High Efficiency Video Coding (HEVC)

Oscar C. Au (PhD, Princeton Univ)

Dept of Electronic and Computer Engineering
Hong Kong University of Science and Technology
Email: eeau@ust.hk
Oscar C. Au

- BS, Toronto. MA/PhD, Princeton. Postdoc, Princeton.
- Professor, HKUST. Director, Multimedia Tech Center.
- Steering Committee, ICME/TMM.
- IEEE/HKIE Fellow. BoG, APSIPA.
- Best Paper Awards: SiPS/PCM/MMSP/ICIP
- AE of 8 journals: TCSVT, TIP, TCAS1, TVCIR, JSPS, TSIP, JMM, JFI.
- Chair of 3 TC: CAS MSATC, SPS MMSP TC, APSIPA IVM TC.
- Member of 5 TC: CAS VSPS/DSP, SPS IVMSP/IFS, ComSoc MMC.
- 400+ papers. H-index=29. 100+patents filed. 20 granted.
- 80+ standard contribution (MPEG/VCEG/JCTVC/AVS).
Outline

- HEVC standardization status
- HEVC structure
- Intra Prediction
- Inter Prediction
- Scanning/ Transform
- Entropy coding
- In-loop filtering
- Parallel processing
- Performance
- Continuing Work
Development of Video Coding Standards

- **H.261** (1990)
- **H.263** (1996)
- **MPEG-1** (1993)
- **MPEG-2** (H.262) (1995)
- **MPEG-4 v1** (1999)
- **MPEG-4 v2** (2000)
- **MPEG-4 v3** (2001)
- **H.263+** (1998)
- **H.263++** (2000)
- **H.264 (AVC)** (2004)
- **SVC (H.264-G)** (2007)
- **HEVC (H.265)** (2013)

**Standards and Organizations:**

- **ISO/IEC**
- **MPEG**
- **ITU-T**
- **VCEG**
- **SVC (H.264-G)**
High Efficiency Video Coding (HEVC)

- A new standard under development by ISO and ITU-T
  - Joint MPEG & VCEG new team: JCT-VC
- Target at HDTV or ultra-HDTV compression, with substantially improved coding efficiency compared to H.264/AVC, i.e. 50% bit rate reduction
- Very active project (hundreds of documents every meeting), very diverse company & university participation.
HEVC Timeline

2010.01  Formal joint CfP from VCEG and MPEG
2010.04  JCT-VC team, HEVC joint project, full proposals
2010.07  TMuC SW ready, tool experiments (TE)
2010.10  HM SW ready, core experiments (CE)
2011.02  WD
2012.02  CD
2012.07  DIS
2012.10  SoDIS (Study of Draft International Standard)
          HEVC conformance testing is triggered.
2013.01  FDIS (expected)
<table>
<thead>
<tr>
<th></th>
<th>Location</th>
<th>Attendees</th>
<th>Documents</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Dresden</td>
<td>188</td>
<td>~40</td>
</tr>
<tr>
<td>2</td>
<td>Geneval</td>
<td>221</td>
<td>~120</td>
</tr>
<tr>
<td>3</td>
<td>Guangzhou</td>
<td>244</td>
<td>~300</td>
</tr>
<tr>
<td>4</td>
<td>Daegu</td>
<td>248</td>
<td>~400</td>
</tr>
<tr>
<td>5</td>
<td>Geneva</td>
<td>226</td>
<td>~500</td>
</tr>
<tr>
<td>6</td>
<td>Torino</td>
<td>254</td>
<td>~700</td>
</tr>
<tr>
<td>7</td>
<td>Geneva</td>
<td>284</td>
<td>~1000</td>
</tr>
<tr>
<td>8</td>
<td>San Jose</td>
<td>255</td>
<td>~700</td>
</tr>
<tr>
<td>9</td>
<td>Geneva</td>
<td>2xx</td>
<td>~550</td>
</tr>
<tr>
<td>10</td>
<td>Stockholm</td>
<td>2xx</td>
<td>~500</td>
</tr>
<tr>
<td>11</td>
<td>Shanghai</td>
<td>235</td>
<td>~350</td>
</tr>
</tbody>
</table>
HEVC Architecture

Key coding tools basically similar to AVC (and prior standards)

- Block-based coding tools
- Variable block sizes
- Block-based motion compensation
- Fractional-pel motion vectors (1/4-pel)
- Spatial intra prediction
- Spatial transform of residual difference
- Integer-based transform
- Arithmetic entropy coding
- In-loop filtering to form final decoded picture
- Largest Coding Unit (LCU), now called Coding Tree Unit (CTU)
- Coding Unit (CU), now called Coding Block (CB)
- Prediction unit (PU), now called Prediction Block (PB)
- Transform unit (TU), now called Transform Block (TB)
Unit Definition: Coding Unit (CU)

- **CU** is basic processing block (square)
  - similar to macroblock
  - various power-of-2 sizes (square)
  - Quad-tree structure indicated by recursive split flag
  - largest CU (LCU, now called Coding Tree Unit, CTU)
  - User-defined max hierarchical depth in SPS (sequence parameter set)
  - size independent syntax: simple and elegant
  - Chroma CU is quarter size of luma CU in Main Profile

\[
\text{e.g. CTU N=64,}
\]
Picture Coding structure

LCU Quad-tree split example

0/1 shows the quad-tree splitting flag

64x64 LCU

32x32 CU

16x16 CU

8x8 CU
Picture Coding Structure

- **Prediction Unit (PU) is the basic unit for prediction**
  - **PU** is the basic unit for prediction of a 2Nx2N CU
    - 2Nx2N, NxN, 2NxN, Nx2N, 2NxU, 2NxN, 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 (2NxU, 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)** is basic unit for transform and quantization.
  
  – 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
• Transform Unit (TU) is the basic unit for transform and quantization.

  • For inter, may exceed size of PU, but cannot exceed size of CU
  • For intra, cannot exceed PU size
  • MaxTUSize is coded in SPS, its value in range of [4, 8, 16, 32]
  • In HM, Max. TU size is 32x32 for Luma and 16x16 for Chroma
  • Absolute Min. TU size is 4x4 for both Luma and Chroma
    – MaxTUDepth is coded in SPS to control the minTUSize.
    – Separate MaxTUDepth control for intra and inter.
• An example of the CU, PU & TU in a LCU

• MaxTUDepth = 3 in this example
• Flags to indicate the TU split status

• Max TU is 32x32 for Luma and 16x16 for chroma
• Min TU is 4x4 for luma & chroma
• The green TU block is not applicable to intra_NxN PUs
Intra Prediction

- Intra PCM mode
- Intra Luma and Chroma prediction modes
- Intra reference padding
- Mode dependent intra reference sample filtering
Intra Prediction

• Intra PCM mode

  – In I_PCM mode, prediction, transform, quantization and entropy coding are bypassed.

  – Samples are directly represented by a predefined number of bits. (signal characteristic are ill-posed, e.g. noise-like signals.)

  – I_PCM mode is only available for 2Nx2N PU
    • Max and min I_PCM CU size is signalled in SPS
    • Legal I_PCM CU sizes are 8x8, 16x16 and 32x32
    • User-selected PCM sample bit-depths, signalled in SPS for luma and chroma, separately

  – Take Luma sample as an example:

    \[
    \text{recSamplesL}[i, j] = \text{pcm_sample_luma}[ (nS \ast j) + i] \ll (\text{BitDepthY} - \text{PCMBitDepthY})
    \]

  • Lossless coding when \text{PCMBitDepthY} = \text{BitDepthY}. 
Intra Prediction

- Intra Luma and Chroma prediction modes

Luma:
  - **35 modes**: Planar + DC + 33 angular prediction modes for all block sizes.
  - **Mode Coding**: 3 MPM modes (VL coding) + 32 rest modes (FL coding)

Chroma:
  - **5 modes**: DM + Planar + DC + Horizontal + Vertical

Fig. 6. Modes and directional orientations for intra-picture prediction.
Intra Prediction

- Planar mode (Surface fitting)
  - Final T at (0, N+1), final L at (N+1, 0)

Interpolation indicated by dashed arrow
Replication indicated by dotted arrow
Intra Prediction

- DC mode

- A variable DCVal is derived as $(k = \log_2(nS))$

$$\left( \sum_{x'=-1}^{nS-1} p[x', -1] + \sum_{y'=-1}^{nS-1} p[-1, y'] + nS \right) \gg (k + 1)$$

$$\text{predSamples}[x, y] = \text{DCVal},$$

with $x, y = 0..nS-1$

- Boundary smoothing is applied on Luma DC predictions when TU size is less than 32x32

- predSamples[0, 0] = $(p[-1, 0] + 2*\text{DCVal} + p[0, -1] + 2) \gg 2$
- predSamples[x, 0] = $(p[x, -1] + 3*\text{DCVal} + 2) \gg 2$, with $x = 1..nS-1$
- predSamples[0, y] = $(p[-1, y] + 3*\text{DCVal} + 2) \gg 2$, with $y = 1..nS-1
Intra Prediction

- Vertical and horizontal mode

Conventional prediction

Example 1
Original image
Small distortion
Prediction image

Example 2
Original image
Large distortion
Prediction image
Prediction direction
Intra Prediction

• Vertical and horizontal mode

Horizontal prediction:

\[ S'(x, y) = \begin{cases} 
S(-1, y) + \frac{(S(x, -1) - S(-1, -1))}{2} & (y = 0) \\
S(-1, y) & (y \geq 1) 
\end{cases} \]

- Original image
- Prediction image

Prediction block

Conventional prediction

Prediction direction

apply only to this line

\[ dS = \frac{S(x, -1) - S(-1, -1)}{2} \]
**Intra Reference Padding**

- Totally \( nS^4 + 1 \) ref. samples from top & left neighbours for a \( nS \times nS \) TU
  
  \[ p[x,y] \text{ with} \]
  
  \[ x = -1, y = -1..nS^2 - 1 \text{ and} \]
  
  \[ x = 0..nS^2 - 1, y = -1 \]

- \( p[x,y] \) is marked “unavailable” when:
  
  - the corresponding neighbour TU is not available,
  
  - it is not intra coded & constrained_intra_pred_flag = 1
Intra Prediction

• Mode dependent intra reference sample filtering
  ✓ Filtering ref. samples before using them to predict current TU
    ➢ Only applied to Luma component, no smoothing for Chroma
  ✓ Mode dependent filtering condition
    ➢ 4x4: No filtering
    ➢ 8x8: Only filtering for mode 0, 2, 18, 34
    ➢ 16x16: Filter for all modes except 1, 9, 10, 11, 25, 26, 27
    ➢ 32x32: Filter for all modes except 1, 10, 26

✓ Smoothing filtering
  ➢ Filter \{1 \ 2 \ 1\} is applied when filtering is needed
  ➢ No smoothing for the right-most and bottom-most ref. pixels
  ➢ L-shape \{1 \ 2 \ 1\} filter for the corner ref. pixel
Intra Prediction

✓ Anti-contour smoothing filtering (for 32x32 TU only) (K0139)

LD-main at QP=37 in HM8.0
✓ Anti-contour smoothing filtering (for 32x32 TU only) (K0139)

When:
- strong_intra_smoothing_enable_flag=1
- If $|p0+p2-2*p1|>T$, linear interpolation between $p0$ and $p2$ before intra-prediction
  - Top-left ($p[-1, -1]$), right most ($p[63, -1]$) and bottom most ($p[-1, 63]$) ref. samples are unchanged
  - Weighted average for the left and above ref. sample line (Bi-linear)

Propagation of discontinuity at block boundary to sample locations within the predicted sample array.
Inter Prediction

- Limits on small PU sizes
- AMVP
- Merge
- Interpolation filters
- Reference motion data compression
Inter Prediction

• Limits on small PU size
  
  – Inter_NXN is only allowed for smallest allowed CU size.
  – Bi-directional prediction is not allowed for inter 8x4 and 4x8
    • For AMVP
    • For merge
**Inter Prediction**

- **AMVP**
  - Advanced Motion Vector Prediction (AMVP)
    - Two MVP candidates for one reference list
    - Explicit motion vector predictor signaling
      - Two reference lists for bi-prediction.
      - One MVP index for each reference list.
      - MV difference (MVD) will be coded
      - Each PU has its own reference index and MVs
Inter Prediction

- **AMVP**
  - Construction process of AMVP MVP candidate list
    - Two predictors derived from 2 spatial candidates and 1 temporal candidate
      - Spatial A: the first available one of A0 & A1
      - Spatial B: the first available one of B0, B1 & B2
      - If A is equal to B, drop B and add the temporal candidate
      - If list size < 2, add zero motion candidate
      - Fix final list size to 2
Inter Prediction

• **Merge**
  
  • Merge means current PU has identical MV with one of its neighbours
  
  – One flag to signal if the PU is coded in Merge Mode or not
  
  – If Merge Mode, explicitly signals which candidate to merge (Merge Index)
  
  – Candidate list is built on-line from the spatial candidates and temporal candidates.
  
  – No MVD information need to be coded
    
    • But need to code DCT coef. (Like Direct mode in AVC)
  
  – Only one merge index for a PU (even when bi-predicted)
Inter Prediction

- Interpolation filters for luma
  - For Luma fractional pixels
    - A single consistent separable interpolation process to generate all fractional positions.
    - Pixel accuracy is 1/4
    - 7-tap for quarter pixel and 8-tap for half pixel
    - Filter kernel were partially derived from DCT basis function equations.

<table>
<thead>
<tr>
<th>index</th>
<th>-3</th>
<th>-2</th>
<th>-1</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>hfilter[i]</td>
<td>-1</td>
<td>4</td>
<td>-11</td>
<td>40</td>
<td>40</td>
<td>-11</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>qfilter[i]</td>
<td>-1</td>
<td>4</td>
<td>-10</td>
<td>58</td>
<td>17</td>
<td>- 5</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
Inter Prediction

- Interpolation filters for chroma

- For Chroma fractional pixels
  - Pixel accuracy is 1/8
  - \( ab_{0,0}, ac_{0,0}, ad_{0,0}, ae_{0,0}, af_{0,0}, ag_{0,0}, ah_{0,0} \): 4-tap filtering on \( B_{i,0} \) (\( i = -1..2 \)) in hor. direction
  - \( ba_{0,0}, ca_{0,0}, da_{0,0}, ea_{0,0}, fa_{0,0}, ga_{0,0}, ha_{0,0} \): 4-tap filtering on \( B_{0,j} \) (\( j = -1..2 \)) in vert. direction
  - \( bX_{0,0}, cX_{0,0}, dX_{0,0}, eX_{0,0}, fX_{0,0}, gX_{0,0}, hX_{0,0} \) (\( X = b, c, d, e, f, g, h \)): 4-tap filter \( aX_{0,i} \) (\( i = -1..2 \)) in vert. direction
  - Different filter coefficients for different positions.

\[
\begin{align*}
1/8: & \{-2, 58, 10, -2\} \\
1/4: & \{-4, 54, 16, -2\} \\
3/8: & \{-6, 46, 28, -4\} \\
1/2: & \{-4, 36, 36, -4\}
\end{align*}
\]
Scanning/ Transform

- Mode dependent coefficient scanning (MDCS)
- Coefficient Grouping (CG)
- Intra DCT/DST transforms
- Transform skip
Scanning/ Transform

• Scanning
  - No zigzag scan for coefficients scanning in HEVC.
  - Inter block uses Diagonal scan
  - MDCS applies to only 4x4 and 8x8 TU in intra.

• MDCS Scan pattern is determined based on TU size and intra prediction mode
  - Diagonal scan, Horizontal scan, Vertical scan
  - A LUT is defined as shown in next slide

![Coefficient scanning methods in HEVC](image.png)

Fig. 9. Three coefficient scanning methods in HEVC; (a) diagonal up-right scan (b) horizontal scan and (c) vertical scan
### Scanning

0: Diagonal, 1: Horizontal, 2: Vertical

<table>
<thead>
<tr>
<th>IntraPredMode</th>
<th>log2TrafoSize − 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0 (4x4)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2-3</td>
<td>0</td>
</tr>
<tr>
<td>4-5</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>7-8</td>
<td>2</td>
</tr>
<tr>
<td>9-10</td>
<td>0</td>
</tr>
<tr>
<td>11-12</td>
<td>1</td>
</tr>
<tr>
<td>13-14</td>
<td>0</td>
</tr>
<tr>
<td>15-16</td>
<td>2</td>
</tr>
<tr>
<td>17-19</td>
<td>0</td>
</tr>
<tr>
<td>20-23</td>
<td>1</td>
</tr>
<tr>
<td>24-27</td>
<td>0</td>
</tr>
<tr>
<td>28-31</td>
<td>2</td>
</tr>
<tr>
<td>32-33</td>
<td>0</td>
</tr>
<tr>
<td>34</td>
<td>0</td>
</tr>
<tr>
<td>35</td>
<td>0</td>
</tr>
</tbody>
</table>
• Coefficient grouping

- TU larger than 4x4 will be sub-divided into CGs of size 4x4, and scanned at 2 levels.
- Significant_coef_CG_flag is signaled for each CG level to indicate whether the CG has non-zero coefficients.
• Transform

• Integer DCT/DST transform on scaled coefficients
  – DST is applied to intra 4x4 Luma TU, DCT is applied to other TUs
  – Core transforms: 32x32, 16x16, 8x8, 4x4

• Transform skipping of 4x4 TU
  – A solution for screen content coding in HEVC
    – BD rate reduction on class F: 7.8/5.6/3.4% for AI/RA/LD
  – This feature could be turn on by *transform_skip_enabled_flag* in PPS
    – Both luma and chroma 4x4 TUs could be transform skipped
  – If on, a flag per 4x4 TU to indicate transform skip is applied or not
    – No change to prediction, CABAC, in-loop filter, coef. scan and quantization
Entrophy Coding

- Only one entropy coding method: CABAC (context adaptive binary arithmetic coding)
  - Similar core algorithm with CABAC in AVC
  - Better context modeling, less context numbers, but higher compression ratio!
  - Dependencies between coded data are also improved to enable further throughput maximization.
In-loop filtering

• Cascaded in-loop filters

- Deblocking filter to reduce blocking artifact
  - HEVC uses similar (boundary strength based) deblocking filter as H.264/AVC but
    - Suitable modification for big blocks and complex TU boundaries
    - Filtering order modification for parallel processing

• Sample adaptive offset (SAO)
  - Adaptively add offsets to enhance deblocked pixels
  - Each LCU can be enhanced by Band Offset or Edge Offset
In-loop filtering

- **Deblocking filter: Block strength (bS) decision**
  - Filtering a picture in LCU unit
    - Picture vertical edge filtering first, then picture horizontal edge filtering
  - Filtering both PU and TU edges
    - For both Luma and Chroma
    - Expect the picture boundary
    - bS calculated for each 4 pixels along the PU/TU edges
      - Chroma shares the same bS with Luma
  - Deblocking steps for a LCU
    - Derive TU and PU edges
    - Derive bS values
    - Filter luma edges
      - Decision (no-/weak-/strong-filtering)
      - Filtering
    - Filter chroma edges
      - Filtering
In-loop filtering

• Sample adaptive offset (SAO)

- SAO is a process which modifies the samples after the deblocking filter through a look-up table. (non-linear)
- Depending on the local gradient at the sample position, a certain offset value from a look-up table is added to the sample.
- Found to be efficient to suppress pseudo-edges referred to as “banding artifacts” and “ringing artifacts”, etc.
- Performed on a region basis, adapted per LCU.
  - sao_type_idx=0, SAO is not applied; sao_type_idx=1, band offset types.
  - Sao_type_idx=2, edge offset types.
    - Sao_eo_class = 1…4 to indicate the which directional gradients is used in the edge offset types.

![Diagram](a) 1-D 0-degree  (b) 1-D 90-degree  (c) 1-D 135-degree  (d) 1-D 45-degree
In-loop filtering

- **Sample adaptive offset (SAO)**
  - For a specified EO type, decoder derives for each pixel which category it belongs to, and then add the received offset of the category to the pixel
    - 4 offsets are sent to decoder for categories 1~4
    - Offset value should be >=0 for category 1 & 2, and <= 0 for category 3 & 4.

<table>
<thead>
<tr>
<th>Category</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>c &lt; 2 neighboring pixel values</td>
</tr>
<tr>
<td>2</td>
<td>c &lt; 1 neighbor &amp;&amp; c == 1 neighbor</td>
</tr>
<tr>
<td>3</td>
<td>c &gt; 1 neighbor &amp;&amp; c == 1 neighbor</td>
</tr>
<tr>
<td>4</td>
<td>c &gt; 2 neighbors</td>
</tr>
<tr>
<td>0</td>
<td>None of the above</td>
</tr>
</tbody>
</table>
In-loop filtering

• Sample adaptive offset (SAO)

Band offset (BO)
  • What is band?
    – Classify each pixel according to its intensity
      ➢ Intensity is equally divided into 32 bands/categories
    – sao_band_position to signal the first band that SAO will be applied
      ➢ SAO will be applied to the 4 bands after sao_band_position
      ➢ 4 offset values will be coded for the 4 bands

  • Per-pixel process at decoder-side
    – Decide which category the pixel value belongs to
    – Add the received offset of the category to the pixel value
In-loop filtering

- Sample adaptive offset (SAO)

  - SAO parameters signalling
    - Per-LCU based SAO parameter signalling
    - Luma and chroma have independent SAO parameters (type, offset…)
      - For each component, a LCU can use the SAO parameters from neighbouring LCUs
        - “from left” syntax: `sao_merge_left_flag`, or
        - “from above” syntax: `sao_merge_up_flag`
Parallel processing

- WPP (Wave-front parallel processing)
- Tile base partition

(WPP is not allowed to be used in combination with tiles in HEVC)
Parallel processing

- **WPP (Wave-front parallel processing)**

At the end of each LCU line
- Perform a CABAC flush of LCU line (write remaining bits + stop bit)
- Initialize CABAC states (lower bound L and range R of the interval)
- Then, Each line produces a “**Chunk**” of compressed data
Parallel processing

• **Tiles**

• **What is “tile”?**
  – Self-contained and independently-decodable rectangular regions of the picture.
  – Vertical and horizontal boundaries partition of a picture into columns and rows
  – Boundary locations may be specified individually or uniformly spaced (signalled in SPS and PPS)
  – Always rectangular with an integer number of LCUs

• **Why partitioning into tiles?**
  – High level parallel processing
Parallel processing

- WPP (Wave-front parallel processing)
- Tile base partition
Performance

- HEVC Draft 7 Main Profile vs. AVC High Profile (J0236)

<table>
<thead>
<tr>
<th></th>
<th>All Intra</th>
<th>Random Access</th>
<th>Low Delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class A</td>
<td>−23.4%</td>
<td>−36.9%</td>
<td></td>
</tr>
<tr>
<td>Class B</td>
<td>−22.8%</td>
<td>−39.5%</td>
<td>−41.1%</td>
</tr>
<tr>
<td>Class C</td>
<td>−20.3%</td>
<td>−30.4%</td>
<td>−32.5%</td>
</tr>
<tr>
<td>Class D</td>
<td>−16.9%</td>
<td>−28.2%</td>
<td>−29.7%</td>
</tr>
<tr>
<td>Class E</td>
<td>−28.8%</td>
<td></td>
<td>−42.9%</td>
</tr>
<tr>
<td>Class F</td>
<td>−22.7%</td>
<td>−26.0%</td>
<td>−29.9%</td>
</tr>
<tr>
<td>Average</td>
<td>−22.2%</td>
<td>−32.5%</td>
<td>−35.1%</td>
</tr>
<tr>
<td>Average without F</td>
<td>−22.1%</td>
<td>−34.1%</td>
<td>−36.4%</td>
</tr>
</tbody>
</table>
Continuing Work

- Ad-Hoc Groups after Oct. 2012 Shanghai Meeting
  - JCT-VC project management
  - HEVC Draft and Test Model editing
  - HEVC HM software development and software technical evaluation
  - HEVC Still Picture profile
  - HEVC conformance test development
  - HEVC in-loop filtering
  - HEVC range extensions
  - Screen content coding
  - High-level syntax
  - SHVC tool experiments
  - SHVC software
  - SHVC upsampling and downsampling filters
Q&A