 ms (stable control)
- System remarks
  - For transmission systems: PLLs are required to recover master clock for D/A conversions
  - For recording systems: stand-alone video and audio clock for

MPEG System / System Target decoder

- STD is a model!
  - Describes timing and buffering of decoder exactly
  - Parameterized in MPEG (-1 or -2) fields
  - Encoder responsible that STD can decode all
- Physical decoder
  - Must compensate for its differences with the STD

\[ \text{td}(i) = \text{DTS decoding time stamp} \]
\[ \text{tp}(i) = \text{PTS presentation time stamp} \]
MPEG System / Synchronization

* Presentation Time Stamps enable synchronization in program stream
  - PTS apply to the presentation time of compression layer constructs (video packet can start anywhere in the bit stream)
  - E-to-E synchronization occurs when
    1. encoder saves time stamps at capture time
    2. PTS propagate with corresponding data to the decoders
    3. decoder uses the PTS to start presentation
* PTS is sent to the decoder
* Accuracy
  - units of 90 kHz
  - SCR+PCR extensions with resolution of 27 MHz (MPEG-2)

MPEG System / Packs (Program Str.)

* Data units of Pack layer (MPEG-1)
  - SCP Start Code Prefix (unique code in AV stream @ system level)
  - ID Pack Identifier
  - SCR System Clock Reference, 33-bit counter, incr. at 90 kHz
  - MUX RATE multiplied bit rate in units 50 Bytes/s, 22-bit field which can vary each pack (VBR support). Often not req. by receiver

MPEG System / Packets

* Data units of Packet layer (MPEG-1)
  - Packet has only data of one input stream, chronological order
  - Total number of packets per pack is not defined
  - ID Stream ID, 110X XXXX Audio (32 available), 1110 XXXX Video (16 available), 1111 XXXX Data stream (16 available)
  - LEN, Distance to start code next packet in Bytes (16 bits)
  - BUFF TS, System Target Decoder buffer size information
  - STREAM DATA, Data for Video, Audio, data decoder, or specific data, (may include PTS and/or decoding time stamps)

MPEG System / MPEG-2 System – (1)

New system specification: based on MPEG-1, but extended

* New requirements
  - Fixed-length data packets option for error-sensitive media
  - Super MUX: program or channel MUX enabling more programs
  - Lower multiplexing complexity
* Solution: TWO STREAMS for different applications
  1. Program Stream
     - Special AV programs,
     - Common time base for error-free environments (variable packets)
  2. Transport stream
     - Multi-program broadcast, program storage, allows editing
     - One or more distinct time bases
     - Fixed-length packets of 188 Bytes for error-sensitive systems
**MPEG Sys./ MPEG-2 system diagram**

- Distinction between program stream and transport stream
- Packetizing formats optimal for different environments

---

**MPEG System / MPEG-2 Transport Str.**

* Transport Stream (TS) definition
  - One or more programs, each containing one or more elementary streams multiplexed together
  - Fixed-length 188 Byte packets, with usually incl. 4-Byte header (Sync, PID, control-bit parameters, etc.)

* Transport stream extraction possibilities (examples)
  - Select data one program, decode, and presentation of results
  - Extract TS packets of one or more programs from multiple TS and construct new TS
  - Extract 1 program and constitute new program stream (DVD record)
  - Convert program stream to TS, transport it over error-sensitive media, and recover valid program stream (DVD+ DVB broadcast)

* Conclusion: important for broadcasting applications

---

**Intraframe Coding**

* Encode frame-by-frame, disregarding all temporal information
  - Example: Motion-JPEG (AVI compressed)

- Easy bit allocation per frame
- Random access is possible
- Robust to transmission/decompression problems
- But … Moderate compression capabilities
Interframe Coding of temporal differences

- Encode differences between frames (temporal DPCM); consider motion of parts of the frames

Complexity is between intra-frame and 3-D coding
- Can always fall back onto intra-frame coding
- Moderate delay

Principles of Hybrid Coding – (1)

- Basic idea
  - Predict current frame on basis of (coded) previous one
  - Transmit only quantized prediction differences
  - Usually done on 8x8 blocks

Prediction difference:
\[ \Delta x(q,t) = x(q,t) - \bar{x}(q,t-1) \]

Temporal Prediction Gain

- Like normal DPCM, assess the effect of interframe prediction by the prediction gain

\[ G_p = \frac{\text{variance of the original frame}}{\text{variance of the prediction difference}} \]

<table>
<thead>
<tr>
<th>Frame number</th>
<th>( \sigma_x^2 )</th>
<th>( \sigma_{\Delta x}^2 )</th>
<th>( G_p )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1888.2</td>
<td>282.0</td>
<td>6.7</td>
</tr>
<tr>
<td>2</td>
<td>1885.9</td>
<td>225.7</td>
<td>8.4</td>
</tr>
<tr>
<td>3</td>
<td>1873.6</td>
<td>265.7</td>
<td>7.1</td>
</tr>
<tr>
<td>4</td>
<td>1884.6</td>
<td>329.1</td>
<td>5.7</td>
</tr>
<tr>
<td>5</td>
<td>1889.4</td>
<td>342.6</td>
<td>5.5</td>
</tr>
<tr>
<td>6</td>
<td>1901.1</td>
<td>368.9</td>
<td>5.2</td>
</tr>
</tbody>
</table>

around 1.2 - 1.5 bit
Principles of Hybrid Coding – (2)

* Complete temp. DPCM system (1-st ord. predictor, \( h_1 = 1 \))
* Quantized prediction difference: \( \Delta x^*(q, t) = Q[\Delta x(q, t)] \)
* Reconstruction: \( \hat{x}(q, t) = \Delta x^*(q, t) + \hat{x}(q, t - 1) \)

Principles of Hybrid Coding – (3)

* Prediction difference locally still contains a lot of spatial correlation and lots of zeroes: Decorrelate via 8x8 DCT
* Quantization in DCT domain \( \Delta X^*(q, t) = Q[\Delta X(q, t)] \)
* Reconstruction \( \hat{x}(q, t) = \text{DCT}^{-1}(\Delta X^*(q, t)) + \hat{x}(q, t - 1) \)

Example: DCT of differences

Frame difference

DCT of frame difference in bands

- Higher DCT coefficients contain more variance than lower ones
- Taking frame differences removes spatial low-frequency components

Motion Compensation

* Part of the spatial correlation is due to unsuccessful temporal prediction
  - Unpredictable information (occluded regions)
  - Moving spatial information (object movement)
* Find for each block \( x(q, t) \) a corresponding block in encoded frame \( t-1 \): Motion estimation
  - Difference in positions is called motion or displacement vector
Principles of Hybrid Coding – (4)

* Form difference between \( x(q, t) \) and the corresponding block found in encoded frame \( t-1 \): Motion-Comp. Predict.

\[
\Delta x(q, t, d) = x(q, t) - \tilde{x}(q, t - 1, d)
\]

- Motion-compensated prediction difference
- Overhead: 1 motion v. / block

Example – Motion-Compens. Prediction

Motion Comp. / Prediction Gain

<table>
<thead>
<tr>
<th>Frame number</th>
<th>( \sigma_x^2 )</th>
<th>( \sigma_{\Delta x}^2 )</th>
<th>( G_P )</th>
<th>( \sigma_{\Delta x} )</th>
<th>( G_P )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1888.2</td>
<td>282.0</td>
<td>6.7</td>
<td>73.5</td>
<td>25.7</td>
</tr>
<tr>
<td>2</td>
<td>1885.9</td>
<td>225.7</td>
<td>8.4</td>
<td>86.8</td>
<td>21.7</td>
</tr>
<tr>
<td>3</td>
<td>1873.6</td>
<td>265.7</td>
<td>7.1</td>
<td>89.5</td>
<td>20.9</td>
</tr>
<tr>
<td>4</td>
<td>1884.6</td>
<td>329.1</td>
<td>5.7</td>
<td>91.9</td>
<td>20.5</td>
</tr>
<tr>
<td>5</td>
<td>1889.4</td>
<td>342.6</td>
<td>5.5</td>
<td>96.0</td>
<td>19.7</td>
</tr>
<tr>
<td>6</td>
<td>1901.1</td>
<td>368.9</td>
<td>5.2</td>
<td>99.5</td>
<td>19.1</td>
</tr>
</tbody>
</table>

around 0.2 - 0.4 bit
MPEG Video / Principles

- MPEG video exploits both spatial and temporal redundancy

- Temporal redundancy
  - Motion estimation
  - Motion compensation

- Spatial redundancy
  - Block transformation
  - Variable-length coding

MPEG Video / Picture types: I, P, B

- Intraframe (I) coded pictures
  - serve as starting point for a group of pictures
  - modest compression
  - reference picture for other picture types

- Predictive (P) coded pictures
  - coded with reference of past pictures (I or P)
  - reference of remaining B-pictures

- Bi-directional (B) coded pictures
  - coded with reference of past and future pictures (I,P and P)
  - never used as a reference
  - highest compression

- DC (D) intra-coded pictures (not used, spec. modes)
MPEG Video / Group Of Pictures (GOP)

- GOP in normal order at input
  - Step 1: code next P picture
  - Step 2: code intermediate B pics.
- GOP after compression
  - Non-chronological order
  - Unequal bit size

GOP after compression

1st step: predictive
2nd step: bi-directional

MPEG Video / Predict, Bidirectional

- MPEG encoder block diagram
- ME process requires
  - Reorder memories
  - ME unit
  - Two-sided extensions
- VLC at output
  - Rate-control buffer
  - qscale parameter coding

MPEG Video / MPEG-1 sampling – (1)

- MPEG-1 resampling is required
  - target 1.5 Mbit/s too low
  - factor of 5 compression by SIF at 30 Mbit/s
- CCIR-601 (4:2:2 sampling)
  - Y: 720x576 frame, 2:1 interlace, 50 Hz
  - U,V: 360x576 frame, id.
- SIF (.2:1:0“ sampling)
  - Y: 352x288 lines
  - U,V: 176x144 lines
  - 25 frame/s
  - 1:1 progressive
MPEG Video / MPEG-2 sampling

* Alternative sampling in MPEG-2
  - 4:2:2, The CCIR-601 studio standard
  - 4:4:4, for High-quality RGB applications

MPEG Video / Intraframe coding part

* MPEG intraframe coder / decoder block diagram
  - Local encoding
  - Reconstruction for motion compensation
  - Based 8x8 DCT, adapt. Quantization and 2-D VLC
  - Feedback coding with rate buffer control

MPEG Video / Quantization – (1)

MPEG Quantization for inter- / intraframe data

* DC coefficient
  - Human eye very sensitive for DC errors, thus fixed quantizer
  - MPEG-1: DCQ=DC / 8.0 and inverse DC = 8.0 x DCQ
  - MPEG-2: higher DC precision 8-11 bits (n x DCQ)

* AC coefficients
  - Weighting W(u,v) according to perception
MPEG Video / Quantization – (2)

* AC coefficients (cont.)
  - MPEG-1 encoder formula
    \[ F(u,v) = 16 \frac{F(u,v)}{(2 \ q\_scale \ W(u,v))} \]
  - MPEG-1 decoder formula
    \[ F(u,v) = 2 \ (F(u,v) + k) \ q\_scale \ W(u,v) / 16 \]
  - \( k = 0 \) for intrablocks, and
  - \( k = \text{sign}(F(u,v)) \) for non-intra blocks
  - Mismatch control (value closest to zero):
    if \( F(u,v) \) even, then
    \[ F(u,v) = F(u,v) - \text{sign}(F(u,v)) \]

MPEG Video / Quantization-(3) MPEG-2

* MPEG-2 has more precise quantization
* DC coefficients: up to 11 bits precision
* AC coefficients
  - MPEG-2 decoder formula
    \[ F(u,v) = 2 \ (F(u,v) + k) \ q\_scale \ W(w,u,v) / 32 \]
  - \( q\_scale \) is mapped onto larger range than 0...31
  - \( w \) is defined by intra / non-intra and colour sampling
  - \( k = 0 \) for intrablocks, and \( k = \text{sign}(F(u,v)) \) for non-intra blocks
  - special additional mismatch control: \( F(7,7) = F(7,7) \)
    if \( \text{SUM ac}(F(u,v)) \) is odd, and \( F(7,7) = F(7,7) +/- 1 \) if \( F(7,7) \) is even/odd and \( \text{SUM} \) is even.

MPEG Video / Quantization-(4) MPEG-2

MPEG-2 AC coefficients (cont.)

* Extra adaptivity possibilities
  - Quantizer matrix \( W(w,u,v) \) can be reloaded in frame header, giving adaptive weighting on sequence or application
  - \( q\_scale \) Parameter can be modified on macroblock basis, enables smooth regulation of bit rate locally in the image
  - In any case, MPEG-2 different weighting for Y and C

MPEG Video / Quantization – (5)

MPEG-2 AC coefficients

* Larger range of \( q\_scale \) by mapping of transmit code
  - Two characteristics: uniform and non-uniform
  - Non-uniform curve enables different control for low bit-rates
MPEG Video / VLC Scanning – (1)

* Scanning of transform coefficients (MPEG-1/2)
  - Preprocessing step for variable-length coding
  - Scanning functions reorder coefficients to cluster zeros for runlength coding
  - Start with “low-frequency” coefficients
  - Fundamental scanning pattern is diagonal

Example of zigzag scanning

MPEG Video / VLC Scanning – (2)

* MPEG-2 extension of scanning function
  - Picture header extension: indicate the use of alternative scanning pattern (on picture basis)
  - In case of quantizer matrix download: use always diagonal (zigzag) scanning

MPEG Video / Var. Length Coding – (1)

* Variable-length coding of AC coefficients: algorithm of (runlength, amplitude) coding
  - STEP 1: (load coefficient), test of coefficient is zero
  - STEP 2: (update runlength), if zero coefficient, increment zero counter, go to STEP 4
  - STEP 3: (jointly code), if non-zero coefficient, then
    3a. jointly code [runlength, amplitude] in one codeword
    3b. reset runlength counter
  - STEP 4: (do next coefficient), go to STEP 1. If last coefficient, then go to STEP 5.
  - STEP 5: (EOB) Terminate block with EOB-word, ignore runlength value. Codetable is modified Huffman code.

MPEG Video / Var. Length Coding – (2)

* 2-D VLC table of codewords
  1. Unlikely symbols are coded by [esc. code]+[fixed suffix]
  2. Also VLC coding of macroblock address, motion vectors,...

Example of wordlength table
### MPEG Video / Var. Length Coding – (3)

- **2-D. VLC**
- **Table of code words**
  - Special code for 1st coeff.
  - Escape code to avoid long code words
  - Appended sign bit

<table>
<thead>
<tr>
<th>code</th>
<th>runlength</th>
<th>amplitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>EOB</td>
<td></td>
</tr>
<tr>
<td>1s</td>
<td>(note2)</td>
<td>0</td>
</tr>
<tr>
<td>11s</td>
<td>(note3)</td>
<td>0</td>
</tr>
<tr>
<td>011s</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>000s</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>010s</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>0010s</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>0011s</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>011s</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>00010s</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>00011s</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>00101s</td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>00110s</td>
<td></td>
<td>7</td>
</tr>
<tr>
<td>000010s</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>000011s</td>
<td></td>
<td>8</td>
</tr>
<tr>
<td>000101s</td>
<td></td>
<td>9</td>
</tr>
<tr>
<td>000110s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>000111s</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>code</th>
<th>runlength</th>
<th>amplitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>0010 0110s</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>0010 0010s</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0010 0101s</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>0010 1100s</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>0010 1111s</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>0100 111s</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>0101 1s</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>0111 1s</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>00010 0001s</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>00010 0011s</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>00010 1100s</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>00010 1111s</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>00010 1111s</td>
<td>14</td>
<td>1</td>
</tr>
<tr>
<td>00010 1100s</td>
<td>15</td>
<td>1</td>
</tr>
<tr>
<td>0000 0000s</td>
<td>16</td>
<td>1</td>
</tr>
</tbody>
</table>

Note 1: s=sign bit, 0=pos/1=neg.
Note 2: code for dct_coeff_first
Note 3: code for dct_coeff_next

---

### MPEG Video / Var. Length Coding – (4)

- **VLC table for motion vectors**
  - Symmetrical
  - Special code
  - Appended sign bit except 0

<table>
<thead>
<tr>
<th>VL code</th>
<th>motion code</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000 0011 001</td>
<td>-16</td>
</tr>
<tr>
<td>0000 0111 011</td>
<td>-10</td>
</tr>
<tr>
<td>0000 0101 011</td>
<td>-9</td>
</tr>
<tr>
<td>0000 0101 110</td>
<td>-8</td>
</tr>
<tr>
<td>0000 0100 000</td>
<td>-7</td>
</tr>
<tr>
<td>0000 0000 000</td>
<td>-6</td>
</tr>
<tr>
<td>0000 0000 000</td>
<td>-5</td>
</tr>
<tr>
<td>0000 0000 000</td>
<td>-4</td>
</tr>
<tr>
<td>0000 0000 000</td>
<td>-3</td>
</tr>
<tr>
<td>0000 0000 000</td>
<td>-2</td>
</tr>
<tr>
<td>0000 0000 000</td>
<td>-1</td>
</tr>
<tr>
<td>0000 0000 000</td>
<td>0</td>
</tr>
</tbody>
</table>

Note: 1: sign bit, 2:pos/1:neg.
Note: 2: code for dct_coeff_first
Note: 3: code for dct_coeff_next