Video Coding and HEVC

Dr. Dan Grois, E-mail: grois@ieee.org
Tutorial Agenda

Part I: Introduction to block based hybrid coding

Part II: Brief Overview of H.264/MPEG-4 AVC

Part III: HEVC version 1 (and version 2)

Part IV: Future Video Coding Development
Part I: Introduction to block-based hybrid video coding

Video Compression Goal:

- Efficiently condense visual data, while
- Minimizing the loss of visual quality due to the compression.

Why do we need to compress video?

- To reduce the storage space;
- To reduce the transmission/delivery bit-rate...
Video Coding, Transmission, and Applications

Sender

HD Video uncoded
~600 MBit/s
~100%

Channel

Bitstream
~7 MBit/s
~1.2%

Receiver

HD Video decoded
~600 MBit/s
~100%

[Grois2015]
Hybrid Video Coding (I)

Block based hybrid coding:
- Division into blocks
- Intra/Inter prediction
- Transform coding

[Grois2015]
Hybrid Video Coding: Motion Compensation

Current picture

Previous picture

[Grois2015]
Hybrid Video Coding: Residual Processing

Current picture

Previous picture

Residual (difference) picture after motion compensation

[Grois2015]
Hybrid Video Coding: Residual Processing

Things can be more complicated...

Current picture  Previous picture  Residual picture  Another previous picture

[Grois 2015]
Transform and Quantization of the Residual Picture

Goals:

- Adjust bit rate vs. fidelity;
- Remove subjectively irrelevant details.

- **Apply transform** (e.g., DCT) for converting the spatial domain pixels into transform domain coefficients;
- **Apply quantization** (e.g., scalar) for reducing a number of levels for transformed coefficients.

[Grois2015]
Standardization organizations

Video Coding Experts Group (VCEG)

- Formally structured as Question 6 of ITU-T Study Group 16 (Q.6/SG16)
- Informal nickname, VCEG, originated in 1998
- History spans from the development of first standards for digital video coding (ITU-T H.120 and H.261) through recent development of HEVC (ITU-T H.265/MPEG-H Part 2) and its extensions
- Also responsible for still-picture coding in partnership with ISO/IEC JTC 1/SC 29/WG 1 (JPEG).

[Grois2015]
Standardization organizations

Moving Pictures Experts Group (MPEG)

- Formally ISO/IEC JTC 1/SC 29/WG 11 – Coding of moving pictures and audio (ISO/IEC Joint Technical Committee 1, Subcommittee 29, Working Group 11)
  - Established in 1988
  - Mission to develop standards for coded representation of digital audio and video and related data
  - Several sub-groups:
    - Requirements
    - Systems
    - Video
    - Audio
    - 3D Graphics Compression
    - Test, Communication

[Grois2015]
H.264/MPEG-4 Advanced Video Coding (AVC)
Aug. 1999
- First test model (TML-1) of H.26L in VCEG

Dec. 2001
- Formation of Joint Video Team (JVT) between VCEG and MPEG
- JVT Chairs: G. J. Sullivan, A. Luthra, and T. Wiegand

May 2003
- ITU-T SG16 Recommendation H.264 approved
- International Standard ISO/IEC 14496-10

until April 2004
- Extensions Project: Fidelity range extensions (FRExt)
H.264/MPEG-4 AVC Common Profiles

H.264/MPEG-4 AVC Summary

- Based on **hybrid video coding** and similar in its virtue to other standards but with several improvements.

- **Some new key aspects are:**
  - Enhanced motion compensation;
  - Small blocks for transform coding;
  - Improved de-blocking filter;
  - Enhanced entropy coding.

- Significant **bit-rate savings of about 50% compared** to H.262/MPEG-2 for the same perceptual quality (especially, for higher-latency applications allowing B pictures).

[Grois2015]
H.264/AVC Basic Coding Structure

Input Video Signal

Split into Macroblocks 16x16 luma pels

- Coder Control
- Transform/Scal./Quant.
- Scaling & Inv. Transform
- Deblocking Filter
- Intra-frame Prediction
- Motion-Compensation
- Motion Estimation

Decoder

Output Video Signal

Control Data
Quant. Transf. coeffs

[Grois2015]
Common Elements with Previous Standards

- Macroblocks: 16x16 luma + 2 x 8x8 chroma samples;
- Input: Association of luma and chroma and conventional sub-sampling of chroma (4:2:0);
- Block motion displacement;
- Motion vectors over picture boundaries;
- Variable block-size motion;
- Block transforms;
- Scalar quantization;
- I, P, and B coding types.

[Grois2015]
Macroblocks and partitions

Input Video Signal

Split into Macroblocks 16x16 luma pels

16x16 luma pels

Coder Control

Transform/Scal./Quant.

Decoder

Scaling & Inv. Transform

Entrophy Coding

Control

Data

Quant. Transf. Coeffs

Intra-frame Prediction

Motion-Compensation

Motion Estimation

Intra/Inter

Motion vector accuracy 1/4 (6-tap filter)

16x16 16x8 8x16 8x8

0 0 0 0

1 1 1 1

2 3 2 3

8x8 8x4 4x8 4x4

0 0 0 0

1 1 1 1

2 3 2 3

[Grois2015]
Multiple Reference Frames and Generalized Bi-Predictive Frames

- Extend motion vector by reference picture index $\Delta$.
- Provide reference pictures at decoder side.
- In case of bi-predictive pictures: decode 2 sets of motion parameters.

Can jointly exploit scene cuts, aliasing, uncovered background and other effects with one approach

[Grois2015]
Weighted Prediction

In addition to shifting in spatial position, and selecting from among multiple reference pictures, each region’s prediction sample values can be:

- Multiplied by a weight; and
- Given an additive offset.

Some key uses:

- Improved efficiency for B coding, e.g.: accelerating motion, multiple non-reference B temporally between reference pics.

[Grois2015]
Intra Prediction

- Directional spatial prediction (9 types for luma, 1 chroma)

- Input Video Signal
- Split into Macroblocks 16x16 luma pels
- Coder Control
- Transform/Scal./Quant.
- Decoder
- Scaling/Transf.
- De-block Filter
- Intra-frame Prediction
- Motion-Compensation
- Motion Estimation
- De-blocking Filter

[Grois2015]
Transform Coding

- **4x4 Block Integer Transform**
  \[ H = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \]

- Repeated transform of DC coeffs for 8x8 chroma and some 16x16 Intra luma blocks

[Grois2015]
Entropy Coding

Input Video Signal

Split into Macroblocks 16x16 luma pels

Coder Control

Transform/Scal./Quant.

Decoder

Inv. Scal. & Transform

De-blocking Filter

Intra-frame Prediction

Motion-Compensation

Motion Estimation

Control Data

Quant. Transf. coeffs

Output Video Signal

Motion Data

[Grois2015]
Binary Coding: CAVLC

Context-Adaptive Variable Length Coding (CAVLC)

- Exp-Golomb code for all symbols except for transform coefficients
- Context adaptive VLCs for coding of transform coefficients
  - No end-of-block, but number of coefficients is decoded;
  - Coefficients are scanned backwards;
  - Contexts are built dependent on transform coefficients.

[Grois2015]
Binary Coding: CABAC

Context-based Adaptive Binary Arithmetic Coding (CABAC)

- Usage of adaptive probability models for most symbols;
- Exploiting symbol correlations by using contexts;
- Restriction to binary arithmetic coding:
  - Simple and fast adaptation mechanism;
  - Fast binary arithmetic codec based on table look-ups and shifts only.

[Grois2015]
Part II: High Efficiency Video Coding Version 1

@ Prof. Masayuki Nakajima, Uppsala University
A Guide to the H.265/HEVC Standard and its Extensions

- Cambridge University Press.
- Printed Fully In Color.
- Includes HEVC Extensions.
- Coming in 2019.
Motivation for Improved Video Compression

History of Video Coding Standards

Jevons Paradox

"The efficiency with which a resource is used tends to increase (rather than decrease) the rate of consumption of that resource."

[William Stanley Jevons]

[Wikipedia, Bross2016]
Motivation for Improved Video Compression

- IP video traffic will be 82% of all consumer Internet traffic by 2021 [Cisco2017].

- For enabling High-Quality video services, efficient compression techniques are required, especially for 3840x2160 (4K) or 7680x4320 (8K) resolutions.

http://www.theexpgroup.com

http://moneytipcentral.com
H.265/HEVC – Applications

- Internet streaming, download and play
- Real-time conversational services
- Broadcast
- Mobile streaming, conversational services and broadcast
- Content production and distribution
- Home and Digital Cinema
- Camcorders
- Medical imaging
- Remote video surveillance
- Storage media (e.g., disks, digital video tape recorder)
- Wireless display
HEVC and the JCT-VC Partnership

- ITU-T VCEG and ISO/IEC MPEG established Joint Collaborative Team on Video Coding (JCT-VC) and issued joint call for proposals (CfP) on video coding technology in 2010.

- As a result, there was an intensive development of the so-called High-Efficiency Video Coding (HEVC) standard during the next two and a half years.
  - 2013: HEVC version 1;
  - 2014: HEVC version 2 – Range Extensions (RExt), Scalable Extensions (SHVC), Multiview Extensions (MV-HEVC);
  - 2015: HEVC version 3 – 3D Video Coding Extensions (3D-HEVC);
  - 2016: HEVC version 4 – Screen Content Coding Extensions (HEVC-SCC);
  - 2018: HEVC version 5 – additional SEI messages that include omnidirectional video SEI messages, a Monochrome 10 profile, a Main 10 Still Picture profile.
Scope of Video Coding Standardization

Only restrictions on bitstream (syntax & semantics), and decoding process are standardized:

- Permits optimization beyond the obvious;
- Permits complexity reduction for implementations;
- Provides no guarantees of quality.

[Grois2015]
History: Call for Proposals Testing

- **27 complete proposals** submitted (some multi-organizational);
- Each proposal was a **major package** – lots of encoded video, extensive documentation, extensive performance metric submissions, sometimes software, etc;
- **Extensive subjective testing** (3 test labs, 4200 video clips evaluated, 850 human subjects, 300,000 scores);
- Quality of the proposal video was **compared to H.264/MPEG-AVC** anchor encodings;
- In a number of cases, **comparable quality at half bit rate**;
- Test report issued as document JCTVC-A204.

[Grois2015]
History: Call for Proposals Results

All proposals basically conceptually similar to H.264 / AVC (and prior standards):

- Block-based with variable block sizes;
- Block motion compensation;
- Fractional-pel motion vectors;
- Spatial intra prediction;
- Spatial transform of residual difference;
- Integer-based transform designs;
- Arithmetic or VLC-based entropy coding;
- Various methods of in-loop filtering to form final decoded picture and improve prediction.

"Test Model under Consideration" was set up from common design elements of well-performing proposals (Document JCTVC-A205).

[Grois2015]
History: JCT-VC Meetings

April 2010 to January 2013 – 12 JCT-VC Meetings

First Test Model under Consideration (TMuC)  
First Working Draft and Test Model (Draft 1 / HM1)  
ISO/IEC CD (Draft 6 / HM6)  
ISO/IEC DIS (Draft 8 / HM8)  
ISO/IEC FDIS & ITU-T Consent (Draft 10 / HM10)

- HEVC version 1 was officially finalized in January 2013.

[Grois2015]
Design Overview

- High-level design similar to H.264/MPEG-4 AVC:

Network abstraction layer (NAL):
- Packet oriented payload
- NAL payload types for parameter sets (header information) and data of coding layer
- Pictures structured into slices, with additional features supporting parallel processing (tiles, wavefront) and improved packetization granularity (dependent slices)

Video coding layer (VCL):
- Coded representation of the picture samples
- Hybrid video coding

[Grois 2015]
Coding-Layer Design

Input Video Signal

Subdivision into Blocks

Motion Estimation

Intra-Picture Prediction

Inter-Picture Prediction

Transform, Scaling & Quantization

Entropy Coding

Quant. Transf. Coeffs.

Output Bitstream

Output Video Signal

[Grois2015]
Block Structures

[Diagram showing block structures in video compression, including Coding Blocks, Transform Blocks, Intra-Picture Prediction, Inter-Picture Prediction, Motion Estimation, In-Loop Filter, Transform, Scaling & Quantization, Entropy Coding, and Bitstream output.]

[Grois2015]
Block Structures (Cont.)

Variable Block sizes starting from grid of Coding Tree Units (CTU):

- Consisting of Coding Tree Blocks (CTB) with fixed sizes for luma and chroma;
- Luma CTB size typically 64x64 but can be set to 32x32 and 16x16 (size signaled in SPS).
- Processed in raster scan order.

<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>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<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>
<td>21</td>
</tr>
<tr>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
<td>29</td>
<td>30</td>
<td>31</td>
<td>32</td>
</tr>
<tr>
<td>33</td>
<td>34</td>
<td>35</td>
<td>36</td>
<td>37</td>
<td>38</td>
<td>39</td>
<td>40</td>
<td>41</td>
<td>42</td>
<td>43</td>
</tr>
<tr>
<td>44</td>
<td>45</td>
<td>46</td>
<td>47</td>
<td>48</td>
<td>49</td>
<td>50</td>
<td>51</td>
<td>52</td>
<td>53</td>
<td>54</td>
</tr>
<tr>
<td>55</td>
<td>56</td>
<td>57</td>
<td>58</td>
<td>59</td>
<td>60</td>
<td>61</td>
<td>62</td>
<td>63</td>
<td>64</td>
<td>65</td>
</tr>
<tr>
<td>66</td>
<td>67</td>
<td>68</td>
<td>69</td>
<td>70</td>
<td>71</td>
<td>72</td>
<td>73</td>
<td>74</td>
<td>75</td>
<td>76</td>
</tr>
<tr>
<td>77</td>
<td>78</td>
<td>79</td>
<td>80</td>
<td>81</td>
<td>82</td>
<td>83</td>
<td>84</td>
<td>85</td>
<td>86</td>
<td>87</td>
</tr>
<tr>
<td>88</td>
<td>89</td>
<td>90</td>
<td>91</td>
<td>92</td>
<td>93</td>
<td>94</td>
<td>95</td>
<td>96</td>
<td>97</td>
<td>98</td>
</tr>
</tbody>
</table>

[Grois2015]
Block Structures (Cont.)

Coding Tree Block is the root of the Coding Quadtree:

- A Coding Block leave can be predicted using Inter- or Intra prediction;
- The coding quadtree is traversed in Z-Scan Order

[Grois2015]
Block Structures (Cont.)

Coding Quadtrees:

Simple scheme to locally adapt block sizes

Large block sizes especially for HD resolutions and beyond

4K sequence Traffic (cropped to 2560x1600)

Detail showing coding quadtree with Coding Blocks

[Grois2015]
Block Structures (Cont.)

A **Coding Block** can be split:

- once into **Prediction Block** partitions;
- recursively into **Transform Blocks** using the Residual Quadtree (RQT): transform sizes ranging from 4x4 to 32x32.

E.g., Nx2N partition with 2 prediction blocks.

E.g., RQT with depth=3 and 10 transform blocks.

[Grois2015]
Block Structures (Cont.)

Example:

[Schwarz2014]

[Grois2015]
Different **Prediction Block** partition modes:

- Each prediction block in a coding block uses the same prediction mode (intra or inter);
- **Inter**: motion compensation is done per prediction block and prediction blocks can be “merged”;
- **Intra**: mode, e.g. DC, planar, angular,.. defined per prediction block.

[Grois2015]
RQT with variable-size Transform Block leaves:

- **Transform** of the residual is performed on transform blocks;
- **Intra prediction** is also performed on transform blocks in intra coding blocks;
- Transform sizes ranging from 4x4 to 32x32;
- **Adaptation to varying space-frequency characteristics** in the residual signal for the DCT-based integer transform.

[Grois2015]
Average bit-rate savings for successively increasing the CTU size and the number of hierarchy levels in the coding tree:

<table>
<thead>
<tr>
<th>CTU Size and Minimum CU Size</th>
<th>Entertainment Applications</th>
<th>Interactive Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>32x32 - 16x16</td>
<td>9.2%</td>
<td>17.4%</td>
</tr>
<tr>
<td>32x32 - 8x8</td>
<td>12.1%</td>
<td>20.2%</td>
</tr>
<tr>
<td>64x64 - 16x16</td>
<td>12.7%</td>
<td>23.8%</td>
</tr>
<tr>
<td>64x64 - 8x8</td>
<td>14.9%</td>
<td>25.5%</td>
</tr>
</tbody>
</table>

Anchor: 16x16 CTU size and a minimum CU size of 8x8 luma samples.

[Schwarz2014]

[Grois2015]
Coding efficiency improvement for successively increasing the maximum TU size:

<table>
<thead>
<tr>
<th>Maximum TU Size</th>
<th>Entertainment Applications</th>
<th>Interactive Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum TU Size of 8x8</td>
<td>6.8%</td>
<td>8.5%</td>
</tr>
<tr>
<td>Maximum TU Size of 16x16</td>
<td>11.9%</td>
<td>14.7%</td>
</tr>
<tr>
<td>Maximum TU Size of 32x32</td>
<td>13.9%</td>
<td>17.5%</td>
</tr>
</tbody>
</table>

[Schwarz2014]

Anchor: 4x4 TUs, 64x64 CTU, all CU and PU sizes

[Grois2015]
Block Structures (Cont.)

Average bit-rate savings for successively enabling various PU sizes:

<table>
<thead>
<tr>
<th>Enabled PU Sizes</th>
<th>Entertainment Applications</th>
<th>Interactive Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>16x16 and 8x8</td>
<td>2.7%</td>
<td>4.4%</td>
</tr>
<tr>
<td>Square PUs from 4x4 to 16x16</td>
<td>6.1%</td>
<td>5.6%</td>
</tr>
<tr>
<td>Square PUs from 4x4 to 32x32</td>
<td>15.4%</td>
<td>23.0%</td>
</tr>
<tr>
<td>Square PUs from 4x4 to 64x64</td>
<td>18.7%</td>
<td>30.3%</td>
</tr>
<tr>
<td>All modes except asym. (+4x4 PUs)</td>
<td>20.0%</td>
<td>31.0%</td>
</tr>
<tr>
<td>All HEVC PU Sizes (+4x4 PUs)</td>
<td>20.7%</td>
<td>33.0%</td>
</tr>
</tbody>
</table>

Anchor: 16x16 PUs and 4x4 TUs

[Schwarz2014]

[Grois2015]
Intra-Picture Prediction
Intra-Picture Prediction: Main Tools

H.264/AVC intra-picture prediction drawbacks:

- **Insufficient range of supported coding block sizes**: especially high-resolution videos (HD or UHD) cannot be coded properly due to the poor representation of some picture textures.

- **Insufficient number of intra prediction directions**: especially for larger block sizes.

- **Insufficient prediction of homogeneous regions**: visible artifacts introduced by discontinuities at block boundaries.

- **Inconsistency across block sizes**: depending on the size of the block, H.264/AVC uses different methods for predicting a block and the color component, which is represented by the block; it turn, this can be especially critical for a large variety of block sizes.

[Grois2015]
HEVC intra-picture prediction addresses H.264/AVC drawbacks by:

- Reference samples computed with **1/32 pel accuracy** (bilinear interpolation);
- Angular prediction with **33 directional modes** as well as Planar and DC modes;
- **Reference sample smoothing** for diagonal directions and planar mode;
- **Boundary smoothing** (across block boundary) for the horizontal, vertical and DC modes;
- Prediction modes **unified across all block sizes.**
Three prediction types:

- Planar prediction (0),
- DC prediction (1),
- 33 angular prediction directions (2-34).

[Grois2015]
Only 5 modes for chroma can be signaled:

- Planar (0);
- DC (1);
- Horizontal (10);
- Vertical (26);
- Same directional mode as luma.

[Grois2015]
Intra Prediction: Reference Sample Smoothing

HEVC adaptive smoothing depends on:

- **Block size**;
- **Directionality**;
- **Amount of detected discontinuity**.

- Two smoothing filters:
  - **Normal filtering**;
  - **Strong filtering**.

- Applied to each reference sample using the neighboring reference samples.

[Grois2015]
Reference Sample Smoothing depends on:

- **Block size**;
- **Intra direction**;
- Discontinuity by comparing minimal distance of direction to horizontal and vertical mode with predefined thresholds

[Grois2015]
Discontinuities along block boundaries may be introduced by applying intra-prediction modes.

- Hence, HEVC employs boundary sample smoothing for:
  - Luma TBs;
  - TB sizes < 32x32;
  - Horizontal prediction mode, applied to the first row;
  - Vertical prediction mode, applied to the first column;
  - DC intra-prediction mode, both the first row and column of samples in the TB are filtered.

- Prediction for chroma components tends to be very smooth - no boundary smoothing for chroma TBs to reduce the computational complexity

[Grois2015]
Inter-Picture Prediction

[Diagram showing the process of video signal encoding and decoding, including steps like input video signal, motion estimation, inter-picture prediction, transform, scaling & quantization, entropy coding, output bitstream, and output video signal.]
Inter-Picture Prediction: Main Tools

HEVC improves motion compensated prediction by:

- Asymmetric Motion Partitions (AMP) as explained in Block Structures;
- Reference Picture Sets (RPS) for simplified decoded picture buffer management;
- Improved interpolation filters for luma and chroma;
- More efficient motion data coding including
  - Advanced Motion Vector Prediction (AMVP);
  - Inter-picture prediction block merging (Merge);
  - Motion Data Storage Reduction (MDSR);
- Simplified weighted sample prediction.
- No 4x4 blocks and only uni-prediction for 4x8/8x4 to reduce memory bandwidth.

[Grois2015]
Transform and Quantization

Transform, Scaling & Quantization

Decoder

Intra-Picture Prediction

Inter-Picture Prediction

Motion Estimation

Scaling & Inverse Transform

In-Loop Filter

Entropy Coding

Output

010110...

Bitstream

Input

Video Signal

Output

Video Signal

[Grois2015]
Transform and Quantization

- **Prediction residual is transformed** (unless signaled as zero by coded block flag, skip mode or alternative bypass modes);

- Core transforms 4x4 ... 32x32 are **integer approximations** of DCT:
  - Close to orthogonality;
  - Nearly identical norms of all basis vectors: Unique scaling factor for each transform size;
  - Matrices of smaller-size transforms are sub-sampled versions of length-32 transform matrix.

- **2D transforms separable** (except for the rounding after each transform step), only square transforms used.

[Grois2015]
Transform and Quantization (Cont.)

- Example: length-16 transform matrix (Euclid. norm approx. 256):

\[
H = \begin{bmatrix}
64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 \\
90 & 87 & 80 & 70 & 57 & 43 & 25 & 9 \\
89 & 75 & 50 & 18 & -18 & -50 & -75 & -89 \\
87 & 57 & 9 & -43 & -80 & -90 & -70 & -25 \\
83 & 36 & -36 & -83 & -83 & -36 & 36 & 83 \\
80 & 9 & -70 & -87 & -25 & 57 & 90 & 43 \\
75 & -18 & -89 & -50 & 50 & 89 & 18 & -75 \\
70 & -43 & -87 & 9 & 90 & 25 & -80 & -57 \\
64 & -64 & -64 & 64 & 64 & -64 & -64 & 64 \\
57 & -80 & -25 & 90 & -9 & -87 & 43 & 70 \\
50 & -89 & 18 & 75 & -75 & -18 & 89 & -50 \\
43 & -90 & 57 & 25 & -87 & 70 & 9 & -80 \\
36 & -83 & 83 & -36 & -36 & 83 & -83 & 36 \\
25 & -70 & 90 & -80 & 43 & 9 & -57 & 87 \\
18 & -50 & 75 & -89 & 89 & -75 & 50 & -18 \\
9 & -25 & 43 & -57 & 70 & -80 & 87 & -90 \\
\end{bmatrix}
\]

8x8

[Grois2015]
Transform and Quantization (Cont.)

- Example: length-16 transform matrix (Euclid. norm approx. 256):

\[
H = \begin{bmatrix}
64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 \\
90 & 87 & 80 & 70 & 57 & 43 & 25 & 9 & -9 & -25 & -43 & -57 & -70 & -80 & -87 & 90 \\
89 & 75 & 50 & 18 & -18 & -50 & -75 & -89 & -89 & -75 & -50 & -18 & 18 & 50 & 75 & 89 \\
75 & -18 & -89 & -50 & 50 & 89 & 18 & -75 & -75 & 18 & 89 & 50 & -50 & -89 & -18 & 75 \\
64 & -64 & -64 & 64 & 64 & -64 & -64 & -64 & -64 & 64 & 64 & -64 & -64 & -64 & 64 & 64 \\
50 & -89 & 18 & 75 & -75 & -18 & 89 & -50 & -50 & 89 & -18 & -75 & 75 & 18 & -89 & 50 \\
18 & -50 & 75 & -89 & 89 & -75 & 50 & -18 & -18 & 50 & -75 & 89 & -89 & 90 & -75 & 50 & 18 \\
\end{bmatrix}
\]

[Grois2015]
Transform and Quantization (Cont.)

Integer approximation of **Discrete Sine Transform (DST)** is applied to 4x4 intra residual:

- In intra prediction (in particular directional and planar), the prediction error increases with larger distance from the boundary;
- DST basis vectors are better fitting such behavior;
- For **intra residual of 4x4 PBs**, the following transform matrix is used (Euclid. norm approx. 128, same as length-4 DCT (*the Euclidean norm is the square root of the sum of all the squares*)).

\[
H = \begin{bmatrix}
29 & 55 & 74 & 84 \\
74 & 74 & 0 & -74 \\
84 & -29 & -74 & 55 \\
55 & -84 & 74 & -29 \\
\end{bmatrix}
\]

- **Provides a 1% bit-rate reduction** but was restricted to Intra 4 × 4 luma transform blocks due to marginal benefits for the other transform cases.

[Grois2015]
Transform and Quantization (Cont.)

- **Quantization similar to H.264/MPEG-AVC** (QP+6 = double step size, QP values are defined from 0 to 51).
- The QP to Qstep mapping has a reduction of about 12.5% in the bitrate for the increase of 1 in QP.

- **CU level adaption** of quantizer step size.
- Coding of QP performed differentially within CTU.

- **Frequency weighted quantization** possible via quantization matrices (encoded in SPS or PPS):
  - For the larger transforms (16x16, 32x32) a subsampled quantization matrix of size 8x8 is used, i.e. same quantization step sizes used for groups of 2x2 or 4x4 adjacent coefficients);
  - Different quantization matrices for intra and inter, Y/Cb/Cr;
  - Matrix entries coded differentially.

[Grois2015]
In-loop Filters

[Diagram of video signal processing flow]

Input Video Signal → Decoder → Intra-Picture Prediction → In-Loop Filter → Inter-Picture Prediction → Motion Estimation → Transform, Scaling & Quantization → Entropy Coding → Output Bitstream

[Grois2015]
In-loop Filters

Two filters to remove coding artifacts and preserve edges applied before writing the output to reference memory (not effective in intra prediction):

- **Deblocking Filter** (similar to H.264/MPEG-4 AVC, but having a reduced number of filtering strengths and better parallelization) – operated only on 8x8 block boundaries (not 4x4) with 4-sample units.

- **Sample Adaptive Offset (SAO)** - two different offset methods:
  - **Individual mapping with *edge offset values*** applied to samples based on classification as an edge or non-edge.
  - **Non-linear amplitude mapping function with *band offset values*** applied to samples based on their values lying in one of four consecutive samples value ranges (bands).

[Grois2015]
Deblocking Filter

Generally, when designing a deblocking filter, several issues should be carefully considered:

- Filter has a **direct impact on the subjective picture quality.**

- The filtering decisions contain **challenges that lead to the following main questions:**
  
  - Is the block boundary **natural edge or artifact (filter off/on)?**
  
  - Which **filtering strength** should be used?

[Grois2015]
The deblocking is performed on a **four-sample portion of a 8x8 block boundary**, when the following conditions are held:

- the block boundary is a prediction unit or transform unit boundary;
- the block boundary strength, which is determined according to a predefined criteria, is greater than zero;
- variation of signal on both sides of a block boundary is below a specific threshold.

[Grois2015]
Deblocking Filter (Cont.)

Splitting a picture on 8x8. The dotted-lines represents non-overlapping blocks of 8x8 samples: these non-overlapping blocks can be deblocked in a parallel manner → better parallelization.

[Grois2015]
Deblocking Filter (Cont.)

• BS values for the boundary between two neighbor luma blocks

• 2 – Strong Filtering, 1 – Weak Filtering, 0 – No Filtering:

<table>
<thead>
<tr>
<th>No.</th>
<th>Conditions for Applying a Deblocking Filter</th>
<th>BS Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>At least one of the blocks P or Q is intra predicted</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>At least one of the blocks P or Q has non-zero coded residual coefficient and boundary is a transform boundary</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>Absolute differences between corresponding spatial motion vector components of the two blocks P and Q are $\geq 1$ in units of integer pixels</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>Motion-compensated prediction for the two blocks P and Q refers to different reference pictures or the number of motion vectors is different for the two blocks</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>Otherwise</td>
<td>0</td>
</tr>
</tbody>
</table>

[Grois2015]
Deblocking: Weak/Strong Filtering

- The **strong filtering** is applied:
  - on three samples on both sides of the block boundary;
  - within the four-sample segment;
  - similar to H.264/MPEG-4 AVC.

- The **weak filtering** is applied:
  - on two samples on both sides of the block boundary;
  - within the four-sample segment.

[Grois2015]
The deblocking process in HEVC provides **improved parallelization**, compared to H.264/MPEG-4 AVC:

- the deblocking process of **non-overlapping 8x8 samples** can be fully parallelized;
- any **vertical block edge** can be deblocked in parallel with the deblocking of any other vertical block edge;
- any **horizontal block edge** can be deblocked in parallel with the deblocking of any other horizontal block edge;
- the updated sample values after the deblocking of vertical block boundaries are used as input for filtering the horizontal block boundaries.

[Grois2015]
Sample Adaptive Offset: Motivation

- Sample Adaptive Offset filter is designed to add corrective offset values for attenuating:
  - **Systematic Errors** introduced by quantization and phase shifts from inaccurate motion vectors;
  - **Ringing Artefacts (Gibbs Phenomenon)**, introduced mainly by large transform sizes.

- Two different offset methods can be used:
  - **Non-linear amplitude mapping function** with band offset values applied to samples based on their values lying in one of four consecutive samples value ranges (bands) out of 32;
  - **Individual mapping with edge offset values** applied to samples based on classification as edge or non-edge, whereas the gradient orientation is determined by the encoder.

[Grois2015]
Entropy Coding

Input Video Signal → Decoder

- Intra-Picture Prediction
- Inter-Picture Prediction
- Motion Estimation

→ Transform, Scaling & Quantization

→ Quant. Transf. Coeffs.

→ Output Bitstream

[Grois2015]
Context-based Adaptive Binary Arithmetic Coding

Single entropy coding scheme in HEVC using Context-based Adaptive Binary Arithmetic Coding (CABAC) instead of two in H.264/AVC:

- Usage of adaptive probability models for most symbols;
- Exploiting symbol correlations by using contexts;
- Restriction to binary arithmetic coding based on table look-ups and shifts only;
- Main improvements, compared to H.264/AVC:
  - Reduced number of contexts;
  - Reduced number of context coded binary decisions to avoid throughput bottleneck;
  - Removed dependencies to facilitate parallel context derivation;
  - Truncated Rice Codes added as binarization.

[Grois2015]
Entropy Coding of Transform Coefficients

- Sub-blocks and the 16 coefficients in a sub-block are processed using a reverse **diagonal scan**.

- Intra 4x4 TBs and 8x8 luma TBs use a **horizontal scan** and a **vertical scan**.

[Grois2015]
Alternative Coding Modes (I)

Three special coding tools give support for special cases (without any significant implementation cost).

- Transform bypass
- Lossless bypass
- Intra PCM Mode

[Grois2015]
**Alternative Coding Modes (II)**

**Transform bypass:**

- Signaled at **TB level** (transform_skip_flag);
- Useful particularly for graphics content (**sharp edges, flat areas**), e.g. as in **screen content coding applications**;
- CABAC contexts of transform coding are directly applied to prediction error signal (**becoming spatial contexts to close neighbor samples**).

**Bypass enabling "lossless" coding:**

- Signaled at **CU level** (cu_transquant_bypass_flag);
- **Bypass of scaling, transform and in-loop filters**;
- CABAC entropy coding still applied as is;
- Can also be used locally, e.g. for graphics/text

[Grois2015]
Alternative Coding Modes (III)

Intra PCM mode:

- Signaled at **CU level** (pcm_flag);
- **Direct coding of sample values**;
- Is intended for "emergency" situations when encoder:
  - faces complicated content and would produce more bit rate than uncoded samples (e.g., in case of the uncorrelated noise);
  - would overrun its processing capabilities.

[Grois2015]
Alternative Coding Modes (IV)

HEVC Bitstream

Transform Quantization Bypass ("lossless") Mode

Transform Skip Mode

Inverse Transform

Intra/Inter-Prediction

Reconstructed Block

Entropy Decoding

De-quantization

Intra PCM Mode

[Grois2015]
High-Level Syntax

• In general, all syntax elements above the slice segment data layer are called high-level syntax.

• High-level syntax:
  ▪ Access to packets
  ▪ Settings of low level coding tools
  ▪ Random-access information
  ▪ Metadata

[Grois2015]
High-Level Syntax: Parameter Sets

- **Sequence Parameter Set (SPS) and Picture Parameter Set (PPS)**
  similar as H.264/MPEG-AVC

- **New Video Parameter Set (VPS)**
  - Describes overall characteristics of coded video sequences, including the dependencies between layers.
  - Enables compatible extensibility in terms of signaling at the systems layer.

[Grois2015]
High-Level Syntax: VPS

The (HEVC version 1) Video Parameter Set contains:

- Maximum number of **layers** in the bitstream and the highest layer ID
- Maximum number of **temporal sub-layers**
- **Profile/Tier/Level** (for each temporal sub-layer)
- DPB parameters (optional for temporal sub-layers)
- Layer sets (which can be extracted as sub-bitstreams)
- Timing (optional)
- HRD parameters (optional)

[Grois2015]
High-Level Syntax: VPS (Cont.)

The Video Parameter Set extension (HEVC version 2) contains:

- Scalability types
  - Scalable
  - Multi-View
  - Depth (v3)
  - Auxiliary pictures
- Dependencies between layers
- Output layer sets
- Representation Format
- DPB information
- VPS VUI

[Grois2015]
High-Level Syntax: SEI and VUI

Supplemental Enhancement Information (SEI) and Video Usability Information (VUI):

- SEI and VUI concepts similar to H.264/MPEG-AVC
  - no impact on normative decoder behavior
  - usage optional
- Example SEI messages: Buffering Information, Picture Timing, Frame packing (for stereo), display orientation, interlaced-scan indicators (frame/field arrangement);
- Example VUI parameters: Hypothetical Reference Decoder (HRD) and clock parameters, color space, aspect ratio, constraint flags.

[Grois2015]
Parallel Picture Processing

- Slices / Slice segments
- Tiles
- Wavefront Parallel Processing (WPP)

[Grois2015]
Parallelization Tools: Slices

**Slice coding:**
- Slices can be coded independent of each other.
- In-loop filters can be disabled at slice boundaries.

**Dependent slice segments**
- Inherit the previous entropy coding state

[Grois2015]
Parallelization Tools: Tiles

Tiles are:

- Independent rectangular regions of the picture
- In a regular pattern (division by horizontal and vertical lines within the picture)

|   | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 | 71 | 72 | 73 | 74 | 75 | 76 |

Processing of tiles

- In raster scan order over the tiles
- and CTU-wise in raster scan order within the tiles.

[Grois2015]
Parallelization Tools: Wavefronts

Wavefront processing:

- Allows to run several processing threads in a slice over rows of CTUs with a delay that allows adaptation;
- CABAC contexts are inherited from a row above

[Grois2015]
Parallelization Tools: Performance

Total coding losses (using weighted BD-Rate) of different tile configurations for 1080 videos.

[Chi2012]  [Grois2015]
## Comparison of Parallelization Approaches

<table>
<thead>
<tr>
<th>Properties</th>
<th>Slices</th>
<th>Tiles</th>
<th>WPP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coding loss</td>
<td>Very high</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>Boundary artifacts</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Single-core issues</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Parallel scalability</td>
<td>Medium</td>
<td>Medium</td>
<td>Medium/high</td>
</tr>
<tr>
<td>Region of interest</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

[Chi2012]
Profiles, Levels and Tiers

Onion-like structure

Main Still Picture
- Intra coding tools,
- Wavefront parallelism only when multiple tiles in a picture are not used
- Minimum luma tile size 256x64
- 4:2:0 chroma sampling
- 8 bit sample bit depth

Main
+ Inter coding tools
+ Timing information / constraints

Main 10
+ 8-10 bit sample bit depth

[Grois2015]
Rate-Distortion Performance

Entertainment Applications (Random Access)

Interactive Applications (Low Delay)

[Ohm2012]
Average bit-rate savings for the HEVC Main Still Picture Profile:

<table>
<thead>
<tr>
<th>Coding Standard</th>
<th>Y'</th>
<th>Y'CbCr</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVC</td>
<td>17%</td>
<td>17%</td>
</tr>
<tr>
<td>VP8</td>
<td>23%</td>
<td>24%</td>
</tr>
<tr>
<td>VP9</td>
<td>11%</td>
<td>10%</td>
</tr>
<tr>
<td>JPEG 2000</td>
<td>22%</td>
<td>23%</td>
</tr>
<tr>
<td>JPEG XR</td>
<td>32%</td>
<td>30%</td>
</tr>
<tr>
<td>WebP</td>
<td>30%</td>
<td>29%</td>
</tr>
<tr>
<td>JPEG</td>
<td>44%</td>
<td>44%</td>
</tr>
</tbody>
</table>

[Nguyen2015]
Average Bit-rate Savings (Still Picture Coding)

Average bit-rate savings relative to JPEG:

[Nguyen2015]
Subjective Verification Tests of HEVC vs. AVC

Verification test completed in April, 2014 (JCTVC-Q1011):

- A **subjective evaluation** was conducted comparing the HEVC Main profile to the AVC High profile;
- 20 test sequences: **480p to Ultra HD (UHD)** various bit rates/quality levels;
- Average bit-rate savings for test sequences:
  - UHD - 64%
  - 1080p - 62%
  - 720p - 56%
  - 480p - 52%
## Average Bit-Rate Savings (Random Access)

### PCS 2013:

<table>
<thead>
<tr>
<th>Sequences</th>
<th>HEVC vs. VP9 [%]</th>
<th>HEVC vs. x264 High Profile [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traffic</td>
<td>50.1</td>
<td>38.2</td>
</tr>
<tr>
<td>PeopleOnStreet</td>
<td>26.4</td>
<td>24.9</td>
</tr>
<tr>
<td>Kimono</td>
<td>33.1</td>
<td>41.2</td>
</tr>
<tr>
<td>ParkScene</td>
<td>44.9</td>
<td>32.9</td>
</tr>
<tr>
<td>Cactus</td>
<td>45.3</td>
<td>39.6</td>
</tr>
<tr>
<td>BQTerrace</td>
<td>49.3</td>
<td>47.3</td>
</tr>
<tr>
<td>Basketball Drive</td>
<td>32.1</td>
<td>45.0</td>
</tr>
<tr>
<td>FourPeople</td>
<td>47.1</td>
<td>34.2</td>
</tr>
<tr>
<td>Johnny</td>
<td>52.2</td>
<td>47.9</td>
</tr>
<tr>
<td>KristenAndSara</td>
<td>49.5</td>
<td>41.9</td>
</tr>
<tr>
<td>BasketballDrillText</td>
<td>45.4</td>
<td>43.4</td>
</tr>
<tr>
<td>ChinaSpeed</td>
<td>44.2</td>
<td>34.8</td>
</tr>
<tr>
<td>Averages</td>
<td>43.3</td>
<td>39.3</td>
</tr>
<tr>
<td><strong>Total Average</strong></td>
<td><strong>43.3</strong></td>
<td><strong>39.3</strong></td>
</tr>
</tbody>
</table>

[Grois2013]
Average Bit-Rate Savings (Random Access)

PCS 2013: Rate-Distortion plot examples

[Graphs showing PSNR-YUV and Bit-rate saving of HEVC for different traffic and ParkScene scenarios]

[Grois2013]
## Average Bit-Rate Savings (Random Access & Low Delay)

### HEVC vs. H.264/AVC High Profile (x264) and VP9

#### Random Access Configuration

<table>
<thead>
<tr>
<th></th>
<th>HEVC</th>
<th>x264</th>
<th>VP9</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCS 2013:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HEVC</td>
<td>-39.3%</td>
<td></td>
<td>-43.3%</td>
</tr>
<tr>
<td>x264</td>
<td>66.4%</td>
<td></td>
<td>-6.2%</td>
</tr>
<tr>
<td>VP9</td>
<td>79.4%</td>
<td>8.4%</td>
<td></td>
</tr>
</tbody>
</table>

[Grois2013]

#### Low Delay Configuration

<table>
<thead>
<tr>
<th></th>
<th>HEVC</th>
<th>x264</th>
<th>VP9</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPIE 2014:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HEVC</td>
<td>-40.8%</td>
<td></td>
<td>-32.5%</td>
</tr>
<tr>
<td>x264</td>
<td>73.5%</td>
<td></td>
<td>17.0%</td>
</tr>
<tr>
<td>VP9</td>
<td>48.2%</td>
<td>-12.5%</td>
<td></td>
</tr>
</tbody>
</table>

[Grois2014]
### Fixed QP Test Case

#### SPIE 2017:

**BD-BR: Weighted PSNR$_{YUV}$**

AV1 and VP9 with Rate Control Disabled

(negative BD-BR values indicate actual bit-rate savings)

<table>
<thead>
<tr>
<th>test candidate</th>
<th>anchor</th>
<th>AV1</th>
<th>JEM</th>
<th>VP9</th>
<th>HM</th>
</tr>
</thead>
<tbody>
<tr>
<td>AV1</td>
<td></td>
<td>111.8%</td>
<td>-17.1%</td>
<td>47.7%</td>
<td></td>
</tr>
<tr>
<td>JEM</td>
<td>-51.4%</td>
<td></td>
<td>-62.0%</td>
<td>-29.8%</td>
<td></td>
</tr>
<tr>
<td>VP9</td>
<td>21.0%</td>
<td>173.7%</td>
<td></td>
<td>92.5%</td>
<td></td>
</tr>
<tr>
<td>HM</td>
<td>-30.6%</td>
<td>43.4%</td>
<td></td>
<td>-46.6%</td>
<td></td>
</tr>
</tbody>
</table>

[Grois2017]
# Rate Control Test Case

### BD-BR: Weighted PSNR$_{YUV}$

AV1 and VP9 with Rate Control Enabled

(negative BD-BR values indicate actual bit-rate savings)

<table>
<thead>
<tr>
<th></th>
<th>anchor</th>
<th>AV1</th>
<th>JEM</th>
<th>VP9</th>
<th>HM</th>
</tr>
</thead>
<tbody>
<tr>
<td>test candidate</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AV1</td>
<td></td>
<td>55.0%</td>
<td></td>
<td>-20.0%</td>
<td>9.5%</td>
</tr>
<tr>
<td>JEM</td>
<td>-34.8%</td>
<td></td>
<td>-47.3%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VP9</td>
<td>28.5%</td>
<td>92.2%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HM</td>
<td>-7.8%</td>
<td></td>
<td></td>
<td>-27.0%</td>
<td></td>
</tr>
</tbody>
</table>

[Grois2017]
Fixed QP Test Case

**PCS 2018:**

**BD-BR: Weighted PSNR\(_{YUV}\)**

AV1 and VP9 with Rate Control **Disabled**

(negative BD-BR values indicate actual bit-rate savings)

<table>
<thead>
<tr>
<th>CODECS</th>
<th>JEM</th>
<th>HM</th>
<th>AV1</th>
<th>VP9</th>
</tr>
</thead>
<tbody>
<tr>
<td>JEM</td>
<td></td>
<td>-31.6%</td>
<td>-45.6%</td>
<td>-58.8%</td>
</tr>
<tr>
<td>HM</td>
<td>47.4%</td>
<td></td>
<td>-21.7%</td>
<td>-40.3%</td>
</tr>
<tr>
<td>AV1</td>
<td>89.3%</td>
<td>30.5%</td>
<td></td>
<td>-23.4%</td>
</tr>
<tr>
<td>VP9</td>
<td>154.8%</td>
<td>73.5%</td>
<td>31.1%</td>
<td></td>
</tr>
</tbody>
</table>

[Nguyen2018]
## Fixed QP Test Case

### PCS 2018:

**Encoders Run Time**

AV1 and VP9 with Rate Control Disabled

(negative BD-BR values indicate actual bit-rate savings)

<table>
<thead>
<tr>
<th>CODECS</th>
<th>JEM</th>
<th>HM</th>
<th>AV1</th>
<th>VP9</th>
</tr>
</thead>
<tbody>
<tr>
<td>JEM</td>
<td></td>
<td>8.48 X</td>
<td>0.26 X</td>
<td>15.2 X</td>
</tr>
<tr>
<td>HM</td>
<td>0.12 X</td>
<td></td>
<td>0.03 X</td>
<td>1.79 X</td>
</tr>
<tr>
<td>AV1</td>
<td>3.83 X</td>
<td>32.45 X</td>
<td></td>
<td>58.16 X</td>
</tr>
<tr>
<td>VP9</td>
<td>0.07 X</td>
<td>0.56 X</td>
<td>0.02 X</td>
<td></td>
</tr>
</tbody>
</table>

[Nguyen2018]
Rate Control Test Case

PCS 2018:

**BD-BR: Weighted PSNR\textsubscript{YUV}**

AV1 and VP9 with Rate Control Enabled

(negative BD-BR values indicate actual bit-rate savings)

<table>
<thead>
<tr>
<th>CODECS</th>
<th>JEM</th>
<th>HM</th>
<th>AV1</th>
<th>VP9</th>
</tr>
</thead>
<tbody>
<tr>
<td>JEM</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HM</td>
<td>47.4%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AV1</td>
<td>48.6%</td>
<td>2.3%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VP9</td>
<td>97.9%</td>
<td>34.5%</td>
<td>30.2%</td>
<td></td>
</tr>
</tbody>
</table>

[Nguyen2018]
# Rate Control Test Case

## PCS 2018:

**Encoders Run Times**

AV1 and VP9 with Rate Control Enabled

(negative BD-BR values indicate actual bit-rate savings)

<table>
<thead>
<tr>
<th>CODECS</th>
<th>JEM</th>
<th>HM</th>
<th>AV1</th>
<th>VP9</th>
</tr>
</thead>
<tbody>
<tr>
<td>JEM</td>
<td></td>
<td>8.48 X</td>
<td>0.40 X</td>
<td>22.58 X</td>
</tr>
<tr>
<td>HM</td>
<td>0.12 X</td>
<td></td>
<td>0.05 X</td>
<td>2.66 X</td>
</tr>
<tr>
<td>AV1</td>
<td>2.47 X</td>
<td>20.95 X</td>
<td></td>
<td>55.82 X</td>
</tr>
<tr>
<td>VP9</td>
<td>0.04 X</td>
<td>0.38 X</td>
<td>0.02 X</td>
<td></td>
</tr>
</tbody>
</table>

[Nguyen2018]
HEVC version 2

- Format Range Extensions (RExt)
- Multilayer Extensions:
  - Scalability Extension (SHVC)
  - Multiview Extension (MV-HEVC)
**RExt: Motivation**

**Extent 4:2:0 8-10 bit consumer oriented scope of HEVC version 1 by:**

- high quality distribution in broadcast, 4:2:0, 12 bit
- contribution in broadcast, 4:2:2, 10/12 bit
- production and high fidelity content acquisition, 4:4:4, 16 bit, R’G’B’, high bit rate
- medical imaging, 4:0:0 monochrome, 12-16 bit, (near) lossless
- alpha channels and depth maps, 4:0:0 monochrome, 8-bit
- high quality still pictures, 4:4:4, 8-16 bit, arbitrary picture size
- and many others...

[Grois2015]
RExt: Modifications of HEVC version 1

Three categories:

1. **Necessary modifications** to extend support for chroma formats beyond 4:2:0 and bit depth beyond 10 bits per sample.

2. **Coding efficiency improvements** for extended formats, lossless and near lossless coding by means of:
   - Modified HEVC v1 tools;
   - New tools:
     - Adaptive chroma QP offset (ACQP)
     - Cross Component Prediction (CCP)
     - Residual Delta Pulse Code Modulation (RDPCM)

3. **Precision and throughput optimizations** for very high bit rates and bit depths.

[Grois2015]
RExt (HEVC Version 2): Profiles

21 profiles added resulting from combination samples of

- Prediction type (Intra/Inter);
- Chroma format;
- Bit-depth;
- Tool set.

[Diagram showing Intra and Inter Profiles with Bit-Depth and Chroma Format combinations.]
## RExt: Profiles (Cont.)

<table>
<thead>
<tr>
<th>Still Picture</th>
<th>Intra</th>
<th>Inter</th>
</tr>
</thead>
<tbody>
<tr>
<td>4:0:0</td>
<td></td>
<td>Monochrome</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Monochrome 12</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Monochrome 16</td>
</tr>
<tr>
<td>4:2:0</td>
<td>Main Still Picture*</td>
<td>Main*</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Main 10*</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Main 12</td>
</tr>
<tr>
<td>4:2:2</td>
<td>Main 4:2:2 10 Intra</td>
<td>Main 4:2:2 10</td>
</tr>
<tr>
<td></td>
<td>Main 4:2:2 12 Intra</td>
<td>Main 4:2:2 12</td>
</tr>
<tr>
<td>4:4:4</td>
<td>Main 4:4:4 Still Picture</td>
<td>Main 4:4:4</td>
</tr>
<tr>
<td></td>
<td>Main 4:4:4 10 Intra</td>
<td>Main 4:4:4 10</td>
</tr>
<tr>
<td></td>
<td>Main 4:4:4 12 Intra</td>
<td>Main 4:4:4 12</td>
</tr>
<tr>
<td></td>
<td>Main 4:4:4 16 Still Picture</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Main 4:4:4 16 Intra</td>
<td>High Throughput 4:4:4 16 Intra</td>
</tr>
</tbody>
</table>

*HEVC v1*
SHVC: Motivation

Traditional approach:
- Encode and store a separate video stream for all possible clients and connection speeds

Scalable Video Coding:
- Encode a common base layer
- Add enhancement layers as needed

[Grois2015]
Video streaming with multiple layers

Images by Freepik.com
SHVC: Advantages and disadvantages of scalable coding

Generic advantages:
- Less storage space
- Less data rate in multi- and broadcast environments
- (Less encoding time)
- More flexibility (i.e. combinations of layers)
- Built-in support for up- and down-switching
- Error resilience: Unequal error protection

Generic disadvantages:
- Overhead compared to single layer
- Higher decoder complexity
SHVC: Single- vs. multi-loop functionality

Single-loop decoder (e.g. SVC):
1. Parse base layer
2. Parse enhancement layer, using information from base layer
3. (decode base-layer intra blocks)
4. Decode enhancement layer

One decoding process overall

Multi-loop decoder (e.g. SHVC):
1. Decode base layer
2. Upsample result as reference signal
3. Decoder enhancement layer

One decoding process per layer

[Grois2015]
SHVC: Properties of single- vs. multi-loop

Single-loop decoder (e.g. SVC):
- Low decoding complexity
- Always requires changes in low-level coding tools
- Minimal re-use of existing components

Multi-loop decoder (e.g. SHVC):
- Higher decoding complexity
- May not need changes in low-level coding tools
- Allows re-use of existing single-layer designs as base for each layer-decoder

[Grois2015]
SHVC: Scalability types (I)

- **Temporal scalability** (HEVC version 1)
  - Higher frame rates in enhancement layer
  - e.g. 30 fps base layer to 60 fps enhancement layer

- **Spatial scalability**
  - Higher spatial resolutions in enhancement layer
  - e.g. 720p base layer to 1080p enhancement layer

- **Coarse grain SNR scalability**
  - Higher SNR qualities in enhancement layers
  - e.g. low-quality base-layer at 1 MBit/s to high-quality enhancement-layer at 8 MBit/s

- **External base layer scalability (new)**
  - Base layer encoded by another external encoder
  - e.g. H.264/AVC base layer with SHVC enhancement layer

[Grois2015]
SHVC: Scalability types (II)

- **Bit depth scalability (new)**
  - Higher bit depths in enhancement layer
  - e.g. base layer with bit depth of 8 to enhancement layer with bit depth of 10 bit

- **Interlace-to-progressive scalability**
  - Base layer in interlace format, enhancement layer in progressive format

- **Color gamut scalability (new)**
  - Higher color gamut in enhancement layers
  - e.g. BT.709 color gamut in base layer to BT.2020 in enhancement layer
MV-HEVC: Motivation

Use cases

- Stereo (2-view) “3D” movies and TV programs
  - Broadcast
  - Distributed media (Blu-Ray)
  - On-demand streaming
  - Cinema
- Multiple view displays

Alternatives

- Frame-compatible formats
- Simulcast

MV has to provide better performance.

[Grois2015]
MV-HEVC: Design

High-Level Syntax only

- Similar to MVC
- Takes advantage on multi-layer syntax elements in HEVC design
- No changes at block-level
- Allows re-use of existing HEVC encoder and decoder components
- Only changes in decoded picture buffer (DPB) and reference picture list creation

[Grois2015]
HEVC Products Forecast Overview

“The outlook for HEVC use in the digital video market is extremely positive...” [DTC2016]
What is Next?
Future video coding standard should consider from the beginning (among others) \cite{Bross2016}:

- UHD: 4K and up
- High Dynamic Range (HDR) and Wide Color Gamut content
- High Frame Rates
- High-Level Scalability (like SHVC)
- Drone Video
- 360 Video

The work on the video coding techniques beyond HEVC started already in 2015:

- Joint Video Exploration Team (JVET) of MPEG and VCEG organizations was established last October, 2015, in Geneva \cite{Bross2016}.
- JEM: Joint Video Exploration Model Software.
Future Video Coding Development (Cont.)

List of several coding tools that improve coding efficiency of HEVC:
[SG16-C0806]

- Larger coding tree blocks and larger transforms:
  - The CTU size signaled in the sequence level is set to be $256 \times 256$ by default;
  - Supporting Coding Tree Units (CTUs) larger than $64 \times 64$;
  - Supporting larger transforms, i.e., $64 \times 64$ DCT.

- Adaptive Transform:
  - Performed in addition to DCT-II and $4 \times 4$ DST-VII, which are employed in HEVC;
  - The newly introduced transform matrices are: DST-VII, DCT-VIII, DST-I and DCT-V.

[Bross2016]
Additional improvements as proposed, for example, by VCEG-AZ07:

- **Intra-Picture Prediction:**
  - Extended to support 65 intra prediction direction;

- And many other tools…

[VCEG-AZ07]

[Grois2015]
What is Next? – Current Standardization Timeline

- **Call For Evidence (CfE):** Subjective verification of the JEM coding efficiency compared to HEVC
- **Call for Proposals (CfP):** Submission and subjective evaluation of new video coding technologies

[Diagram showing timeline with key events and dates]

- Draft CfE (Jan., Geneva, CH)
- CfE Results Draft CfP (Jul., Torino, IT)
- Final CfE (Apr., Hobart, AU)
- Final CfP (Oct., Macao, MO)
- CfP results; and
  - Starting to work on the **Versatile Video Coding (VVC) Standard** (Apr., San-Diego, USA)
- Final Standard (October 2020)

[Based on Bross2016]
Future Video Coding Development

So, what is next?

The next is the VVC development, which already started in April, 2018!
Further Information

Document archives are publicly accessible
- http://phenix.it-sudparis.eu/jct

Overview page:
- http://hevc.hhi.fraunhofer.de

HEVC Reference Model (HM) software:
- https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware

JVET Joint Exploration Model (JEM) software:
- https://jvet.hhi.fraunhofer.de/svn/svn_HMJEMSoftware

JVET VVC Test Model (VTM) software:
- https://jvet.hhi.fraunhofer.de/svn/svn_VVCSoftware_VTM
References


References (Cont.)


Video Coding and HEVC

Dr. Dan Grois, grois@ieee.org