Image and Video Compression Standards

Touradj Ebrahimi
Touradj.Ebrahimi@epfl.ch
Why a standard?

- Like a language, a standard serves to facilitate seamless communication between men, man-machine and machines.
Two major international organizations have been responsible for providing standards on audio-visual coding:

- International Organization for Standardization (ISO)
- International Telecommunication Union (ITU)
What is MPEG?

- Moving Picture Experts Group
- ISO/IEC JTC1/SC29/WG11
- 150 companies & research institutes
  - USA: AT&T, TI, Motorola, Microsoft, IBM, DEC, GI, ...
  - Asia: Sony, JVC, Mitsubishi, Matsushita, Daewoo, Samsung...
  - Europe: Philips, Thomson, Alcatel, Siemens, Bosch, Deutsche Telekom, BT, CNET...
  - Switzerland: EPFL, ETHZ, Motorola, Studer ...
- MPEG meets 3-5 times per year
Existing and projected Standards in ISO

ISO/JPEG: Still pictures
JPEG, JPEG2000

ISO/MPEG: Motion Picture Experts Group

MPEG-1
CD-ROM

MPEG-2
Digital Television

MPEG-7
Content-based search engines

MPEG-4
multi-media compression
MPEG standards

- MPEG-1 (1992)
  - video on CD-ROM (1.5Mbit/s)
- MPEG-2 (1996)
  - digital TV (5-10Mbit/s)
- MPEG-4 (1999)
  - Multimedia applications (10kbit/s-10Mbit/s)
- MPEG-7 (2001)
  - Multimedia content description interface (video databases)
Relation between MPEG standards

• Data representation pyramid

![Data representation pyramid diagram]

- Pixel-based representation
- Object-based representation
- Semantic-based representation
  - Objects features extraction
  - Objects formation and tracking
  - MPEG-7
  - MPEG-4
  - MPEG-1
  - MPEG-2
What is ITU-T?

- Telecommunication standardization sector of ITU (Formerly CCITT)
- Study group 15 is responsible for video conference and video telephony
- Over 20 companies and research centers around the world actively participate
- Meets 3-5 times every year
- Proposes « Recommendations »
ITU-T Recommendations

• H.120 (1984)
  – Video-conferencing over H1 channels
    (1.5-2 Mb/s)
• H.261 (1988)
  – Video-conferencing over ISDN lines
    (p*64 Kb/s; p1…30)
• H.263 (1996)
  – Video-telephony over POTS and Internet (8-64 Kb/s)
Lessons in Video Compression Basics

- Spatial redundancy reduction (compression)
  - transform coding
- Temporal redundancy reduction (compression)
  - motion estimation/compensation
  - Predictive coding
- Entropy coding
  - from symbols to bits through statistics
- Bitstream syntax
  - specific vocabulary...
- Video formats
Color image coding

Color coordinate conversion and sub-sampling of chrominance components
Video formats

- **ITU-R 601 (CCIR 601)**
  - Europe: 576 x 720 pels, YCrCb, 4:2:2, 25 f/s, int
  - USA/Japan: 480 x 720, YCrCb, 4:2:2, 30 f/s, int
Video formats

- **ITU-R (CCIR 601)**
  - Europe: 576 x 720 pels, YCrCb, 4:2:0, 25 f/s, int
  - USA/Japan: 480 x 720, YCrCb, 4:2:0, 30 f/s, int
Video formats

• Common Intermediate Format - CIF
  – 288 x 360 pels, YUV, 4:1:1, 30 f/s, prog
  – QCIF (1/4 CIF): 144x180 pels, prog
Predictive coding

Quantization

Prediction

Codeword assignment
Texture coding

- Predictive
  - Delta Modulation
  - Adaptive Delta Modulation
  - Motion compensation

- Transform
  - Discrete Cosine Transform
  - Wavelet transform
  - Subband decomposition
  - Pyramidal decomposition
Predictive coding

Linear predictor

\[ X = a.A + b. B + c.C + d.D \]

\[ E = X - X \]
Discrete Cosine Transform (forward)

\[
C(0) = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} f(x)
\]

\[
C(u) = \sqrt{\frac{2}{N}} \sum_{x=0}^{N-1} f(x) \cdot \cos \left( \frac{(2x+1)u\pi}{2N} \right)
\]
Transform coding

Discrete Cosine Transform (inverse)

$$f(x) = \frac{1}{\sqrt{N}} C(0) + \sqrt{\frac{2}{N}} \sum_{u=1}^{N-1} C(u) \cdot \cos \frac{(2x + 1)u\pi}{2N}$$

DCT is widely used in compression thanks to its good energy compaction properties and ease of implementation.
Transform coding

2-D Discrete Cosine Transform
### 2-D Discrete Cosine Transform

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>139</td>
<td>144</td>
<td>149</td>
<td>153</td>
<td>155</td>
<td>155</td>
<td>155</td>
<td>155</td>
</tr>
<tr>
<td>144</td>
<td>151</td>
<td>153</td>
<td>156</td>
<td>159</td>
<td>156</td>
<td>156</td>
<td>156</td>
</tr>
<tr>
<td>150</td>
<td>155</td>
<td>160</td>
<td>163</td>
<td>158</td>
<td>156</td>
<td>156</td>
<td>156</td>
</tr>
<tr>
<td>159</td>
<td>160</td>
<td>162</td>
<td>160</td>
<td>160</td>
<td>159</td>
<td>159</td>
<td>159</td>
</tr>
<tr>
<td>159</td>
<td>160</td>
<td>161</td>
<td>162</td>
<td>162</td>
<td>155</td>
<td>155</td>
<td>155</td>
</tr>
<tr>
<td>161</td>
<td>161</td>
<td>161</td>
<td>161</td>
<td>160</td>
<td>157</td>
<td>157</td>
<td>157</td>
</tr>
<tr>
<td>162</td>
<td>162</td>
<td>161</td>
<td>163</td>
<td>162</td>
<td>157</td>
<td>157</td>
<td>157</td>
</tr>
<tr>
<td>162</td>
<td>162</td>
<td>161</td>
<td>162</td>
<td>163</td>
<td>158</td>
<td>158</td>
<td>158</td>
</tr>
</tbody>
</table>

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>236</td>
<td>-1</td>
<td>-12</td>
<td>-5</td>
<td>2</td>
<td>-2</td>
<td>-3</td>
<td>1</td>
</tr>
<tr>
<td>-23</td>
<td>-17</td>
<td>-6</td>
<td>-3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1</td>
</tr>
<tr>
<td>-11</td>
<td>-9</td>
<td>-2</td>
<td>2</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td>-7</td>
<td>-2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>-3</td>
<td>2</td>
<td>-4</td>
<td>-2</td>
<td>2</td>
<td>1</td>
<td>-1</td>
<td>0</td>
</tr>
</tbody>
</table>
Quantization techniques

- **Scalar**
  - Uniform
  - Non-linear
  - Lloyd-Max
  - Weighted quantization

- **Vector**
  - LBG
  - Adaptive
  - Lattice
Weighted quantization

<table>
<thead>
<tr>
<th>Luminance</th>
<th>Chrominance</th>
</tr>
</thead>
<tbody>
<tr>
<td>16 11 10 16 24 40 51 61</td>
<td>17 18 24 47 99 99 99 99</td>
</tr>
<tr>
<td>12 12 14 19 26 58 60 55</td>
<td>18 21 26 66 99 99 99 99</td>
</tr>
<tr>
<td>14 13 16 24 40 57 69 56</td>
<td>24 26 56 99 99 99 99 99</td>
</tr>
<tr>
<td>14 17 22 29 51 87 80 62</td>
<td>47 66 99 99 99 99 99 99</td>
</tr>
<tr>
<td>18 22 37 56 68 109 103 77</td>
<td>99 99 99 99 99 99 99 99</td>
</tr>
<tr>
<td>24 35 55 64 81 104 113 92</td>
<td>99 99 99 99 99 99 99 99</td>
</tr>
<tr>
<td>49 64 78 87 103 121 120 101</td>
<td>99 99 99 99 99 99 99 99</td>
</tr>
<tr>
<td>72 92 95 98 112 100 103 99</td>
<td>99 99 99 99 99 99 99 99</td>
</tr>
</tbody>
</table>
Coefficient re-ordering

Zig-zag

Zero-tree
Spatial redundancy reduction

- Image is divided into 8*8 blocks (Y, U, V)
- Discrete Cosine Transform (DCT) for each block

JPEG!
JPEG compression (lossy)

- Divide the image into blocks of size 8X8
- Every block is transformed using a 2-D 8X8 DCT
- A weighted scalar quantization is applied to each transformed coefficient in every block
- Quantized DC values are coded by DPCM from macroblock to macroblock
JPEG compression (lossy)

- Quantized AC coefficients are scanned in a zig-zag pattern
- Quantized AC coefficients are encoded using a Huffman and runlength coding
- Arithmetic coding can be used instead of Huffman codes for entropy coding of quantized coefficients
JPEG compression (lossy)

- DCT
- Quantization
  - Quantization table
- AC Coefficients zig-zag ordering
- DC coefficients difference prediction
- Huffman (Arithmetic) coding
JPEG compression (lossless)

- Predictive delta modulation
- Huffman coding of prediction error
- Compression ratios of around 2:1 for most natural images
- Applied to gray level and color tone still images
- Three-sample prediction neighborhood
- 8 different modes available
- Very sensitive to errors
JPEG compression (lossless)
## JPEG compression (lossless)

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

```
     C  B
    A X
```

$X = \frac{(A + B)}{2}$
Temporal redundancy reduction

- First frame: **INTRA** coding (spatial)
- How to predict next frames (**INTER** coding)?
  - motion estimation/compensation

frame 1
original 704*576 + color: 600KB

frame 2
original 704*576 + color: 600KB

1/25s
Motion estimation

• ideally: motion info for each pixel
  – too expensive
• semantically: motion info for each *homogeneous* region or object
  – second generation coding techniques
• simplified: motion info for each 16*16 macroblock
Motion model

- **Affine motion:**
  - translation, rotation, scale
  - 6 parameters in 2D
  - complex

- **Simplified translational model:**
  - motion vectors (2 parameters)
Motion estimation

• **optimal motion vector?**
  – investigate all positions within a search window
  – keep the one with minimum Mean Square Error
  – motion vector = corresponding translation

Not standardized!
Motion estimation

• Backward prediction
  – Predict where the pixels in a current frame were in a past frame

• Forward prediction
  – Predict where the pixels in a current frame will go to in a future frame
Backward prediction

- No holes or overlap is created in the predicted image
- All standards use this approach
- Introduces less delays

Previous frame

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
</tr>
</tbody>
</table>

Current (predicted) frame

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
</tr>
</tbody>
</table>
Forward prediction

- Holes or overlap regions are created in the predicted image

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
</tr>
</tbody>
</table>

Current frame

<p>| | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
</tr>
</tbody>
</table>

Future (predicted) frame
H.261 block diagram

- Past reference frame buffer
- Motion compensation
- Motion estimation
- Inverse block difference
- Motion vector coding
- Block difference
- DCT
- Quantization
- Entropy coding
- Inverse quantization
- Inverse DCT
- Intra-coded frames
- Predictive-coded frames
- Coded motion vectors
ITU-T recommendation H.261 (p X 64)

- Only the decoder is standardized along with the bitstream syntax
- The video data is split into Intra and inter frames
- Intra frames are coded in a similar manner to JPEG (variant)
- Inter frames are predicted from previously decoded frames
- Prediction is based on block based motion compensation (16 X 16 blocks)
ITU-T recommendation H.261 (p X 64)

⇒ A loop filter is used in the prediction loop to reduce high frequency noise
⇒ One macroblock per frame is in turn forced to intra to avoid error propagation
⇒ Motion vectors are obtained at integer pixel accuracy and sent to the decoder using DPCM at macroblock basis
⇒ Feed-back rate control in most encoders
⇒ Covers bitrates of multiple of 64 Kb/s (1 to 30)
⇒ Handles video of size CIF and QCIF
⇒ Average frame rate 5-15 f/s
I, P frames

- **INTRA I-frames**
- **Predicted P-frames:**
  - backward predicted from previous anchor picture (I or P)
A sequence is composed of

- Frames structured as
  - ...IPPPP...I...
- Frame
  - Group of Blocks (Basic resync point)
    - Macroblock (motion)
      - Block (DCT)

Diagram:
- group of blocks
- macroblock
- DCT block
Entropy coding

- **Differential coding**
  - prediction from block to block
  - DC coefficient, motion vectors
- **Variable length codes VLC tables (Huffman):**
  - the most likely the parameter, the shortest the symbol
  - motion vectors, DCT coefficients
Rate control

increase/decrease DCT quantization

images → Encoder → Buffer → bitstream

Possible to influence:
DCT quantizers
Mode decision scheme

Not standardized!
MPEG-1 Block diagram
MPEG-1 video compression technique

- Only the decoder is standardized along with the bitstream syntax
- The video sequence is split into intra, predicted and interpolated frames
- The video sequence is divided into group of pictures starting with intra frames
- Motion vectors are obtained at half pel accuracy and sent to the decoder using DPCM at macroblock basis
- Handles CCIR 601 and CIF formats
- Covers bitrates of about up to 1.5 Mb/s
Motion estimation refinement

- half-pel refinement
  - motion estimation from previous reconstructed and interpolated frame
Mode decision

- Macroblock MSE < tsh 1:
  - transmit motion only
- tsh1 < Macroblock MSE < tsh 2:
  - transmit motion + DCT on DFD
  - Displaced Frame Difference: motion compensated error image (predicted-original)
  - Adapted DCT quantization (around 0)
- Macroblock MSE > tsh2:
  - INTRA macroblock

Not standardized!
**I, P, B frames**

- **INTRA I-frames:**
  - random access
  - error robustness

- **Predicted P-frames:**
  - backward predicted from previous anchor picture (I or P)

- **Bidirectionnally predicted B-frames:**
  - forward/backward predicted from previous anchor picture (I or P)
MPEG-2 video compression technique

- Only the decoder is standardized along with the bitstream syntax
- Several modes are considered in order to take into account interlaced frames (field based modes)
- Generic structure in order to cope with several bitrates and picture formats
- Spatial, frequency and temporal scalability
Interlaced/progressive coding in MPEG-2

Frame DCT

Field DCT
Scalable coding in MPEG-2

Spatial or temporal down-scaling

MPEG-2 enhancement layer encoding

Spatial or temporal up-scaling

MPEG-2 base-layer encoding

Enhancement layer bitstream

Base-layer bitstream
Profiles and levels

- Profiles are a set of pre-defined tools and their configurations.
- Profiles are divided into Levels each defining upper bound limits for coding parameters.
Profiles in MPEG-2

- **Simple**
  - Simplest profile similar to Main profile, except for the lack of B frames
- **Main**
  - Non scalable coding providing interlaced coding tools, random access, B mode.
- **SNR Scalable**
  - Similar to Main plus a 2 layer SNR scalability
- **Spatial Scalable**
  - Similar to SNR scalable profile plus a 2 layer spatial scalability
- **High**
  - Similar to Spatial Scalable profile with provisions for 3 layers in spatial and SNR scalability and 4:2:2 coding
- **4:2:2**
  - Similar to Main profile with 4:2:2 coding
Levels in MPEG-2

- **Low**
  - 352x288 pels, 30 f/s, 4Mb/s
- **Main**
  - 720x576 pels, 30 f/s, 15Mb/s
- **High1440**
  - 1440x1152 pels, 60f/s, 60Mb/s
- **High**
  - 1920x1152 pels, 60 f/s, 80Mb/s
Complete MPEG-2 transmission scheme

**Video Sequence** → **MPEG-2 coder** → **Channel** → **Error Concealment** → **MPEG-2 decoder** → **Video Sequence**

Signals which data packets (184 bytes) are erroneous

Conceals lost areas in images
Error detection and resynchronization

- slice = basic resynchronization point (16 lines)

- channel error detection = skip detected erroneous data packets in the bitstream
Error concealment

- Temporal concealment
  - Simple replacement
  - Interpolated motion vector
  - Motion compensated concealment

- Spatial concealment
  - Spatial interpolation from boundary pixels
Temporal propagation of imperfect concealment

imperfect concealment of an I or P frame used as temporal reference

Motion compensated temporal prediction

predicted imperfect area

internally concealed area

low quality for predicted frames
ITU-T recommendation H.263

- The video sequence is split into intra, predicted and PB frames
- Advanced motion estimation is available (8X8 blocks)
- Motion vectors are obtained at half pixel accuracy and sent to the decoder using DPCM at macroblock basis
- Extended motion vectors are available
- Arithmetic coding can be used
- Covers bitrates less than 64 Kb/s