Overview: Video Coding Standards

- Video coding standards: applications and common structure
- ITU-T Rec. H.261
- ISO/IEC MPEG-1
- ISO/IEC MPEG-2
- State-of-the-art: H.264/AVC
Applications of Video Compression

Adapted from [Srinivasan et al., 2004]
## Applications of Video Compression

<table>
<thead>
<tr>
<th>Application</th>
<th>Bandwidth (kbps)</th>
<th>Coding Standards</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digital television broadcasting</td>
<td>2 . . . 6 Mbps</td>
<td>MPEG-2 (H.264/AVC)</td>
</tr>
<tr>
<td></td>
<td>(10...20 Mbps for HD)</td>
<td></td>
</tr>
<tr>
<td>DVD video</td>
<td>5 . . . 8 Mbps</td>
<td>MPEG-2</td>
</tr>
<tr>
<td>Blu-ray Disk</td>
<td>up to 40 Mbps</td>
<td>MPEG-2, H.264/AVC, VC-1 (up to 1080p)</td>
</tr>
<tr>
<td>Internet video streaming</td>
<td>100 . . . 2000</td>
<td>MPEG-1, H.264/AVC, VC-1, or similar proprietary</td>
</tr>
<tr>
<td>streaming</td>
<td>kbps</td>
<td></td>
</tr>
<tr>
<td>Video over 3G wireless</td>
<td>100 . . . 500 kbps</td>
<td>H.263, MPEG-4, H.264/AVC</td>
</tr>
</tbody>
</table>
Motion-compensated Hybrid Coding

H.261, MPEG-1, MPEG-2, H.263, MPEG-4, H.264/AVC

Diagram of motion-compensated hybrid coding process.
Video Compression Standards: Hierarchical Syntax
ITU-T Rec. H.261

- International standard for ISDN picture phones and for video conferencing systems (1990)
- Image format: CIF (352 x 288 Y samples) or QCIF (176 x 144 Y samples), frame rate 7.5 ... 30 fps
- Bit-rate: multiple of 64 kbps (= ISDN-channel), typically 128 kbps including audio
- Picture quality: for 128 kbps acceptable with limited motion in the scene
- Stand-alone videoconferencing system or desk-top videoconferencing system, integrated with PC
Macroblocks

- Macroblock (MB) of 16x16 pixels
- Sampling format: 4:2:0
- MB consists of 4 luminance and 2 chrominance blocks

![Diagram showing the sampling format and block structure.](image_url)
H.261 Motion-Compensated Prediction

- Integer-pel accuracy
- One displacement vector per macroblock
- Maximum displacement vector range +/-16 horizontally and vertically
- Adaptive loop filter, separable in 1-D horizontal and vertical impulse response: \([\frac{1}{4}, \frac{1}{2}, \frac{1}{4}]\)
- Differential encoding of motion vectors
H.261 Residual Coding

- 8x8 DCT
- Quantization
  - Uniform quantizer ($\Delta=8$) for intra-mode DC coefficients
  - Uniform threshold quantizer ($\Delta=2,4,…,62$) for AC coefficients in intra-mode and all coefficients in inter-mode
- Zig-zag scan
- Run-level coding for entropy coding
  - (zero-run, value) symbols
  - zero-run: the number of coefficients quantized to zero since the last nonzero coefficient
  - value: the amplitude of the current nonzero coefficient
H.261 Macroblock Types (VLC Table)

<table>
<thead>
<tr>
<th>Prediction</th>
<th>MQUANT</th>
<th>MVD</th>
<th>CBP</th>
<th>TCOEFF</th>
<th>VLC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intra</td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td>0001</td>
</tr>
<tr>
<td>Intra</td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td>0000 001</td>
</tr>
<tr>
<td>Inter</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td>1</td>
</tr>
<tr>
<td>Inter</td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td>0000 1</td>
</tr>
<tr>
<td>Inter+MC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0000 0000 1</td>
</tr>
<tr>
<td>Inter+MC</td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td>0000 0001</td>
</tr>
<tr>
<td>Inter+MC</td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td>0000 0000 01</td>
</tr>
<tr>
<td>Inter+MC+FIL</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>001</td>
</tr>
<tr>
<td>Inter+MC+FIL</td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td>01</td>
</tr>
<tr>
<td>Inter+MC+FIL</td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td>0000 01</td>
</tr>
</tbody>
</table>
MPEG-1/2: GOP Structure

Group of Pictures = GOP

| 1 | 3 | 4 | 2 | 6 | 7 | 8 | 5 |

I-Picture
P-Picture
B-Pictures

(time)
MPEG-1/2 Encoder
MPEG-1: coding of I-pictures

- I-pictures: intraframe coded
- 8x8 DCT
- Arbitrary weighting matrix for coefficients
- Differential coding of DC-coefficients
- Uniform quantization
- Zig-zag-scan, run-level-coding
- Entropy coding
- Unfortunately, not quite JPEG
MPEG-1: coding of P-pictures

- Motion-compensated prediction from an encoded I-picture or P-picture (DPCM)
- Half-pel accuracy of motion compensation, bilinear interpolation
- One displacement vector per macroblock
- Differential coding of displacement vectors
- Coding of prediction error with 8x8-DCT, uniform threshold quantization, zig-zag-scan as in I-pictures
MPEG-1: coding of B-pictures

- Motion-compensated prediction from two consecutive P- or I-pictures
  - either
    - only forward prediction (1 vector/macroblock)
  - or
    - only backward prediction (1 vector/macroblock)
  - or
    - Average of forward and backward prediction = interpolation (2 vectors/macroblock)

- Half-pel accuracy of motion compensation, bilinear interpolation

- Coding of prediction error with 8x8-DCT, uniform quantization, zig-zag-scan as in I-pictures
MPEG-2 vs. MPEG-1

- Efficiently compress interlaced digital video at broadcast quality
  - Frame pictures or field pictures
  - Adaptive frame/field prediction
  - Adaptive frame/field DCT
- Improved coding efficiency by different quantization, VLC tables, and additional coefficient scan patterns
- Spatial, temporal and SNR scalability profiles (rarely used)
Field 1
Frame = Both Fields Combined
Adaptive Frame/Field DCT
Adaptive Frame/Field Motion Compensation

Frame Prediction

Field Prediction
JVT Project

- **August 1999**: 1st test model (TML-1) of H.26L
- **December 2001**: Formation of the Joint Video Team (JVT) between VCEG and ISO/IEC JTC 1/SC 29/WG 11 (MPEG) to establish a joint standard project - **H.264 / MPEG4-AVC**
- **ITU-T Approval**: May 2003
- **ISO/IEC Approval**: October 2003

[source: G. Sullivan, VCEG]
H.264/AVC Coder

Input Video Signal

Split into Macroblocks 16x16 pixels

Coder Control

Transform/Scal./Quant.

Decoder

Intra/Inter Compensation

Motion Estimation

Scal./Inv. Transform

Deblocking Filter

Entropy Coding

Intra-frame Prediction

Output Video Signal

Motion Data

[source: G. Sullivan, VCEG]
Common Elements with other Standards

- Macroblocks: 16x16 luma + 2 x 8x8 chroma samples
- Block-wise motion compensation
- Variable block-size motion compensation
- Block transform of prediction error
- Scalar quantization
- I, P, and B coding types

[source: G. Sullivan, VCEG]
H.264 Motion Compensation Accuracy

- Input Video Signal
- Split into Macroblocks 16x16 pixels
- Coder Control
  - Transform/Scal./Quant.
  - Scaling & Inv. Transform
  - Entropy Coding
  - Control Data
  - Quant. Transf. coeffs
- Decoder
- Intra/Inter Coder Control
- Motion Estimation
- Intra-frame Prediction
- Motion Compensation

Types:
- MB Types: 16x16, 16x8, 8x16, 8x8
- 8x8 Types: 0, 1, 2, 3

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

[source: G. Sullivan, VCEG]
H.264 Multiple Reference Frames

- Multiple Reference Frames
- Generalized B Frames
- Weighted Prediction

[source: G. Sullivan, VCEG]
H.264 Intra Prediction

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

- e.g., Mode 3: diagonal down/right prediction
  a, f, k, p are predicted by 
  \((A + 2Q + I + 2) >> 2\)

[source: G. Sullivan, VCEG]
H.264 4x4 Transform

- **4x4 Block Integer Transform**

\[
A = \begin{bmatrix}
1 & 1 & 1 & 1 \\
2 & 1 & -1 & -2 \\
1 & -1 & -1 & 1 \\
1 & -2 & 2 & -1
\end{bmatrix}
\]

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

[source: G. Sullivan, VCEG]
Deblocking Filter

One dimensional visualization of an edge position

Filtering of $p_0$ and $q_0$ only takes place if:

1. $|p_0 - q_0| < \alpha(QP)$
2. $|p_1 - p_0| < \beta(QP)$
3. $|q_1 - q_0| < \beta(QP)$

Where $\beta(QP)$ is considerably smaller than $\alpha(QP)$

Filtering of $p_1$ or $q_1$ takes place if additionally:

1. $|p_2 - p_0| < \beta(QP)$ or $|q_2 - q_0| < \beta(QP)$

(QP = quantization parameter)

[source: G. Sullivan, VCEG]
Deblocking: Subjective Result for Intra

Highly compressed first decoded intra picture at 0.28 bit/sample

Without Filter

With H264/AVC Deblocking

[source: G. Sullivan, VCEG]
Deblocking: Subjective Result for Inter

Highly compressed decoded inter picture

Without Filter  With H264/AVC Deblocking

[source: G. Sullivan, VCEG]
Example Streaming Test Result

[Wiegand, et al. 2003]
Example Streaming Test Result

![Graph showing rate saving relative to MPEG-2 for different codecs (H.264/AVC MP, MPEG-4 ASP, H.263 HLP) vs. Y-PSNR (dB) for Tempete CIF 15Hz.]

[Wiegand, et al. 2003]
Example Entertainment-Quality Applications Result

Entertainment SD (720x576i) 25Hz

Y-PSNR [dB]

Bit-rate [Mbit/s]

MPEG-2

H.264/AVC MP

[Wiegand, et al. 2003]
Example Entertainment-Quality Applications Result

Entertainment SD (720x576i) 25Hz

Rate saving relative to MPEG-2

Y-PSNR [dB]

H.264/AVC MP

[Wiegand, et al. 2003]
Further reading


Bidirectional motion compensation.
<table>
<thead>
<tr>
<th>Levels</th>
<th>Profiles</th>
<th>Nonscalable</th>
<th>Scalable</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Simple 4:2:0</td>
<td>Main 4:2:0</td>
<td>Main+ 4:2:0</td>
<td>Next 4:2:2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>High</td>
<td>Max resol/</td>
<td>N/A</td>
<td>1920 x</td>
<td>N/A</td>
<td>1920 x</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>rate (Hz)</td>
<td></td>
<td>1152/60</td>
<td></td>
<td>1152/60</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Min. resol/</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>960 x</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>rate (Hz)</td>
<td></td>
<td></td>
<td></td>
<td>576/30</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Bitrate (Mbits/s)</td>
<td>N/A</td>
<td>80</td>
<td>N/A</td>
<td>100 (all layers)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>80 (base+mid)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>25 (base layer)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>High-1440</td>
<td>Max resol/</td>
<td>N/A</td>
<td>1440 x</td>
<td>1440 x</td>
<td>1440 x</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>rate (Hz)</td>
<td></td>
<td>1152/60</td>
<td>1152/60</td>
<td>1152/60</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Min. resol/</td>
<td>N/A</td>
<td>N/A</td>
<td>720 x</td>
<td>720 x</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>rate (Hz)</td>
<td></td>
<td></td>
<td>576/30</td>
<td>576/30</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Bitrate (Mbits/s)</td>
<td>N/A</td>
<td>60</td>
<td>N/A</td>
<td>80 (all layers)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>60 (all layers)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>40 (base+mid)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>15 (base layer)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Main</td>
<td>Max resol/</td>
<td>720 x</td>
<td>720 x</td>
<td>720 x</td>
<td>720 x</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>rate (Hz)</td>
<td>576/30</td>
<td>576/30</td>
<td>576/30</td>
<td>576/30</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Min. resol/</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>352 x</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>rate (Hz)</td>
<td></td>
<td></td>
<td></td>
<td>288/30</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Bitrate (Mbits/s)</td>
<td>15</td>
<td>15</td>
<td>15 (all layers)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>15 (all layers)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10 (base layer)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4 (base layer)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low</td>
<td>Max resol/</td>
<td>N/A</td>
<td>352 x</td>
<td>352 x</td>
<td>N/A</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>rate (Hz)</td>
<td></td>
<td>288/30</td>
<td>288/30</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Min. resol/</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>rate (Hz)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Bitrate (Mbits/s)</td>
<td>N/A</td>
<td>4</td>
<td>N/A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4 (all layers)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3 (base layer)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
HEVC/H.265 History

- HEVC Objective:
  - Achieves 2x higher compression compared to H.264/AVC
  - High throughput (Ultra-HD 8K @ 120fps) & low power
    - Implementation friendly features (e.g. built-in parallelism)
  - Benefits include
    - reduce the burden on global networks
    - easier streaming of HD video to mobile devices
    - account for advancing screen resolutions (e.g. Ultra-HD)

"HEVC will provide a flexible, reliable and robust solution, future-proofed to support the next decade of video"


Credit: Vivienne Sze & Madhukar Budagavi, ISCAS 2014 Tutorial
New Design Features

- HEVC (H.265) vs AVC (H.264)

Credit: Vivienne Sze & Madhukar Budagavi, ISCAS 2014 Tutorial
HEVC Development History

- Long hard fight, many casualties...

- Chairs
  - G. J. Sullivan (Microsoft)
  - J. R. Ohm (Aachen University)

- Meet Quarterly
  - 1st meeting (A) [January 2010]
  - 12th meeting (L) [January 2013]

- ~250 attendees per meeting representing ~70 companies

- Several hundred contributions per meeting

- Each meeting is around 9 - 10 days (14+ hours/day)

- Multiple parallel tracks
HEVC vs AVC

TABLE VI
AVERAGE BIT-RATE SAVINGS FOR EQUAL PSNR FOR ENTERTAINMENT APPLICATIONS

<table>
<thead>
<tr>
<th>Encoding</th>
<th>Bit-Rate Savings Relative to H.264/MPEG-4 AVC HP</th>
<th>MPEG-4 ASP</th>
<th>H.263 HLP</th>
<th>MPEG-2/ H.262 MP</th>
</tr>
</thead>
<tbody>
<tr>
<td>HEVC MP</td>
<td>35.4%</td>
<td>63.7%</td>
<td>65.1%</td>
<td>70.8%</td>
</tr>
<tr>
<td>H.264/MPEG-4 AVC HP</td>
<td>–</td>
<td>44.5%</td>
<td>46.6%</td>
<td>55.4%</td>
</tr>
<tr>
<td>MPEG-4 ASP</td>
<td>–</td>
<td>–</td>
<td>3.9%</td>
<td>19.7%</td>
</tr>
<tr>
<td>H.263 HLP</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>16.2%</td>
</tr>
</tbody>
</table>

PSNR = 10 \log_{10} \frac{(2^{\text{bitdepth}} - 1)^2 \cdot W \cdot H}{\sum_i (O_i - D_i)^2}

HEVC vs AVC

<table>
<thead>
<tr>
<th>Sequences</th>
<th>Bit-rate Savings</th>
</tr>
</thead>
<tbody>
<tr>
<td>BQ Terrace</td>
<td>63.1%</td>
</tr>
<tr>
<td>Basketball Drive</td>
<td>66.6%</td>
</tr>
<tr>
<td>Kimono1</td>
<td>55.2%</td>
</tr>
<tr>
<td>Park Scene</td>
<td>49.7%</td>
</tr>
<tr>
<td>Cactus</td>
<td>50.2%</td>
</tr>
<tr>
<td>BQ Mall</td>
<td>41.6%</td>
</tr>
<tr>
<td>Basketball Drill</td>
<td>44.9%</td>
</tr>
<tr>
<td>Party Scene</td>
<td>29.8%</td>
</tr>
<tr>
<td>Race Horse</td>
<td>42.7%</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td><strong>49.3%</strong></td>
</tr>
</tbody>
</table>

Subjective Tests for Entertainment Applications
(Random Access)

J. Ohm et al., "Comparison of the Coding Efficiency of Video Coding Standards—Including High Efficiency Video Coding (HEVC)," IEEE Transactions on Circuits and Systems for Video Technology, 2012
HEVC Coding Pipeline

HEVC Diagram

Credit: Prof. Oscar Au, Tutorial on HEVC
Quad Tree Decomposition:

- Better adaptation to different video content
- CTU divided into Coding Units (CU) with Quad tree
- Coding units divided into prediction units (PU)
- PU have different motion data or prediction modes

Ref:
HEVC Coding Structure

- **CTU** - Coding Tree Unit
  - Upto 64x64 pel block
  - Macro Block equivalent
  - Aka LCU: Largest Coding Unit

- **CU** – Coding Unit
  - Square NxN pel
  - Like Block

- **PU** – Prediction Unit
  - Inter Prediction
  - Intra Prediction

0/1 shows the quad-tree splitting flag
Prediction Unit

PU:
- Basic Unit for Transform & Quantization
  - PU is the basic unit for prediction of a 2Nx2N CU
    - 2Nx2N, Nxn, 2NxN, Nxn2N, 2NxuN, 2NxD, nLx2N, nRx2N
  - Allowed PU partitions are depending on prediction type
    - NxN PU allowed only for minimum CU. NOT allowed for larger CU.
    - Asymmetric motion partition or AMP (2NxuN, 2NxD, nLx2N, nRx2N) is not applied to inter 8x8 CU. (i.e. 2x8, 6x8, 8x2, 8x6 not allowed)

Example: Available PU for 64x64 CU
- Skip: PU = 64x64
- Intra: PU = 64x64
- Inter: PU = 64x64, 64x32, 32x64, (32x32)
  - 64x16, 64x48, 16x64, 48x64
Transform Unit (TU)

- **TU**
  - Square blocks for transform & quantization/coding, size: 4x4, 8x8, 16x16, 32x32
  - Max TU size: 32 for luma, 16 for chroma
  - Min TU size: 4x4 for both luma and chroma
  - TU size can be larger than PU for inter PU case (MV merge)
    - Each CU subdivided into square TU using quad-tree (split flag)
    - One square transform for each TU (DCT for inter/intra, DST for 4x4 intra luma)

Solid line = CU, dashed gray line = TU
HEVC Transforms

- Transform + Quant:

- HEVC supports 4x4, 8x8, 16x16, 32x32 integer transforms
  - Two types of 4x4 transforms (IDST-based for Intra, IDCT-based for Inter); IDCT-based transform for 8x8, 16x16, 32x32 block sizes
  - Integer transform avoids encoder-decoder mismatch and drift caused by slightly different floating point representations.
  - Parallel friendly matrix multiplication/partial butterfly implementation
  - Transform size signaled using Residual Quad Tree
- Achieves 5 to 10% increase in coding efficiency
- Increased complexity compared to H.264/AVC
  - 8x more computations per coefficient
  - 16x larger transpose memory
HEVC Intra-Prediction

- Intra-Prediction Modes
  - H.264/AVC has 10 modes
    - angular (8 modes), DC, planar
  - HEVC has 35 modes
    - angular (33 modes), DC, planar
  - Angular prediction
    - Interpolate from reference pixels at locations based on angle
  - DC
    - Constant value which is an average of neighboring pixels (reference samples)
  - Planar
    - Average of horizontal and vertical prediction
Intra-Predicted Basis

- As if it is a 1-non zero coefficient transform...

Fig. 4.2 Examples of $8 \times 8$ luma prediction blocks generated with all the HEVC intra prediction modes. Effects of the prediction post-processing can be seen on the top and left borders of the DC prediction (mode 1), top border of horizontal mode 10 and left border of vertical mode 26.

Ref:
Pre-Processing for Intra Prediction

- Smoothing filtering before intra prediction
  - Reference Sample Smoothing
    - Smooth out neighboring pixels (i.e., reference samples) before using them for prediction
    - Reduce contouring artifacts caused by edges in the reference sample arrays
    - Two modes
      - Three-tap smoothing filter
      - Strong intra smoothing with corner reference pixels
    - Application of smoothing depends on PU size and prediction mode

Post-Processing for Intra Prediction

- Remove discontinuity:
  - Boundary Smoothing
    - Intra prediction may introduce discontinuities along block boundaries
    - Filter first prediction row and column with three-tap filter for DC prediction, and two-tap for horizontal and vertical prediction

![Diagram showing post-processing for Intra Prediction]

Image source: JCTVC-F172, July 2011
HEVC Intra as Still Image Coding Standard

- Advancing the state of art by quite a bit
  - HEVC also provides improved compression for still images

<table>
<thead>
<tr>
<th></th>
<th>BD-Rate Reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>H.264/AVC (intra only)</td>
<td>15.8%</td>
</tr>
<tr>
<td>JPEG 2000</td>
<td>22.6%</td>
</tr>
<tr>
<td>JPEG XR</td>
<td>30.0%</td>
</tr>
<tr>
<td>Web P</td>
<td>31.0%</td>
</tr>
<tr>
<td>JPEG</td>
<td>43.0%</td>
</tr>
</tbody>
</table>

HEVC Inter Prediction

- Supports 4x4 block $\frac{1}{4}$ pel accuracy Motion Compensation
  - Motion vectors can have up to $\frac{1}{4}$ pixel accuracy (interpolation required)

- In H.264/AVC, luma uses 6-tap filter, and chroma uses bilinear filter
- In HEVC, luma uses 8/7-tap and chroma uses 4-tap
  - Different coefficients for $\frac{1}{4}$ and $\frac{1}{2}$ positions
- Restricted prediction on small PU sizes
Quite a bit of memory and computing cost:

- Require integer pixels (highlighted in red) to interpolate fractional pixels (highlighted in blue).
- To interpolate $N \times N$ pixels requires up to $(N+7) \times (N+7)$ reference pixels.
- Use 1-D filters (order matters for greater than 8-bit video).
Diamond Search

- Integer pixel motion estimation
  - Search Strategy
    1. Search center is motion vector predictor
    2. Diamond search around center (search range = 64 → 7 steps [1, 2, 4.. 64]); early termination if best candidate doesn’t change in 3 steps.
    3. If best candidate > 5 pixels away from search center, do raster scan search (5 pixel steps).
    4. Perform diamond search around best candidate from step 2 or 3. If new best candidate found repeat 4.

Image Source: N. Purnachand et al., IEEE ICCE-Berlin, 2012

Reference
HEVC Entropy Coding

- Binary Arithmetic Coding:
  - HEVC uses Context Adaptive Binary Arithmetic Coding (CABAC)
    - 10 to 15% higher coding efficiency compared to CAVLC

HEVC CABAC Throughput Improvement

Context Bypassing Mode:

- Reduce total number of bins
- Reduce context coded bins
- Reduce context dependencies
- **Grouping bypass bins**
- Reduce parsing dependencies
- Reduce memory requirements

Reduction in **worst case** bins for 16x16 pixels

<table>
<thead>
<tr>
<th></th>
<th>Total bins</th>
<th>Context bins</th>
<th>Bypass bins</th>
</tr>
</thead>
<tbody>
<tr>
<td>H.264/AVC</td>
<td>20861</td>
<td>7805</td>
<td>13056</td>
</tr>
<tr>
<td>HEVC</td>
<td>14301</td>
<td>884</td>
<td>13417</td>
</tr>
<tr>
<td>Ratio</td>
<td>1.5x</td>
<td>9x</td>
<td>1x</td>
</tr>
</tbody>
</table>

15 cycles: 01101010001010101
9 cycles: 01110111000000000

- 3x reduction in context memory
- 20x reduction in line buffer for context selection

Parallel Processing Tools: Slice/Tile

Slices (also in H.264/AVC)

Slice 0
Slice 1
Slice 2
Slice 3

Tiles

Tile 0
Tile 1
Tile 2
Tile 3

Wavefront Parallel Processing (Interleaved Entropy Slices*)

Substream 0
Substream 1
Substream 2
Substream 3


Credit: Vivienne Sze & Madhukar Budagavi, ISCAS 2014 Tutorial
Profiles, Levels and Tiers

- Operating HEVC in Profiles/Levels/Tiers

  - Profile defines set of tools for different applications
    - Main, Main 10, Main Still Picture
    - 8-bits/sample → 16.78 million colors
    - 10-bits/sample → 1.07 billion colors

  - Level defines the maximum supported resolution and frame rate
    - e.g. Level 4.0, 1920x1080 @ 32 fps
    - Level 5.0, 4096x2160 @ 30 fps

  - Bit-rates defined by level and tier
    - Main and High (professional)
HEVC Resources

- **Main Spec:**

- **T-CSVT Special Issue:**

- **Springer Book:**

- **HM (open source software):**
  - https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSovware/

- **HARP: A python HM access tool:**
  - [http://www.lms.lnt.de/HARP/](http://www.lms.lnt.de/HARP/)

---

**OPEN SOURCE HEVC ANALYZER FOR RAPID PROTOTYPING (HARP)**

- **HM 14-0 (C++ image and matrix processing)**
- **CMake (build chain management)**
- **Qt (C++ class library)**
- **OpenCV (C++ image and matrix processing)**
- **PicklingTools (C++ Python Dictionaries)**
- **Valgrind Dev (linking against C-API)**
- **Extensions HM Code Base**
  - TApp-TLib-Encoder-Decoder-Common (BSD)
- **HARP C++ Core**
  - CShowUnitCloseup (GPL)
  - CShowPredictionUnits (GPL)
  - CShowTransformUnits (GPL)
  - CMake Script (GPL)
- **HARP Python GUI**
  - GUI Sources (GPL)
  - Plot Examples (GPL)
  - Callgrind (path analysis, optimization)
- **Numerical Python (Numpy)**
- **Dict. Examples (GPL)**
- **Eclipse IDE (C++/Python development)**
- **PyQtGraph (real-time plotting)**
- **Matplotlib (publication-ready plotting)**
Summary

- Light Field Compression
  - Sensor data at lenslet
  - Support new perspective and re-focusing
  - Interesting R-D optimization issue at new perspective and focusing point in the light field
  - Calls for new coding tools, based on HEVC intra, e.g.,

- HEVC
  - Crown Jewel of 20+ years of research and engineering
  - Significant performance gains over the previous state of art
  - Worthwhile to master this tool for a variety of research and biz opportunities.