Source Coding and Compression

Heiko Schwarz

Contact:
Dr.-Ing. Heiko Schwarz
heiko.schwarz@hhi.fraunhofer.de
Part II:
Application in Image and Video Coding
Outline

Part I: Source Coding Fundamentals
- Probability, Random Variables and Random Processes
- Lossless Source Coding
- Rate-Distortion Theory
- Quantization
- Predictive Coding
- Transform Coding

Part II: Application in Image and Video Coding
- Still Image Coding / Intra-Picture Coding
  - Representation of Images and Video
  - JPEG
  - Intra-Picture Coding in MPEG-2 Video
  - Intra-Picture Coding in H.263 and MPEG-2 Visual
  - Intra-Picture Coding in H.264/AVC
  - Intra-Picture Coding in H.265/HEVC
- Hybrid Video Coding (From MPEG-2 Video to H.265/HEVC)
Still Image Coding / Intra-Picture Coding – Overview

Still Image Coding

- Exchange, transmission and storage of images
- Used in virtually all digital cameras and picture editing applications
- JPEG: Most widely used image compression standard (based on DCT)
- JPEG-2000: Wavelet-based image compression (not discussed in lecture)
- JPEG-XR: Several improvements over JPEG (not discussed in lecture)

Intra-Picture Coding for Video

- Intra-picture coding: Some pictures of a video sequence need to be coded without referring to other picture inside the video sequence
- First picture of a video sequence has to be intra-picture coded
- Intra pictures in regular intervals (e.g., 1s) are required for enabling random access
- Typically, regularly inserted intra pictures consume large amount of bit rate
- H.262 | MPEG-2 Video / H.263 / MPEG-4 Visual: Conceptually similar to JPEG
- H.264 | MPEG-4 AVC: Additional coding tools yielding improved coding efficiency
- H.265 | MPEG-H HEVC: Increased flexibility and improved coding efficiency
Representation of Images and Video
Digital Images and Video

Image
- 2-d function $s(x, y)$ relating light intensity $s$ to spatial coordinates $(x, y)$

Digital image
- Representation of a continuous image at discrete coordinates $[x, y]$  
- Amplitudes $s[x, y]$ have finite alphabet, typically determined by the used bit depth
- Digital gray-level image of size $M \times N$ can be represented using a matrix notation

$$
\begin{bmatrix}
  s[0, 0] & s[1, 0] & \cdots & s[M - 1, 0] \\
  s[0, 1] & s[1, 1] & \cdots & s[M - 1, 1] \\
  \vdots & \vdots & \ddots & \vdots \\
  s[0, N - 1] & s[1, N - 1] & \cdots & s[M - 1, N - 1]
\end{bmatrix}
$$
- Typically characterized by image size $M \times N$ and bit depth
- Color images are typically composed of 3 sample arrays (for different color components)

Digital video
- Sequence of digital images captured at successive time instances
- Typically characterized by frame rate (in addition to image size and bit depth)
Spatial Resolution

Number of samples ($M \times N$) for discrete matrix representation

- 320 $\times$ 240 samples
- 160 $\times$ 120 samples
- 80 $\times$ 60 samples
- 40 $\times$ 30 samples
Gray-Level Resolution / Bit Depth

Number of gray levels for image representation (typically determined by bit depth)

- 256 gray levels (8 bit)
- 64 gray levels (6 bit)
- 16 gray levels (4 bit)
- 4 gray levels (2 bit)
Still Image Coding

Representation of Images and Video

Representation of Color Images

- Color components
  - Require at least 3 color components (trichromatic vision)
  - RGB typically used as reference color space
  - Need to specify color of RGB primaries in CIE XYZ reference space

BT.709 RGB parameters

<table>
<thead>
<tr>
<th>primary</th>
<th>$x$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>red</td>
<td>0.6400</td>
<td>0.3300</td>
</tr>
<tr>
<td>green</td>
<td>0.3000</td>
<td>0.6000</td>
</tr>
<tr>
<td>blue</td>
<td>0.1500</td>
<td>0.0600</td>
</tr>
<tr>
<td>white D65</td>
<td>0.3127</td>
<td>0.3290</td>
</tr>
</tbody>
</table>
YCbCr Color Space

Definition of YCbCr color space

- More correct name is Y’CbCr, since Y’ is a gamma-adjusted luminance component
- Not an absolute color space, but a different representation for RGB data
- Transform of gamma-adjusted and normalized RGB components \( r', g', \) and \( b' \) with a range of 0..1
- Typically used transform for 8-bit components \( Y, Cb \) and \( Cr \)

\[
Y' = \text{Round}(219 \cdot y' + 16) \\
Cb = \text{Round}(224 \cdot pb' + 128) \\
Cr = \text{Round}(224 \cdot pr' + 128)
\]

with

\[
y' = K_R \cdot r' + (1 - K_R - K_B) \cdot g' + K_B \cdot b' \\
pb' = 0.5 \cdot (b' - y')/(1 - K_B) \\
pr' = 0.5 \cdot (r' - y')/(1 - K_R)
\]

- The coefficients \( K_R \) and \( K_B \) are specified by application standards
- ITU-R Rec. BT.709 specifies \( K_R = 0.2126 \) and \( K_B = 0.0722 \)
- \( Y' \) is called luma component, \( Cb \) and \( Cr \) are called chroma components
Advantages of YCbCr Color Representation

Properties of the YCbCr color representation

- Similar color decorrelation as in human visual system:
  - \( Y' \) component is related to brightness
  - \( Cb \) component represents a yellow-blue difference signal
  - \( Cr \) component represents a red-green difference signal
- Coding errors are introduced in perceptual meaningful way
- Effectiveness for coding of images and video experimentally verified
Subsampling of Chroma Components

Visual importance of luma and chroma components

- YCbCr color space roughly approximates the color decorrelation in the human visual system
- Human visual system is more sensitive to details in luma (brightness) channel than to details in chroma channels
- Chroma channels can be downsampled for saving bit rate
- Chroma downsampling by a factor of 2 in horizontal or both spatial directions
- Location of chroma samples relative to luma samples has to be specified
Demonstration of Luma/Chroma Perception

Compare luma and chroma perception
- Selective low-pass filtering for luma or chroma components
- Use low-pass filter \((1,4,6,4,1)/16\)

Order of presentation
1. Original luma and chroma components
2. Low-pass filtered luma component but original chroma components
3. Original luma and chroma components (repeated)
4. Original luma component but low-pass filtered chroma components
Demonstration: Original Picture
Demonstration: Filtered Luma Component
Demonstration: Original Picture (Repeated)
Demonstration: Filtered Chroma Components
Chroma Sampling Formats for Image and Video Coding

- **RGB**
- **YCbCr 4:4:4**
- **YCbCr 4:2:2**
- **YCbCr 4:2:0**

*most common format*
JPEG
**Joint Photographic Experts Group (JPEG)**
- Standard is named after the group which created it
- Joint committee between ITU-T (formerly CCITT) and ISO/IEC JTC 1

**Standard “Digital Compression and Coding of Continuous-Tone Still Images”**
- Officially ITU-T Rec. T.81 and ISO/IEC 10918-1
- Commonly referred to as JPEG
- Specifies compression for gray-level and color images

**Applications of JPEG**
- Storage format used in virtually all digital cameras (except for “raw” sensor data)
- Most pictures in the Internet are JPEG pictures
- Motion-JPEG is de facto standard for digital video editing
The Scope of Image and Video Coding Standardization

What is standardized?

- **Data format** including constraints for the data
- **Decoding result** to be produced by a conforming decoder
  ⇒ Provides interoperability between different devices
  ⇒ Permits optimization beyond the obvious
  ⇒ Permits complexity reduction for implementability
  ⇒ Provides no guarantee of quality

![Image Processing Flowchart]
JPEG: Basic Codec Structure

Encoder and decoder structure for each color component

```
image in → block-wise 8 × 8 DCT → scaling / quantization → entropy coding → bitstream out

image out ← block-wise inverse 8 × 8 DCT ← inverse scaling ← entropy decoding ← bitstream in

table specifications (transmitted as side information)
```
JPEG: Partitioning of Color Components into $8 \times 8$ Blocks

Color components are coded independently of each other
- Color components are partitioned into $8 \times 8$ blocks (padding at borders)
- The $8 \times 8$ blocks are coded using transform coding
Two-dimensional Transform for Image Compression

Separable and symmetric 2-d orthogonal block transform

- 2-d linear transform: Each input block is represented as a linear combination of 2-d basis functions (or basis blocks)
- Separable and symmetric 2-d orthogonal block transform: Transform of an $N \times N$ block $s$ can be written as

$$
    u = A \cdot s \cdot A^T
$$

(588)

where $A$ is the $N \times N$ transform matrix and $u$ is the $N \times N$ block of transform coefficients

- Inverse of separable and symmetric 2-d orthogonal block transform is given by

$$
    s' = A^T \cdot u' \cdot A
$$

(589)

- Great practical importance:
  Separable transform requires 2 matrix multiplications of size $N \times N$ instead of one multiplication of a vector of size $1 \times N^2$ with a matrix of size $N^2 \times N^2$

$\implies$ Complexity reduction from $O(N^4)$ to $O(N^3)$
2-d Transform used in JPEG

2-d block transform in JPEG

- Separable DCT of type II
- $8 \times 8$ transform matrix $A$ consisting of elements $a_{ik}$, with $i, k = 0, \ldots, 7$, given by

$$a_{ik} = \alpha_i \cos \frac{\pi (2k + 1) i}{16}$$  \hspace{1cm} (590)

with

$$\alpha_i = \frac{1}{4} \begin{cases} 
1 & : \quad i = 0 \\
\sqrt{2} & : \quad i > 0
\end{cases}$$  \hspace{1cm} (591)

- Transform can be implemented using a fast butterfly algorithm

Transform specification in JPEG

- Ideal forward and backward transform are given in informative clause
- Specification contains normative accuracy requirements
2-d DCT Example – Step 1: Vertical Transform

Example for a $16 \times 16$ DCT

- **Step 1**: Column-wise DCT on image block yielding intermediate block of transform coefficients
- Notice the energy concentration in the first row (DC coefficients)
2-d DCT Example – Step 2: Horizontal Transform

Example for a $16 \times 16$ DCT

- **Step 2**: Row-wise DCT on intermediate block of transform coefficients yielding the final block of DCT coefficients
- Notice the energy concentration in the DC coefficient (top-left)
Quantization in JPEG

JPEG specifies uniform reconstruction quantizers for the transform coefficients.

- Inverse quantization (scaling) in decoder is specified by

  \[ t'_{ik} = \Delta_{ik} \cdot q_{ik} \] (592)

  where
  - \( q_{ik} \): Quantization index for coefficient at location \((i, k)\) inside block
  - \( \Delta_{ik} \): Quantization step size for coefficient at location \((i, k)\)
  - \( t'_{i, k} \): Reconstructed transform coefficient at location \((i, k)\)

- No normative encoding procedure, but informative quantization rule

  \[ q_{ik} = \text{round} \left( \frac{t_{ik}}{\Delta_{ik}} \right) \quad (t_{i,k}: \text{original transform coefficient}) \] (593)

- Standard specifies accuracy requirements for combination of DCT and quantization
  \[ \implies \text{Leaves freedom for encoder designers} \]

- Separate quantization step sizes can be selected for each coefficient location \((i, k)\) and color component
  \[ \implies \text{Quantization tables have to be transmitted as side information (no defaults)} \]
  \[ \implies \text{Additional freedom for encoder optimization} \]
Quantization Tables in JPEG

Quantization tables

- Determine rate and distortion (among other parameters)
- Need to be transmitted (no default tables in JPEG)
- Example tables for YCbCr format are specified in Annex K of standard (empirically derived based on psychovisual threshold experiments)

<table>
<thead>
<tr>
<th>luma blocks</th>
<th>chroma blocks</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>
Entropy Coding in JPEG Baseline

Entropy coding of transform coefficient levels (quantization indices)
- Different concepts for DC and AC levels
- DC levels: Differential coding using codeword tables
- AC levels: Run-level coding of scanned coefficients

Coding of DC transform coefficient levels
- DC level is predicted by previous DC level as predictor
- Difference to predictor is coded using VLC and FLC
- Category $C$ is coded using VLC
  - $\Rightarrow$ Specifies range of values
  - $\Rightarrow$ Specifies number of following bits (for FLC)
- FLC specifies actual value of $DIFF$ inside category $C$
  - $\Rightarrow$ If $DIFF > 0$, low-order bits of $DIFF$
  - $\Rightarrow$ If $DIFF < 0$, low-order bits of $DIFF - 1$
- VLC table for category needs to be transmitted
  - $\Rightarrow$ Increases side information (no default table)
  - $\Rightarrow$ Allows adaptation to actual statistics

$DIFF = DC(N) - DC(N-1)$
Example VLC Table for Coding DC Difference Category

Example VLC table for coding category $C$

- Range of $DIFF$ values is specified in standard
- Codeword assignment has to be transmitted
- Example shows recommended table for luma DC (Annex K of JPEG)

<table>
<thead>
<tr>
<th>Category $C$</th>
<th>Range of $DIFF$ value</th>
<th>Example codeword</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>00</td>
</tr>
<tr>
<td>1</td>
<td>-1, 1</td>
<td>010</td>
</tr>
<tr>
<td>2</td>
<td>-3, -2, 2, 3</td>
<td>011</td>
</tr>
<tr>
<td>3</td>
<td>-7..-4, 4..7</td>
<td>100</td>
</tr>
<tr>
<td>4</td>
<td>-15..-8, 8..15</td>
<td>101</td>
</tr>
<tr>
<td>5</td>
<td>-31..-16, 16..31</td>
<td>110</td>
</tr>
<tr>
<td>6</td>
<td>-63..-32, 32..63</td>
<td>1110</td>
</tr>
<tr>
<td>7</td>
<td>-127..-64, 64..127</td>
<td>11110</td>
</tr>
<tr>
<td>8</td>
<td>-255..-128, 128..255</td>
<td>111110</td>
</tr>
<tr>
<td>9</td>
<td>-511..-256, 256..511</td>
<td>1111110</td>
</tr>
<tr>
<td>10</td>
<td>-1023..-512, 512..1023</td>
<td>11111110</td>
</tr>
<tr>
<td>11</td>
<td>-2047..-1024, 1024..2047</td>
<td>1111111110</td>
</tr>
</tbody>
</table>
Entropic Coding of AC Transform Coefficient Levels

Representation of AC levels

- Convert into sequence using a zig-zag scan
- AC coefficients are likely to be quantized to zero (in particular those at high-frequency locations)
- Successive runs of zeros are represented using a run (number of consecutive levels equal to zero)
- Non-zero AC levels are represented by a category and a value inside the category (same as for DC levels)

Coding of AC levels

- AC levels are coded using a combination of VLC and FLC
- Variable-length code table is used for coding events \{run,category\}
- VLC table includes a special symbol (EOB) for signaling the end-of-block (all remaining AC levels are equal to zero)
- Fixed-length code is used for coding the exact value inside a category (number of bits is given by category) – same as for DC difference levels
- VLC table has to be transmitted (no default table)
Example VLC Table for Run-Category Coding of AC Levels

Example for VLC table

- Standard defines ranges for categories (same as for DC, but no categories 0 and 11)
- Codeword assignment has to be transmitted
- Example shows first entries of recommended table for luma AC (Annex K of JPEG)

<table>
<thead>
<tr>
<th>run/category</th>
<th>codeword</th>
<th>run/category</th>
<th>codeword</th>
</tr>
</thead>
<tbody>
<tr>
<td>EOB</td>
<td>1010</td>
<td>2/1</td>
<td>11100</td>
</tr>
<tr>
<td>0/1</td>
<td>00</td>
<td>2/2</td>
<td>11111001</td>
</tr>
<tr>
<td>0/2</td>
<td>01</td>
<td>2/3</td>
<td>111110111</td>
</tr>
<tr>
<td>0/3</td>
<td>100</td>
<td>2/4</td>
<td>11111110100</td>
</tr>
<tr>
<td>0/4</td>
<td>1011</td>
<td>2/5</td>
<td>111111110001001</td>
</tr>
<tr>
<td>0/5</td>
<td>11010</td>
<td>2/6</td>
<td>111111110001010</td>
</tr>
<tr>
<td>0/6</td>
<td>1111000</td>
<td>2/7</td>
<td>111111110001011</td>
</tr>
<tr>
<td>0/7</td>
<td>11111000</td>
<td>2/8</td>
<td>111111110001100</td>
</tr>
<tr>
<td>0/8</td>
<td>1111110110</td>
<td>2/9</td>
<td>111111110001101</td>
</tr>
<tr>
<td>0/9</td>
<td>111111110000010</td>
<td>2/10</td>
<td>111111110001110</td>
</tr>
<tr>
<td>0/10</td>
<td>1111111110000111</td>
<td>3/1</td>
<td>111010</td>
</tr>
<tr>
<td>1/1</td>
<td>1100</td>
<td>3/2</td>
<td>111110111</td>
</tr>
<tr>
<td>1/2</td>
<td>11011</td>
<td>3/3</td>
<td>111111110101</td>
</tr>
<tr>
<td>1/3</td>
<td>1111001</td>
<td>3/4</td>
<td>111111110001111</td>
</tr>
<tr>
<td>1/4</td>
<td>111110110</td>
<td>3/5</td>
<td>111111110010000</td>
</tr>
<tr>
<td>1/5</td>
<td>1111110110</td>
<td>3/6</td>
<td>111111110010001</td>
</tr>
<tr>
<td>1/6</td>
<td>111111110000100</td>
<td>3/7</td>
<td>111111110010010</td>
</tr>
<tr>
<td>1/7</td>
<td>111111110000101</td>
<td>3/8</td>
<td>1111111100100111</td>
</tr>
<tr>
<td>1/8</td>
<td>111111110000110</td>
<td>3/9</td>
<td>111111110010100</td>
</tr>
<tr>
<td>1/9</td>
<td>1111111100000111</td>
<td>3/10</td>
<td>111111110010101</td>
</tr>
<tr>
<td>1/10</td>
<td>1111111100010000</td>
<td>3/11</td>
<td>111111110010101</td>
</tr>
</tbody>
</table>

...
Example: JPEG Transform Coefficient Level Coding

Example for an 8×8 luma block:
- Last DC level: \( DC(N - 1) = 178 \)
- Use recommended luma tables

Coding of DC transform coefficient level
- Prediction difference: \( DIFF = 185 - 178 = 7 \)
- Category \( C = 3 \): Codeword “100”
- Fixed-length code (lowest 3 bits of “7”): “111”
- Final bit representation (6 bit): “100111”

Coding of AC transform coefficient levels
- Zig-zag scanning and conversion into (run,level) pairs yields
  \[(0, 3) \ (0, 1) \ (2, 1) \ (1, -1) \ (6, -3) \ (0, -1) \ (0, -2) \ (0, -1) \ (EOB)\]
- Representation as (run,category) [FLC bits] sequence
  \[(0, 2)[11] \ (0, 1)[1] \ (2, 1)[1] \ (1, 1)[0] \ (6, 2)[00] \ (0, 1)[0] \ (0, 2)[01] \ (0, 1)[0] \ (EOB)\]
- Bit sequence: VLC bits [FLC bits] (in total: 46 bits for 63 AC levels)
JPEG Compression Example – Original (YCbCr 4:2:0, 12 bpp)
JPEG Compression Example – 1:10 Compression (1.2 bpp)
JPEG Compression Example – 1:25 Compression (0.48 bpp)
JPEG Compression Example – 1:50 Compression (0.24 bpp)
JPEG Compression Example – 1:200 Compression (0.06 bpp)
Summary of JPEG

JPEG Baseline
- Minimum of capabilities (required for all DCT-based JPEG codecs)
- Source image: 1-4 color components with 8-bit per sample
- Sequential processing of $8 \times 8$ blocks
- Transform: Separable $8 \times 8$ discrete cosine transform (DCT) of type II
- Quantizer: Scalar uniform reconstruction quantizer (using quantization table)
- DC coding: Prediction and combination of VLC and FLC
- AC coding: Zig-zag scan and run-level coding (combination of VLC & FLC)
- VLC coding: 2 DC tables (category) & 2 AC tables (run/category)

Extended JPEG features
- Extended bit depth
- Adaptive binary arithmetic coding
- Progressive and hierarchical coding
- Lossless coding mode
- Extended file formats (e.g., EXIF)
Intra-Picture Coding in H.262 | MPEG-2 Video
H.262 | MPEG-2 Video

- Video coding standard jointly developed by ITU-T and ISO/IEC JTC 1
- Standard was finalized in 1994
- Still widely used in digital television and the DVD-Video optical disc format
- Three pictures types: I (intra), P (predictive) and B (bi-directional)
- Includes tools for interlaced video
- Most important conformance point: Main Profile
  - Color format: YCbCr 4:2:0
  - Bit depth: 8 bit per sample

Intra-picture coding in H.262 | MPEG-2 Video

- Conceptually very similar to JPEG Baseline
- Details and actual syntax are different
- Fixed variable-length entropy coding tables
Macroblocks and Blocks in H.262 | MPEG-2 Video

Picture partitioning into macroblocks
- Picture is partitioned into fixed-size **macroblocks**, which consist of $16 \times 16$ luma samples and the corresponding areas in the chroma components
- In 4:2:0 chroma sampling format, a macroblock corresponds to
  - one $16 \times 16$ luma block
  - two $8 \times 8$ chroma blocks

Coding of macroblocks
- Different **coding modes**, also referred to as **macroblock modes**
  - Intra picture: 2 coding modes
    - Intra
    - Intra+Q (quantizer change)
  - Intra mode: Transform coding for all six $8 \times 8$ blocks of a macroblock (4 luma and 2 chroma blocks)
Coding of $8 \times 8$ Intra Blocks in H.262 | MPEG-2 Video

Transform coding of $8 \times 8$ blocks
- Orthogonal block transform + scalar quantization + entropy coding
- Very similar to JPEG (but some differences in details)

Orthogonal block transform
- 2-d discrete coding transform (DCT) – same as in JPEG

Scalar quantization
- Quantization step size is specified by a quantization matrix and a quantization parameter (scaling for all coefficients, can be modified on macroblock basis)

Entropy coding of transform coefficient levels
- DC coefficient: Differential coding similar to JPEG
- AC coefficients: Zig-zag scan and run-level coding with EOB symbol
Quantization in H.262 | MPEG-2 Video

Standard specifies only construction of transform coefficients from levels

- Inverse quantization of intra DC coefficients (intra_dc_precision in range 8..11)
  \[ t'_{00} = q_{00} \cdot 2^{3 - \text{intra\_dc\_precision}} \]  
  \[ (594) \]

- Inverse quantization of AC coefficients in intra blocks
  \[ t'_{ik} = \text{sgn}(q_{ik}) \cdot \left\lfloor \frac{|q_{ik}| \cdot w_{ik} \cdot \text{QP}}{16} \right\rfloor \]  
  \[ (595) \]

  with
  \[ q_{ik} \quad : \quad \text{Quantization index for coefficient at location } (i, k) \text{ inside block} \]
  \[ w_{i,k} \quad : \quad \text{Entry of quantization matrix for coefficient at location } (i, k) \]
  \[ \text{QP} \quad : \quad \text{Quantization parameter (also called “quantizer\_scale”) } \]
  \[ t'_{i,k} \quad : \quad \text{Reconstructed transform coefficient at location } (i, k) \]

- Inverse quantization is followed by clipping to range \([-2048, 2047]\) and, thereafter, the so-called mismatch control operation

  \[ t'_{77} = \begin{cases} 
  t'_{77} & : \quad s \text{ is odd} \\
  t'_{77} - 1 & : \quad s \text{ is even and } t'_{77} \text{ is odd} \\
  t'_{77} + 1 & : \quad s \text{ is even and } t'_{77} \text{ is even} 
\end{cases} \quad \text{with } s = \sum_{i=0}^{7} \sum_{k=0}^{7} t'_{ik} \]  
  \[ (596) \]
Quantization matrices

- Quantization matrices define variation of quantizer step sizes among frequencies
  - Can be used for psychovisual optimization (to some extent)
- Quantization parameter QP is used for scaling the quantization matrices
  - Operation point can be modified on macroblock basis with a few bits
- For 4:2:0 data, two matrices are used: one for intra and one for non-intra
- Default quantization matrices can be replaced by user-defined matrices

<table>
<thead>
<tr>
<th>default matrix for intra blocks</th>
<th>default matrix for non-intra blocks</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 16 19 22 26 27 29 34</td>
<td>16 16 16 16 16 16 16 16</td>
</tr>
<tr>
<td>16 16 22 24 27 29 34 37</td>
<td>16 16 16 16 16 16 16 16</td>
</tr>
<tr>
<td>19 22 26 27 29 34 34 38</td>
<td>16 16 16 16 16 16 16 16</td>
</tr>
<tr>
<td>22 22 26 27 29 34 37 40</td>
<td>16 16 16 16 16 16 16 16</td>
</tr>
<tr>
<td>22 26 27 29 32 35 40 48</td>
<td>16 16 16 16 16 16 16 16</td>
</tr>
<tr>
<td>26 27 29 32 35 40 48 58</td>
<td>16 16 16 16 16 16 16 16</td>
</tr>
<tr>
<td>26 27 29 34 38 46 56 69</td>
<td>16 16 16 16 16 16 16 16</td>
</tr>
<tr>
<td>27 29 35 38 46 56 69 83</td>
<td>16 16 16 16 16 16 16 16</td>
</tr>
</tbody>
</table>
Coding of Transform Coefficients Levels for Intra $8 \times 8$ Blocks

DC transform coefficient level in intra blocks
- Very similar to JPEG
- Prediction using last coded DC level (reset at start of slice or non-intra MB)
- Difference is coded by a category (called dct_dc_size) a fixed-length code
- Entropy coding table is fixed in standard (cannot be modified)

AC transform coefficient levels
- Levels of a block are converted into vector using a zig-zag scan (same as in JPEG)
- Vector of levels is coded using run-level code
  - Entropy coding table for most frequent combinations of run and level (actual levels including sign, not categories as in JPEG)
  - Includes end-of-block (EOB) symbol
  - Includes escape symbol for less likely combinations, for which the actual run and level are transmitted with 6 and 12 bit, respectively
- Entropy coding tables are fixed in standard (cannot be modified)
- For intra, one of two defined tables can be selected on a picture basis
Encoder Control for Intra Pictures in H.262 | MPEG-2 Video

What parameters can be chosen in encoder?
- On sequence/picture level: Quantization matrix, intra vlc table, intra DC precision
- On macroblock level: Quantization parameter QP
- On block level: Transform coefficient levels

⇒ Transform coefficient levels have largest impact on coding efficiency

Do the parameters of different blocks influence each other?
- Only last DC coefficient is used for prediction
- Typically very small impact on coding efficiency

⇒ Interdependencies between blocks can be neglected

How can the selection of transform coefficient levels be optimized?
- Selection determines distortion and rate!
- Rounding to next level minimizes distortion, but typically produces a large rate
- For an operation point given by a Lagrange parameter $\lambda$, the combined cost measure $D + \lambda \cdot R$ should be minimized
- Not straightforward due to run-level coding

⇒ Need to consider dependencies in coding of successive levels
⇒ Fixed decision levels cannot be optimal
Example: Impact of Considering Rate in Quantization

Quantization example with $\Delta = 10$ and $\lambda = 10$

- Consider quantization of the following vector of transform coefficients:

$$\begin{bmatrix} 36 & 18 & 23 & 3 & 12 & -4 & -3 & 2 & -6 & 0 & \cdots & 0 \end{bmatrix}$$

- Rounding to nearest quantization level according to $q = \text{round}(c/\Delta)$ yields:

$$\begin{bmatrix} 4 & 2 & 2 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & \cdots & 0 \end{bmatrix}$$

$\Rightarrow$ Sequence of (run,level) values: $(0,4)(0,2)(0,2)(1,1)(3,-1)(\text{EOB})$

$\Rightarrow$ Bit sequence: $(00001100)(01000)(01000)(0110)(001111)(10)$

$\Rightarrow$ Distortion $D = 87$, rate $R = 30$ $\Rightarrow$ $J = D + \lambda \cdot R = 387$

- Alternative quantization (considering rate by minimizing $J = D + \lambda \cdot R$)

$$\begin{bmatrix} 4 & 2 & 2 & 0 & 1 & 0 & 0 & 0 & 0 & \cdots & 0 \end{bmatrix}$$

$\Rightarrow$ Sequence of (run,level) values: $(0,4)(0,2)(0,2)(1,1)(\text{EOB})$

$\Rightarrow$ Bit sequence: $(00001100)(01000)(01000)(0110)(001111)(10)$

$\Rightarrow$ Distortion $D = 107$, rate $R = 24$ $\Rightarrow$ $J = D + \lambda \cdot R = 347 \ (< 387)$
Rate-Distortion Optimized Quantization (RDOQ)

General idea of rate-distortion optimized quantization for run-level coding

- Evaluate “all possible” vectors of transform coefficient levels
- Choose vector that minimizes \( J = D + \lambda \cdot R \), with
  - \( D \): Distortion for block (can be measured in transform domain)
  - \( R \): Rate for transmitting levels using given entropy coding tables
  - \( \lambda \): Lagrange parameter, e.g., given as function of quantization parameter QP

\[ \Rightarrow \text{Very complex: Require restriction for levels and suitable algorithm} \]

Select reasonable set of potential levels for each coefficient

- Following observations can be made (considering absolute levels)
  - Levels greater than the level obtained by mathematically correct rounding don’t need to be considered (larger distortion and larger rate)
  - Levels that are significantly smaller than the level obtained by mathematically correct rounding don’t need to be considered (very large distortion)
- Good compromise is obtained by following set of 1-3 levels (absolute values)
  - Level \( q_0 \) obtained by mathematically correct rounding, \( q_0 = \text{round}(c/\Delta) \)
  - Level \( q_1 \) obtained by rounding towards zero, \( q_1 = \lfloor c/\Delta \rfloor \)
  - If \( q_1 > 1 \), level \( q_2 = q_1 - 1 \)
RDOQ Algorithm for Run-Level Coding

Start with first position $k = 0$ in scanning order
- For all potential levels $\{q_0\}$ determine
  - distortion $D_0 = (c_0 - q_0 \cdot \Delta)^2$
  - for non-zero levels $q_0$, rate $R_0$ using run-level coding tables
- Among all non-zero levels $q_0$, keep only the one that minimizes $J_0 = D_0 + \lambda \cdot R_0$
- At most two candidate vectors $q_0 = [q_0]$ are considered for the further steps, one with $q_0 = 0$ and one with $q_0 \neq 0$

For each of the remaining positions $k$ in scanning order
- Combine all potential levels $\{q_k\}$ for current position $k$ with the selected candidate vectors $q_{k-1}$ to new candidate vectors $q_k$ and determine
  - distortion $D_k = D_{k-1} + (c_k - q_k \cdot \Delta)^2$
  - for non-zero levels $q_k$, rate $R_k$ using run-level coding tables
- Among all vectors $q_k$ with the last level $q_k$ unequal to 0, discard all vectors except the one that minimizes $D_k + \lambda \cdot R_k$

After last scanning position $k$
- Determined the final cost $J = D + \lambda \cdot R$ for all remaining candidate vectors $q$ (including EOB symbol) and choose the one that minimizes $J$
RDOQ Example

Simple RDOQ example using quantization step size $\Delta = 10$ and $\lambda = 10$

- Consider transform coefficient vector of 9 coefficients

| 36 | 18 | 23 | 3  | 12 | -4 | -3 | 2  | -6 |

- Coefficient “36” at position $k = 0$ with potential levels {4, 3, 2}:
  - $[4]$: $D = 4^2 = 16, R = 8 \rightarrow J = 96$
  - $[3]$: $D = 6^2 = 36, R = 6 \rightarrow J = 96$ [discard]
  - $[2]$: $D = 16^2 = 256, R = 5 \rightarrow J = 306$ [discard]

- Coefficient “18” at position $k = 1$ with potential levels {2, 1, 0}:
  - $[4, 2]$: $D = 16 + 2^2 = 20, R = 8 + 5 = 13 \rightarrow J = 150$
  - $[4, 1]$: $D = 16 + 8^2 = 80, R = 8 + 3 = 11 \rightarrow J = 190$ [discard]
  - $[4, 0]$: $D = 16 + 18^2 = 340, R = 8$ (does not include last zero)

- Coefficient “23” at position $k = 2$ with potential levels {2, 1}:
  - $[4, 2, 2]$: $D = 20 + 3^2 = 29, R = 13 + 5 = 18 \rightarrow J = 209$
  - $[4, 2, 1]$: $D = 20 + 13^2 = 189, R = 13 + 3 = 16 \rightarrow J = 349$ [discard]
  - $[4, 0, 2]$: $D = 340 + 3^2 = 349, R = 8 + 7 = 15 \rightarrow J = 499$ [discard]
  - $[4, 0, 1]$: $D = 340 + 13^2 = 509, R = 8 + 4 = 12 \rightarrow J = 629$ [discard]

- Coefficient “3” at position $k = 3$ with potential levels {0}:
  - $[4, 2, 2, 0]$: $D = 29 + 3^2 = 38, R = 18$ (without trailing zeros)
RDOQ Example (continued)

- Coefficient “12” at position $k = 4$ with potential levels $\{1, 0\}$:
  $\Rightarrow [4, 2, 2, 0, 1]: \quad D = 38 + 2^2 = 42, \quad R = 18 + 4 = 22 \quad \rightarrow J = 262$
  $\Rightarrow [4, 2, 2, 0, 0]: \quad D = 38 + 12^2 = 182, \quad R = 18 \quad \text{(without trailing zeros)}$

- Coefficient “-4” at position $k = 5$ with potential levels $\{0\}$:
  $\Rightarrow [4, 2, 2, 0, 1, 0]: \quad D = 42 + 4^2 = 58, \quad R = 22 \quad \text{(without trailing zeros)}$
  $\Rightarrow [4, 2, 2, 0, 0, 0]: \quad D = 182 + 4^2 = 198, \quad R = 18 \quad \text{(without trailing zeros)}$

- Coefficient “-3” at position $k = 6$ with potential levels $\{0\}$:
  $\Rightarrow [4, 2, 2, 0, 1, 0, 0]: \quad D = 58 + 3^2 = 67, \quad R = 22 \quad \text{(without trailing zeros)}$
  $\Rightarrow [4, 2, 2, 0, 0, 0, 0]: \quad D = 198 + 3^2 = 207, \quad R = 18 \quad \text{(without trailing zeros)}$

- Coefficient “2” at position $k = 7$ with potential levels $\{0\}$:
  $\Rightarrow [4, 2, 2, 0, 1, 0, 0, 0]: \quad D = 67 + 2^2 = 71, \quad R = 22 \quad \text{(without trailing zeros)}$
  $\Rightarrow [4, 2, 2, 0, 0, 0, 0, 0]: \quad D = 207 + 2^2 = 211, \quad R = 18 \quad \text{(without trailing zeros)}$

- Last coefficients “-6” with potential levels $\{-1, 0\}$ (including 2 bits for EOB):
  $\Rightarrow [4, 2, 2, 0, 1, 0, 0, 0, -1]: \quad D = 71 + 4^2 = 87, \quad R = 22 + 8 = 30 \quad \rightarrow J = 387$
  $\Rightarrow [4, 2, 2, 0, 1, 0, 0, 0, 0]: \quad D = 71 + 6^2 = 107, \quad R = 22 + 2 = 24 \quad \rightarrow J = 347$
  $\Rightarrow [4, 2, 2, 0, 0, 0, 0, 0, -1]: \quad D = 211 + 4^2 = 227, \quad R = 18 + 9 = 27 \quad \rightarrow J = 497$
  $\Rightarrow [4, 2, 2, 0, 0, 0, 0, 0, 0]: \quad D = 211 + 6^2 = 247, \quad R = 18 + 2 = 20 \quad \rightarrow J = 447$

$\Rightarrow$ Selected transform coefficient levels: $[4, 2, 2, 0, 1, 0, 0, 0, 0]$ with $J = 347$

$\Rightarrow$ For comparison, rounding would yield: $[4, 2, 2, 0, 1, 0, 0, 0, -1]$ with $J = 387$
Experimental Analysis of RDOQ for Intra-Picture Coding

Coding experiment with H.262 | MPEG-2 Video comparing
- Encoder with simple quantization using mathematically correct rounding
- Encoder using rate-distortion optimized quantization

Coding conditions
- 6 video conferencing sequences with a resolution of 1280 × 720
- 6 more complex video sequences with a resolution of 1920 × 1080
- 10 pictures of each sequence have been coded (intra only)
- Flat quantization matrices (since quality is measured using PSNR)
- Same quantization parameter for all macroblocks
- Bitstreams with different quantization parameters
- PSNR and rate have been measured
- Lagrange parameter $\lambda$ has been coupled to quantization step size $\Delta$ using the experimentally determined relationship

$$\lambda = \text{const} \cdot \Delta^2$$
Quantization Comparison – Sequence “Johnny”

Johnny, 1280x720, 60Hz

Y-PSNR [dB] vs. bit rate [kbit/s]

Simple Quantization
RD-opt. Quantization
Quantization Comparison – Sequence “Cactus”

Cactus, 1920x1080, 50Hz

Y-PSNR [dB] vs. bit rate [kbit/s]

Simple Quantization
RD-opt. Quantization
Quantization Comparison – Summary

Bit-rate savings of RDOQ versus simple quantization

- Bit-rate saving at a PSNR value is obtained by interpolating the r-d curves
- Average bit-rate savings are obtained by averaging the savings for 100 PSNR values
- Average bit-rate saving for all sequences: 9%

Johnny, 1280x720, 60Hz
RD-opt. Quantization

Cactus, 1920x1080, 50Hz
RD-opt. Quantization
Summary of Intra-Picture Coding in H.262 | MPEG-2 Video

Intra-picture coding in H.262 | MPEG-2 Video

- Video coding standard of ITU-T and ISO/IEC JTC 1
- Still widely used in digital television and DVD-Video
- Intra-picture coding is very similar to JPEG Baseline
- Picture partitioning in macroblocks and blocks
- Scalar quantization of transform coefficients
- Quantization matrices are combined with quantization parameter
- Run-level coding for transform coefficients

Rate-distortion optimized quantization

- Quantization with fixed decision levels cannot be optimal due to dependencies in run-level entropy coding
- Actual rate for coding transform coefficient levels need to be considered
- Optimal quantization can be achieved by a trellis-like procedure
- Rate-distortion optimized quantization yields average bit-rate savings of about 10% relative to simple rounding (for intra-only coding)
Intra-Picture Coding in H.263 and MPEG-4 Visual
ITU-T Recommendation H.263 – Overview

ITU-T Recommendation H.263
- Video coding standard of ITU-T
- Developed by Visual Coding Experts Group (VCEG – ITU-T SG16/WP3/Q6)
- Primarily designed for low bit-rate video conferencing
- Example applications:
  - Video conferencing
  - Was used for Flash Video content
  - RealVideo codec (before RealVideo 8) was based on H.263
- First version (1995)
  - Very similar structure as H.262 | MPEG-2 Video
  - Several improvements relative to H.262 | MPEG-2 Video
  - For intra: Run-level-last coding and optimized coding tables
- Second version H.263+ (1998)
  - Improvements and new features
  - For intra: AC level prediction, adaptive scans, specialized quantization and entropy coding tables for intra blocks
  - Multiple reference pictures for motion-compensated coding
Intra-Picture Coding in H.263 Baseline

Basic design for intra-picture coding – similar to H.262 | MPEG-2 Video
- Partitioning into $16 \times 16$ macroblocks as in H.262 | MPEG-2 Video
- Transform coding of $8 \times 8$ blocks (DCT + scalar quantization + entropy coding)

Orthogonal block transform
- Same separable $8 \times 8$ DCT as in H.262 | MPEG-2 Video and JPEG

Scalar quantization of transform coefficients (only reconstruction is specified)
- Step size determined by quantization parameter QP (can be modified on MB basis)
- DC coefficient (uniform reconstruction quantizer)
  \[ t_{00}' = 8 \cdot q_{00} \]  

\[ t_{ik}' = \begin{cases} 
0 & : q_{ik} = 0 \\
\text{sgn}(q_{ik}) \cdot \text{QP} \cdot (2 \cdot |q_{ik}| + 1) & : q_{ik} \neq 0 \text{ and QP is odd} \\
\text{sgn}(q_{ik}) \cdot \text{QP} \cdot (2 \cdot |q_{ik}| + 1) - 1 & : q_{ik} \neq 0 \text{ and QP is even} 
\end{cases} \]  

Reconstructed levels are always odd-valued numbers (except zero)
Has been found to prevent accumulation of IDCT mismatches (for inter)
Coding of Transform Coefficient Levels in H.263 Baseline

Coded block pattern (CBP)

- Signal which $8 \times 8$ blocks of a macroblock contain non-zero levels (AC levels)
- Concept was also used in H.262 | MPEG-2 Video, but only for inter macroblocks
- Required for run-level-last coding of AC levels
- In H.263, split into two components:
  - CBPC (two bits for the chroma blocks): Coded together with MB type
  - CBPY (four bits for the luma blocks): Coded as separate codeword

<table>
<thead>
<tr>
<th>MB type</th>
<th>CBPC</th>
<th>codeword</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intra</td>
<td>00</td>
<td>1</td>
</tr>
<tr>
<td>Intra</td>
<td>01</td>
<td>001</td>
</tr>
<tr>
<td>Intra</td>
<td>10</td>
<td>010</td>
</tr>
<tr>
<td>Intra</td>
<td>11</td>
<td>011</td>
</tr>
<tr>
<td>Intra+Q</td>
<td>00</td>
<td>0001</td>
</tr>
<tr>
<td>Intra+Q</td>
<td>01</td>
<td>0000 01</td>
</tr>
<tr>
<td>Intra+Q</td>
<td>10</td>
<td>0000 10</td>
</tr>
<tr>
<td>Intra+Q</td>
<td>11</td>
<td>0000 11</td>
</tr>
<tr>
<td>stuffing</td>
<td>–</td>
<td>0000 0000 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CBPY</th>
<th>codeword</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>0011</td>
</tr>
<tr>
<td>0001</td>
<td>0010 1</td>
</tr>
<tr>
<td>0000</td>
<td>0010 0</td>
</tr>
<tr>
<td>0000</td>
<td>1011</td>
</tr>
<tr>
<td>0000</td>
<td>0011 1</td>
</tr>
<tr>
<td>0000</td>
<td>0101</td>
</tr>
<tr>
<td>0000</td>
<td>0000 10</td>
</tr>
<tr>
<td>0000</td>
<td>1011</td>
</tr>
<tr>
<td>0000</td>
<td>1001</td>
</tr>
<tr>
<td>0000</td>
<td>1100</td>
</tr>
<tr>
<td>0000</td>
<td>0100</td>
</tr>
<tr>
<td>0000</td>
<td>1101</td>
</tr>
<tr>
<td>0000</td>
<td>1000</td>
</tr>
<tr>
<td>0000</td>
<td>1110</td>
</tr>
<tr>
<td>0000</td>
<td>0110</td>
</tr>
<tr>
<td>0000</td>
<td>1111</td>
</tr>
<tr>
<td>stuffing</td>
<td>1</td>
</tr>
</tbody>
</table>

Coding of DC transform coefficient level

- No prediction of DC coefficient (in contrast to JPEG and H.262 | MPEG-2 Video)
- Fixed-length code: 8 bit per DC level
Coding of AC Transform Coefficient Levels in H.263 Baseline

Coding of AC transform coefficient levels
- Convert matrix of AC levels to vector using the zig-zag scan
- Vector of AC levels is coded using run-level-last code
- Entropy coding table for most common combinations of
  - Run: Number of preceding levels equal to zero
  - Level: Value of the next non-zero level
  - Last: Flag indicating if the non-zero level is the last non-zero level in block
- Entropy coding table includes an escape symbol for less likely combinations, for which the “run”, “level” and “last” are coded using fixed-length codes (6+8+1 bit)
- Entropy coding tables have been optimized for low rates

<table>
<thead>
<tr>
<th>last</th>
<th>run</th>
<th>level</th>
<th>codeword ($s = \text{sign}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>±1</td>
<td>10s</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>±2</td>
<td>1111 s</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>±3</td>
<td>0101 01s</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>±4</td>
<td>0010 111s</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>±5</td>
<td>0001 1111 s</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>±6</td>
<td>0001 0010 1s</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>±7</td>
<td>0001 0010 0s</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>±8</td>
<td>0000 1000 01s</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>±9</td>
<td>0000 1000 00s</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>±10</td>
<td>0000 0000 111s</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>±11</td>
<td>0000 0000 110s</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>±12</td>
<td>0000 0100 000s</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>last</th>
<th>run</th>
<th>level</th>
<th>codeword ($s = \text{sign}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>31</td>
<td>±1 0000 0100 110s</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>32</td>
<td>±1 0000 0100 111s</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>33</td>
<td>±1 0000 0101 1000 s</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>34</td>
<td>±1 0000 0101 1001 s</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>35</td>
<td>±1 0000 0101 1010 s</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>36</td>
<td>±1 0000 0101 1011 s</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>37</td>
<td>±1 0000 0101 1100 s</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>38</td>
<td>±1 0000 0101 1101 s</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>39</td>
<td>±1 0000 0101 1110 s</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>40</td>
<td>±1 0000 0101 1111 s</td>
</tr>
</tbody>
</table>

escape 0000 011
Advanced Intra Coding Mode in Annex I of H.263+

Advanced intra-picture coding mode (in optional Annex I) specifies

- Adaptive prediction of transform coefficients in intra blocks
- Adaptive scanning of transform coefficient levels (depending on prediction)
- Modified inverse quantization for intra
- Separate entropy coding table for intra blocks (optimized for intra)

Adaptive prediction and scanning

- Predict part of the transform coefficients using reconstructed transform coefficients of neighboring blocks of same color component
- Quantization and entropy coding of prediction residuals $t_{ik} - \hat{t}_{ik}$

$$t'_{ik} = \hat{t}_{ik} + Q^{-1}(q_{ik}) \quad (Q^{-1}: \text{inverse quantization}) \quad (599)$$

- 3 prediction modes (signaled at macroblock level)
  - DC prediction: Predict DC using left and above neighboring block
  - Vertical prediction: Predict first row of coefficients using above block
    $\Rightarrow$ Particularly suitable for vertical structures
  - Horizontal prediction: Predict first column of coefficients using left block
    $\Rightarrow$ Particularly suitable for horizontal structures
Prediction Modes for Intra Blocks in Advanced Intra Coding

Prediction of transform coefficients: 3 modes signaled at macroblock level

- DC prediction:
  \[ \hat{t} = \left\lfloor \frac{t_A + t_B}{2} \right\rfloor \]
- Vertical pred.:
  \[ \hat{t} = t_A \]
- Horizontal pred.:
  \[ \hat{t} = t_B \]

Modified quantization for intra macroblocks

- Use same quantization for all coefficients (including DC coefficient)
- Uniform reconstruction quantizer without extra-wide deadzone and without mismatch control (only important for inter macroblocks)

\[ t'_{ik} = 2 \cdot QP \cdot q_{ik}, \quad \forall i, k = 0..7 \] (600)
Scanning pattern is chosen based on prediction mode

- **Goal:** Concentrate zero levels at end of scanning pattern
- **Non-zero coefficient distribution is depending on prediction mode**
  - **DC prediction:** Conventional zig-zag scan
  - **Vertical prediction:** Horizontal scan (suitable for vertical structures)
  - **Horizontal prediction:** Vertical scan (suitable for horizontal structures)

**Coding of vector of transform coefficient levels**

- **No separate coding of DC coefficient** $\implies$ included in run-level-last code
- **Entropy coding table for run-level-last code optimized for intra statistics**
Further Extensions in H.263 for Improving Intra Coding

Annex E: Syntax-based arithmetic coding
- Specifies non-adaptive arithmetic coding for syntax elements
- Rarely used in practice

Annex J: Deblocking filter mode
- Specifies deblocking filter for reducing block-edge artifacts
- Strength of smoothing filter is controlled by quantization parameter
- Most useful for coding of following inter pictures, but also improves quality of intra pictures at low rates

Annex T: Modified quantization mode
- Improves ability to control bit rate (finer steps for modifying QP)
- Extends range of representable transform coefficient values
- Improves chroma fidelity by choosing smaller quantization step size for chroma than for luma (particularly for low rates)
Intra-Picture Coding in MPEG-4 Visual

International standard ISO/IEC 14496-2 (MPEG-4 Visual)
- Video coding standard of Moving Pictures Experts Group (MPEG)
- Includes H.263 Baseline decoder, contains several extensions
- Was used in digital cameras, DivX, ...

Coding of intra macroblocks
- Prediction of transform coefficients (similar to H.263 Annex I)
  - DC level is always predicted from left or above block
    (direction is determined by differences of neighboring DC levels)
  - First row/column of ACs can be optionally predicted using same prediction
direction as for DC (usage is signaled at macroblock level)
- Two methods for quantization
  - MPEG-style: Quantization as in MPEG-2 Video (including weighting matrix)
  - H263-style: Quantization as in H.263 Baseline
- Scanning of transform coefficients
  - 3 scans: Zig-zag, horizontal, vertical (chosen based on prediction mode)
- Coding of vector of transform coefficient levels
  - Coded block pattern and run-level-last coding
Encoder Control for Intra Coding in H.263 and MPEG-4 Visual

Increased degree of freedom relative to H.262 | MPEG-2 Video

- In addition to transform coefficient levels, the method for transform coefficient selection can be selected on macroblock level
- H.263 (Annex I): DC, vertical or horizontal prediction mode
- MPEG-4 Visual: DC or DC and AC prediction

Rate-distortion optimized encoder control

- Determine bitstream $b$ so that the distortion $D(s, s')$ between original picture $s$ and reconstructed picture $s'$ is minimized given a particular target rate $R \leq R_{\text{target}}$
- With $B_c$ being the set of conforming bitstreams with $R \leq R_{\text{target}}$, we can write

$$b^* = \arg \min_{b \in B_c} D(s, s'(b))$$ \hspace{1cm} (601)

⇒ Not feasible due to huge parameter space
⇒ Split into smaller optimization problems by partially ignoring dependencies

- Consider block of samples $s_k$ (e.g., picture or macroblock) and optimize with respect to coding parameters $p_k$ (e.g., modes and transform coefficient levels)

$$\min_{p_k} D(s_k, s'_k(p_k)) \quad \text{subject to} \quad R(p_k) \leq R_c$$ \hspace{1cm} (602)
Rate-distortion optimized encoder control

Lagrangian encoder control

- Constrained optimization problem for a block of samples $s_k$

$$\min_{p_k} D(s_k, s'_k(p_k)) \quad \text{subject to} \quad R(p_k) \leq R_c$$

(603)

can be reformulated as unconstrained optimization problem

$$\min_{p_k} D(s_k, s'_k(p_k)) + \lambda \cdot R(p_k)$$

(604)

- Consider partition of $s_k$ into a number of subsets $s_{k,i}$ (e.g. macroblocks)
- If coding parameters $p_{k,i}$ are independent of each other and an additive distortion measure is used, we can write the optimization problem as

$$\sum_i \min_{p_{k,i}} D(s_{k,i}, s'_{k,i}(p_{k,i})) + \lambda \cdot R(p_{k,i})$$

(605)

⇒ Independent selection of coding parameters $p_{k,i}$

- For coding decisions in image and video coding

  - Coding decisions are typically not independent (e.g., due to prediction)
  - For practical applicability: Consider past decisions, but ignore impact on future
Lagrangian Encoder Control in Image and Video Coding

Application of Lagrangian encoder control

- Can be applied to basically all decisions in an encoder
- **Quantization**: Select vector $q$ of transform coefficient levels according to

$$q^* = \arg \min_q D(q) + \lambda \cdot R(q)$$  \hspace{1cm} (606)

with $D(q)$: SSD distortion for choosing transform coefficient level vector $q$
$R(q)$: Number of bits required for representing $q$

$\implies$ Rate-distortion optimized quantization (as considered for run-level coding)

- **Mode decision**: Select coding mode $c$ for a macroblock or block

$$c^* = \arg \min_c D(c) + \lambda \cdot R(c)$$  \hspace{1cm} (607)

with $D(c)$: SSD distortion for choosing coding mode $c$ for the block
$R(c)$: Number of bits for block when coded with mode $c$

$\implies$ Can be applied for selecting intra prediction mode

- **Motion search**: Will be considered later in lecture
Comparison of H.263 and MPEG-4 Visual with MPEG-2 Video

Comparison of coding efficiency for intra-picture coding

- Selection of all features that contribute to coding efficiency
  - H.262 | MPEG-2 Video conforming to Main profile
  - H.263+ with advanced intra coding, deblocking filter, modified quantization
  - MPEG-4 Visual with MPEG-style quantization

- Apply same level of encoder optimization for fair comparison
  - Best possible coding efficiency for given syntax
  - Ignore constraints such as real-time operation
    \[ \Rightarrow \text{Use rate-distortion optimized quantization for all standards} \]
    \[ \Rightarrow \text{Apply rate-distortion optimized mode decision where applicable} \]

- General coding conditions
  - Encode 10 pictures of 12 video sequences (6 in 720p, 6 in 1080p)
  - Flat quantization matrices (quality is measured using PSNR)
  - Same quantization parameter for all macroblocks
  - Select Lagrangian parameter according to
    \[ \lambda = \text{const} \cdot \Delta^2 \] (with experimentally determined factor) \hspace{1cm} (608)
Intra Coding Comparison – Sequence “Johnny”

Johnny, 1280x720, 60Hz

Y-PSNR [dB] vs. bit rate [kbit/s]

- H.262 | MPEG-2 Video
- H.263+
- MPEG-4 Visual
Intra Coding Comparison – Sequence “Cactus”

Cactus, 1920x1080, 50Hz

H.262 | MPEG-2 Video
H.263+
MPEG-4 Visual
Intra Coding Comparison – Summary

Bit-rate savings of H.263 and MPEG-4 Visual versus H.262 | MPEG-2 Video
- Bit-rate saving at a PSNR value is obtained by interpolating the r-d curves
- Average bit-rate savings are obtained by averaging the savings for 100 PSNR values
- Average bit-rate saving for all sequences
  - H.263+ versus H.262 | MPEG-2 Video: 29%
  - MPEG-4 Visual versus H.262 | MPEG-2 Video: 25%
- Highest savings are obtained for low bit rates

Y-PSNR [dB]

Johnny, 1280x720, 60Hz

Cactus, 1920x1080, 50Hz

rate savings vs. H.262 | MPEG-2 Video
Summary of Intra-Picture Coding in H.263 and MPEG-2 Visual

Intra-picture coding in H.263+
- Transform coding using $8 \times 8$ DCT
- Prediction of DC and, partly, AC coefficients
- Coded block pattern
- Adaptive scanning of transform coefficient levels
- Scalar quantization
- Run-level-last coding of transform coefficient levels
- Optional de-blocking filter

Intra-picture coding in MPEG-4 Visual
- Similar tools as in advanced intra coding mode of H.263+ (Annex I)
- Additionally includes quantization weighting matrices

Rate-distortion optimized encoder control
- Split overall optimization problem into smaller problems
- Encoder decision by minimizing $D + \lambda \cdot R$
  ⇒ Rate-distortion optimized quantization (RDOQ)
  ⇒ Rate-distortion optimized mode decision
Intra-Picture Coding in H.264 | MPEG-4 AVC
H.264 | MPEG-4 AVC – Overview

ITU-T Rec. H.264 | ISO/IEC 14496-10 (MPEG-4 Advanced Video Coding)

- Video coding standard jointly developed by ITU-T VCEG and ISO/IEC MPEG
- Widely used today in many application spaces
  - Digital television, blu-ray optical disc, digital cameras, mobile phones, video streaming, video conferencing
  - Supported in more than 1 billion devices
  - Every second bit in the Internet is part of a H.264 | MPEG-4 AVC bitstream
- Version 1 (2003):
  - Three profiles: Baseline, Main, Extended (4:2:0, 8 bit)
- Fidelity range extensions (2005)
  - Improvements for large picture sizes, other chroma formats, higher bit depth
  - High, High 10, High 4:2:2, High 4:4:4 profiles (removed later)
  - Later: Addition of High 4:4:4 Predictive and intra-only profiles
- Scalable video coding extension (2007)
  - Extension for scalable video coding (SVC)
- Multiview video coding extension (2009)
  - Extension for multiview video (MVC)
  - Used for 3d-blu-ray discs
Main features of intra-picture coding

- Spatial intra prediction with multiple prediction modes
- Transform coding with $4 \times 4$ integer transform
- Optional $8 \times 8$ integer transform (High profile)
- Scalar quantization
- Two entropy coding methods:
  - Context-adaptive variable length coding (CAVLC)
  - Context-adaptive binary arithmetic coding (CABAC) – Main/High profile
- Deblocking filter (conceptually similar to Annex J of H.263)

Picture partitioning and intra coding modes

- Pictures are partitioned into $16 \times 16$ macroblocks
- 4 intra coding modes are supported (selection on MB level)
  - Intra-$4 \times 4$
  - Intra-$8 \times 8$ (High profile)
  - Intra-$16 \times 16$
  - Intra-PCM (direct coding of samples)
Coding of Intra-4×4 macroblocks

Spatial prediction of blocks
- DC, horizontal, vertical prediction of transform coefficients (as in H.263, MPEG-4 Visual) can similarly also be realized in spatial domain
- Prediction in spatial domain offers more possibilities

Coding of luma component
- 16×16 luma block is partitioned into 16 4×4 blocks
- 4×4 blocks are spatially predicted
- 9 intra prediction modes are supported
- Prediction error of 4×4 blocks is transform-coded

Coding of chroma components
- Both 8×8 chroma blocks are spatially predicted
- 4 intra prediction modes are supported
- Prediction error is coded using transform coding
- 4×4 transform and second level transform of DCs

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
<td>6</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>12</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>11</td>
<td>14</td>
<td>15</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>18</th>
<th>19</th>
<th>22</th>
<th>23</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>21</td>
<td>24</td>
<td>25</td>
<td></td>
</tr>
</tbody>
</table>
Spatial Intra Prediction for $4 \times 4$ Blocks

Spatial intra prediction
- Predict block using already coded neighboring samples
- DC prediction (mean value) and 8 directional prediction modes

![Diagram of spatial intra prediction]

Coding order and boundary conditions
- Blocks are coded in z-scan order
- Samples used for prediction have to be "available" (already reconstructed)
- Not all modes are available for all blocks
- Samples "E", "F", "G" and "H" can be replaced with sample "D"
Illustration of Intra Prediction Modes

Mode 0 - Vertical

Mode 1 - Horizontal

Mode 2 - DC

Mode 3 – Diagonal Down/Left

Mode 4 – Diagonal Down/Right

Mode 5 – Vertical-Right

Mode 6 – Horizontal-Down

Mode 7 – Vertical-Left

Mode 8 – Horizontal-Up
Efficiency of Intra Prediction – Sequence “Johnny”

Johnny, 1280x720, 60Hz

Bit rate savings
vs DC-only: 15.5%
vs DC+H+V: 8.0%

Y-PSNR [dB]
bit rate [kbit/s]
Efficiency of Intra Prediction – Sequence “Cactus”

Cactus, 1920x1080, 50Hz

Bit rate savings
vs DC-only: 7.2%
vs DC+H+V: 6.6%
Transform Coding of $4 \times 4$ Blocks

Separable $4 \times 4$ integer transform

- Orthogonal block transform with integer approximation of DCT

$$u_{4\times4} = A_{4\times4} \cdot s_{4\times4} \cdot A_{4\times4}^T$$  with  

$$A_{4\times4} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}$$ (609)

- Inverse is specified by exact integer operations
  \[\implies\]  No accumulation of transform mismatches

- Easy implementation (only additions and bit shift operations)

- Basis vectors have different norms
  \[\implies\]  Compensated by modifying the quantizer step size accordingly

Scalar quantization of transform coefficients

- Uniformly distributed reconstruction levels
- Logarithmic quantization step size control ($\Delta \approx \alpha \cdot 2^{QP/6}$)
- Smaller quantization step sizes for chroma (as in Annex T of H.263)
- Support for quantization weighting matrices (High profile)
- Quantization parameter can change at macroblock level
Coding of Transform Coefficient Levels

Context-adaptive variable length coding (CAVLC)

- Coded block pattern (for all six $8 \times 8$ block) using VLC table
- Zig-zag scan for mapping matrix into vector
- Syntax element “coeff_token” for $4 \times 4$ blocks
  - Specifies number of non-zero coefficients and number of trailing ones
  - Chosen VLC table depends on number of non-zero coefficients in already coded neighboring blocks
- Additionally code “runs” and “levels” as well as signs for trailing ones

Context-adaptive binary arithmetic coding (CABAC) [Main, High profile]

- Binary arithmetic coding of all low-level syntax elements
- Coded block pattern (flag for each of the six $8 \times 8$ blocks)
- Flag for $4 \times 4$ blocks indicating whether non-zero levels are present
- Coding of a significance map
  - “significance_flag” indicating whether level is non-zero
  - if non-zero, “last_flag” indicating whether last non-zero level
- Coding of absolute levels (minus 1) and signs
- Probability models are adapted to statistics during encoding and decoding
Intra-16 × 16 and Intra-8 × 8 Macroblocks

Intra-16 × 16 macroblock mode
- Prediction of 16 × 16 luma block
- Four prediction modes: DC, horizontal, vertical, planar
- 4 × 4 transform of all sixteen 4 × 4 blocks
- Additional 4 × 4 Hadamard transform of DC coefficients
- DC block is treated separately in entropy coding
- Similar concept with 2 × 2 Hadamard transform of DC coefficients is also used for chroma blocks (for all intra coding modes)

Intra-8 × 8 macroblock mode (High profile)
- Prediction of 8 × 8 blocks of luma component
- Same prediction modes as for Intra-4 × 4 (but extended to larger block size)
- Reference samples are low-pass filtered before they are used for prediction
- 8 × 8 integer transform for the four luma sub-blocks
- Entropy coding is extended to 8 × 8 transform blocks
Coding Efficiency of Intra Modes – Sequence “Johnny”

Johnny, 1280x720, 60Hz

![Graph showing PSNR vs bit rate for different intra modes. The graph compares Intra4x4, Intra8x8, Intra16x16, Main profile (4x4, 16x16), and High profile (all modes).]
Coding Efficiency of Intra Modes – Sequence “Cactus”

Cactus, 1920x1080, 50Hz

Y-PSNR [dB] vs. bit rate [kbit/s] for different intra modes in H.264/MPEG-4 AVC.

- Intra4x4
- Intra8x8
- Intra16x16
- Main profile (4x4, 16x16)
- High profile (all modes)
Coding Efficiency Comparison with Older Standards

Comparison of coding efficiency for intra-picture coding

- Selection of all features that contribute to coding efficiency
  - H.262 | MPEG-2 Video conforming to Main profile
  - H.263+ with advanced intra coding, deblocking filter, modified quantization
  - MPEG-4 Visual with MPEG-style quantization
  - H.264 | MPEG-4 AVC High profile with CABAC

- Apply same level of encoder optimization for fair comparison
  - Best possible coding efficiency for given syntax
  - Ignore constraints such as real-time operation
  - Use rate-distortion optimized quantization for all standards
  - Apply rate-distortion optimized mode decision where applicable

- General coding conditions
  - Encode 10 pictures of 12 video sequences (6 in 720p, 6 in 1080p)
  - Flat quantization matrices (quality is measured using PSNR)
  - Same quantization parameter for all macroblocks
  - Select Lagrangian parameter according to
  \[
  \lambda = \text{const} \cdot \Delta^2
  \]
  (with experimentally determined factor) (610)
Intra Coding Comparison – Sequence “Johnny”

Johnny, 1280x720, 60Hz

Y-PSNR [dB] vs. bit rate [kbit/s] for different video coding standards:
- H.262 | MPEG-2 Video
- MPEG-4 Visual
- H.263+
- H.264 | MPEG-4 AVC HP

Heiko Schwarz
Source Coding and Compression
December 7, 2013
Intra Coding Comparison – Sequence “Cactus”

Cactus, 1920x1080, 50Hz

Y-PSNR [dB] vs. bit rate [kbit/s]

- H.262 | MPEG-2 Video
- MPEG-4 Visual
- H.263+
- H.264 | MPEG-4 AVC HP
Intra Coding Comparison – Summary

Bit-rate savings of H.264 | MPEG-4 AVC versus older video coding standards

- Bit-rate saving at a PSNR value is obtained by interpolating the r-d curves
- Average bit-rate savings are obtained by averaging the savings for 100 PSNR values
- Highest savings are obtained for low bit rates
- Average bit-rate saving for all sequences are summarized below

<table>
<thead>
<tr>
<th>codec</th>
<th>average bit rate savings relative to ...</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>H.263+</td>
</tr>
<tr>
<td>H.264 / AVC</td>
<td>21.1 %</td>
</tr>
<tr>
<td>H.263+</td>
<td>9.7 %</td>
</tr>
<tr>
<td>MPEG-4</td>
<td></td>
</tr>
</tbody>
</table>
Intra-picture coding in H.264 | MPEG-4 AVC

- Four intra macroblock modes
  - Intra-4×4, Intra-8×8, Intra-16×16: Prediction & transform coding
  - Intra-PCM: Direct coding of samples

- Intra prediction in spatial domain using neighboring samples
  - Intra-4×4 and Intra-8×8: Eight directional modes & DC prediction
  - Intra-16×16 and chroma: Four intra prediction modes

- Transform: Integer approximation of DCT
  - Intra-4×4: Transform of 4×4 blocks
  - Intra-8×8: Transform of 8×8 blocks
  - Intra-16×16 and chroma: Transform of 4×4 blocks + DC transform

- Scalar quantization
  - Uniform reconstruction quantizer (with optional weighting matrices)
  - Norms of basis vectors are taken into account in quantization

- Two methods for entropy coding
  - Context-adaptive variable length coding (CAVLC)
  - Context-adaptive binary arithmetic coding (CABAC)

- Deblocking filter
Intra-Picture Coding in H.265 | MPEG-H HEVC
H.265 | MPEG-H HEVC — Overview


- Jointly developed by ITU-T VCEG and ISO/IEC MPEG
- Last video coding standard with focus on coding of high-resolution video
- Significantly increased coding efficiency, particularly for high-resolution video
- First version was finalized in January 2013
- First version specifies three profiles
  - Main profile (4:2:0 chroma format, 8 bit per sample)
  - Main 10 profile (4:2:0 chroma format, 10 bit per sample)
  - Main Still Picture profile (intra-only coding subset of Main profile)
- Extensions are under development
  - Fidelity range extension (other chroma samplings, higher bit depth)
  - Scalable video coding extension
  - Multiview video coding extension
  - Multiview video plus depth coding extension
Main Features of H.265 | MPEG-H HEVC

Main improvements relative to H.264 | MPEG-4 AVC
- Larger block sizes for transform coding and motion compensation
- Increased flexibility for partitioning a picture into blocks
- Improved interpolation filters and motion vector coding
- Increased number of intra prediction modes
- Improved coding of transform coefficient levels
- Additional in-loop filter: Sample-adaptive offset filter

Intra-Picture Coding in H.265 | MPEG-H HEVC
- Spatial intra prediction and transform coding of prediction residual
- Increased number of intra prediction modes compared to H.264/AVC
- Larger transform sizes
- More flexible partitioning of a picture
- Improved coding of transform coefficient levels (for larger blocks)
- Deblocking filter and additional sample-adaptive offset filter
Still Image Coding
Intra-Picture Coding in H.265 | MPEG-H HEVC

Picture Partitioning in H.265 | MPEG-H HEVC

Picture partitioning into coding tree blocks
- Coding tree blocks (CTBs): Fixed size of 16×16, 32×32 or 64×64 luma samples
- Size of CTBs chosen by encoder
- Luma and chroma CTBs together with syntax are called coding tree unit (CTU)

Partitioning of coding tree blocks
- Quad-tree partitioning into coding blocks (CBs)
- Luma and chroma CBs together with syntax are called coding unit (CU)
- Maximum CU size: Size of the CTB
- Minimum CU size: Selected by encoder, but equal to or larger than 8×8 luma samples
- Coding mode (intra or inter) is chosen for CU
- CU is similar to macroblock in older standards
- Coding order: Z-scan
Example: Picture Partitioning into Coding Units

Example for picture partitioning into coding units

- Picture with $2560 \times 1600$ luma samples of HEVC test sequence “Traffic”
- Quadtree-based partitioning into coding unit represents a simple scheme for locally adapting the block sizes to the image structure
Partitioning of a CB into **transform blocks** (TBs)

- Nested quad-tree partitioning
- TB corresponds to a single block transform
- Min. and max. TB size are selected by encoder
- Supported transforms: $4 \times 4$, $8 \times 8$, $16 \times 16$, $32 \times 32$
- Luma and chroma TBs together with syntax form a **transform unit** (TU)
- Special case: Chroma $4 \times 4$ blocks are not split

**Intra prediction and mode signaling**

- One or four luma intra prediction modes per coding unit
- One chroma prediction mode per CU
- Actual intra prediction is performed transform block by transform block
  \[\Rightarrow\] Improved prediction accuracy
Spatial Intra Prediction of Transform Blocks

Prediction of luma transform blocks
- Spatial intra prediction using neighboring samples of already coded blocks
- 33 directional prediction modes
- Additionally: DC and planar prediction
- Reference sample smoothing depending on block size and prediction direction

Prediction of chroma transform blocks
- Same prediction mechanism as for luma
- Prediction mode can be selected among the following:
  - Planar, DC, horizontal, vertical prediction mode
  - Same prediction modes as for associated luma block
Number of Intra Prediction Modes – Sequence “Johnny”

Johnny, 1280x720, 60Hz

Bit rate savings vs 9 modes: 12.5%
Number of Intra Prediction Modes – Sequence “Cactus”

Cactus, 1920x1080, 50Hz

Bit rate savings vs 9 modes: 9.4%
Transform and Quantization in H.265 | MPEG-H HEVC

Separable 2-d transform

- 2-d block transform of the size of the transform block
- Separable integer approximation of discrete cosine transform (DCT)
- Basis function have approximately the same norm
- Supported transform sizes: $4 \times 4$, $8 \times 8$, $16 \times 16$, $32 \times 32$
- All transform sizes are specified by single $32 \times 32$ integer matrix
- Exception: $4 \times 4$ luma TBs of intra-picture predicted CUs
  - Separable integer approximation of discrete sine transform (DST)
  - Better fits the statistical properties that the residual amplitudes tend to increase with increasing distance from reference samples

Quantization

- Same uniform reconstruction level quantizer as in H.264 | MPEG-4 AVC
- Quantization step size controlled by quantization parameter (QP)
- Approximately logarithmic mapping between QP and step size
- Quantization scaling matrices are supported
Block Sizes for Prediction and Transform – Sequence “Johnny”

Johnny, 1280x720, 60Hz

Bit rate savings vs $4 \times 4$ and $8 \times 8$: 11.7%
Block Sizes for Prediction and Transform – Sequence “Cactus”

Bit rate savings vs $4 \times 4$ and $8 \times 8$: 4.1%
Entropy Coding of Transform Coefficient Levels

Only a single entropy coding method
- Context-adaptive binary arithmetic coding (CABAC)
- Core algorithm of CABAC is unchanged relative to H.264 | MPEG-4 AVC

Coding of transform coefficient levels
- Coded block flag for transform block
- $x$ and $y$ coordinate of last significant level in scanning order
- Significance flags for $4 \times 4$ sub-blocks
- Significance map for $4 \times 4$ sub-blocks
- Absolute levels and signs

Adaptive coefficient scanning
- For $4 \times 4$ and $8 \times 8$ TB in intra CUs, scan depends on intra prediction mode (horizontal, vertical, diagonal)
- For all other blocks: Diagonal scan
In-Loop Filtering

Deblocking filter
- Adaptive filter for reducing block edge boundaries
- Similar adaptation of boundary filter strength as in H.264 | MPEG-4 AVC
- Applied on $8 \times 8$ sample grid

Sample adaptive offset (SAO)
- Conditional adding of an offset to samples (offset are transmitted)
- Selected on region basis: Either not used or applied in one of two modes
  - Mode 1: Band offset filtering
    - Offset depends on sample value
    - Split amplitude range into 32 bands
    - Offset values are transmitted for four consecutive bands
  - Mode 2: Edge offset filtering
    - Choose one of 4 gradient directions
    - Classify sample into one of 5 categories based on neighbouring samples
    - Offset values are transmitted for 4 of these sample classes
Coding Efficiency Comparison with Prior Standards

Comparison of coding efficiency for intra-picture coding
- Selection of all features that contribute to coding efficiency
  - H.262 | MPEG-2 Video conforming to Main profile
  - H.263+ with advanced intra coding, deblocking filter, modified quantization
  - MPEG-4 Visual with MPEG-style quantization
  - H.264 | MPEG-4 AVC High profile with CABAC
  - H.265 | MPEG-H HEVC Main profile

- Apply same level of encoder optimization for fair comparison
  - Best possible coding efficiency for given syntax
  - Ignore constraints such as real-time operation
  \[ \lambda = \text{const} \cdot \Delta^2 \]  
  (with experimentally determined factor) (611)

General coding conditions
- Encode 10 pictures of 12 video sequences (6 in 720p, 6 in 1080p)
- Flat quantization matrices (quality is measured using PSNR)
- Same quantization parameter for all macroblocks
- Select Lagrangian parameter according to
  \[ \lambda = \text{const} \cdot \Delta^2 \]  
  (with experimentally determined factor) (611)
Intra Coding Comparison – Sequence “Johnny”

Johnny, 1280x720, 60Hz

- Y-PSNR [dB]
- bit rate [kbit/s]

- H.262 | MPEG-2 Video
- MPEG-4 Visual
- H.263+
- H.264 | MPEG-4 AVC HP
- H.265 | MPEG-H HEVC

Heiko Schwarz

Source Coding and Compression
December 7, 2013
Intra Coding Comparison – Sequence “Cactus”

Cactus, 1920x1080, 50Hz

Y-PSNR [dB] vs. bit rate [kbit/s] for different video codecs.
Intra Coding Comparison – Summary

Bit-rate savings of H.265 | MPEG-H HEVC versus older video coding standards

- Bit-rate saving at a PSNR value is obtained by interpolating the r-d curves
- Average bit-rate savings are obtained by averaging the savings for 100 PSNR values
- Highest savings are obtained for low bit rates
- Average bit-rate saving for all sequences are summarized below

<table>
<thead>
<tr>
<th>codec</th>
<th>H.264 / AVC</th>
<th>H.263+</th>
<th>MPEG-4</th>
<th>MPEG-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>H.265 / HEVC</td>
<td>25.1 %</td>
<td>40.3 %</td>
<td>45.2 %</td>
<td>57.9 %</td>
</tr>
<tr>
<td>H.264 / AVC</td>
<td>20.7 %</td>
<td>27.7 %</td>
<td>45.0 %</td>
<td></td>
</tr>
<tr>
<td>H.263+</td>
<td></td>
<td>9.5 %</td>
<td>31.9 %</td>
<td></td>
</tr>
<tr>
<td>MPEG-4</td>
<td></td>
<td></td>
<td>25.9 %</td>
<td></td>
</tr>
</tbody>
</table>
Intra Coding Efficiency versus JPEG Baseline

Casting efficiency comparison using available reference software versions
- Bit-rate savings relative to JPEG Baseline for JPEG test images

BD-rate performance relative to JPEG

[Nguyen, et. al., 2012]
Subjective Quality: Original \((1024 \times 704 \text{ samples, } 4:2:0)\)
Subjective Quality (1:50 Compression): H.262 | MPEG-2 Video
Subjective Quality (1:50 Compression): H.263+
Subjective Quality (1:50 Compression): H.264/AVC
Subjective Quality (1:50 Compression): H.265/HEVC
Summary of Intra-Picture Coding in H.265 | MPEG-H HEVC

Intra-picture coding in H.265 | MPEG-H HEVC

- Partitioning of a picture
  - Partitioning into fixed-size coding tree blocks (typically $64 \times 64$ luma samples)
  - Quadtree-based partitioning into coding blocks and transform blocks
- Spatial intra prediction
  - 35 spatial intra prediction modes (35 directional modes)
  - Intra prediction is applied on transform blocks basis
- Transform and quantization
  - Integer approximation of DCT (special case: DST)
  - Supported transform sizes: $4 \times 4$, $8 \times 8$, $16 \times 16$, $32 \times 32$
  - Scalar quantizer with uniformly distributed reconstruction levels
- Entropy coding of transform coefficient levels
  - Coded block flags
  - Significance map: Last significant coefficient, significance flag for $4 \times 4$ sub-blocks, significance flags for levels
  - Absolute levels and signs
- In-loop filter: Deblocking and SAO filter