CS/EE 5590 / ENG 401 Special Topics (17804, 17815, 17803)

Lec 09

Video Signal Processing II

Zhu Li

Course Web:

http://l.web.umkc.edu/lizhu/teaching/2016sp.video-communication/main.html
Outline

- Lecture 08 ReCap & HW-2
- Video Signal Processing Tricks
  - Motion Vector Coding
  - Intra Prediction
  - De-Blocking Filters
  - Scalable Video Coding
- Video Codec Standards and software
## Color Space: Down-sampling

- **RGB components of an image have strong correlation.**
  - Can be converted to YUV space for better compression.
- **HVS is more sensitive to the details of brightness than color.**
- **Can down-sample color components to improve compression.**

<table>
<thead>
<tr>
<th>Color Space</th>
<th>Luma Sample</th>
<th>Chroma Sample</th>
</tr>
</thead>
<tbody>
<tr>
<td>YUV 4:4:4</td>
<td>×</td>
<td>○</td>
</tr>
<tr>
<td>No downsampling of Chroma</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Color Space</th>
<th>Luma Sample</th>
<th>Chroma Sample</th>
</tr>
</thead>
<tbody>
<tr>
<td>YUV 4:2:2</td>
<td>×</td>
<td>○</td>
</tr>
<tr>
<td>2:1 horizontal downsampling of chroma components</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 chroma samples for every 4 luma samples</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Color Space</th>
<th>Luma Sample</th>
<th>Chroma Sample</th>
</tr>
</thead>
<tbody>
<tr>
<td>YUV 4:2:0</td>
<td>×</td>
<td>○</td>
</tr>
<tr>
<td>2:1 horizontal downsampling of chroma components</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 chroma sample for every 4 luma samples</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**MPEG-1**

**MPEG-2**
GoP, I, P, and B Frames

- **I frames (Key frames):**
  - Intra-coded frame, coded as a still image. Can be decoded directly.
  - Used at GOP head, or at scene changes.
  - Allow random access, improves error resilience.

- **P frames: (Inter-coded frames)**
  - Predicated from the previous frame.

- **B frames: Bi-directional interpolated prediction frames**
  - Predicted from both the previous frame and the next frame: more flexibilities ⇒ better prediction.
  - Useful when new objects come into the scene.
Sub-pixel Motion Estimation

- **Six-tap filter for half-pixel samples:**
  - \( h = \left\lfloor \frac{(E - 5F + 20G + 20H - 5I + J) + 16}{32} \right\rfloor \)
  - Similar operation for \( v_i \) in vertical direction.
  - \( x = \left\lfloor \frac{(v_1 - 5v_2 + 20v_3 + 20v_4 - 5v_5 + v_6) + 16}{32} \right\rfloor \)

- **Bi-linear filter for 1/4-pixel samples:**
  - Matlab:
    - `bilinearInterpolation.m`

\[ I(x', y') = [1-b \ b] \begin{bmatrix} I(x, y) & I(x+1, y) \\ I(x, y+1) & I(x+1, y+1) \end{bmatrix} [1-a \ a] \]
Fast Search in Motion Estimation

- Diamond Pattern Search

![Diamond Pattern Search Diagram]

- Start with large diamond pattern at (0,0)
- If best match lies in the center of large diamond, proceed with small diamond
- If best match does not lie in the center of large diamond, center large diamond pattern at new best match

Ref:
HW-2: Arithmetic Coding

- Coding residual and bi-level images:
  - Predicted image residual is approx. geometrical distribution
  - Binarize via ExpGolomb Coding:
  - Modify the SFU Arithmetic coding to encode the sequence
  - Report average bits per pixel
  - Introduce quantization, for levels > 47, do uniform quantization with step size 5, i.e., levels \{48 49 50 51 52\} all represented by 50. report the average bits per pixel in this case, compute the reconstructed image quality in PSNR
  - For the given bi-level image, apply context (3 neighboring pixels) aware binary arithmetic coding, report bits/pixel.

<table>
<thead>
<tr>
<th>Category</th>
<th>Unary Code</th>
<th>Levels</th>
<th>Fixed Level Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>{0}</td>
<td>Nil</td>
</tr>
<tr>
<td>10</td>
<td>-1, +1</td>
<td>{0, 1}</td>
<td></td>
</tr>
<tr>
<td>110</td>
<td>-3, -2, +2, +3</td>
<td>{00 01 10 11}</td>
<td></td>
</tr>
<tr>
<td>1110</td>
<td>-4, -5, -6, -7, 4, 5, 6, 7</td>
<td>{000 001 010 011 100 101 110 111}</td>
<td></td>
</tr>
<tr>
<td>11110</td>
<td>-15, -14, -13, ... 13, 14, 15</td>
<td>{0000 0001 ... 1110 1111}</td>
<td></td>
</tr>
<tr>
<td>111110</td>
<td>-47, -46, ..., 46, 47</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
HW2 Bonus: DNA Sequence Compression

- Obtain the human mitochondria genome data:
  - % get dna seq data: human mitochondria genome
  seq1 = getgenbank('NC_012920','SequenceOnly',true);
  seq_len = length(seq1);
  h = [5124, 5181, 2169 4094]/seq_len;
  fprintf('
 estimated entropy=%1.2f, total bits=%d',getEntropy(h), getEntropy(h)*seq_len);

- Binarization
  - A -00, T-01, G-10, C-11, ignore N.

- Adaptive Arithmetic Coding
  - Report average symbol bit rate

- Context Adaptive Arithmetic Coding
  - Explore codon stats, and code at codon level with context.

- 2~3 students will be selected to do a project on Genome Coding based on the HW-2
Project 2: Point Cloud Compression

- **Point Cloud Data**
  - Geometry: \((x, y, z)\)
  - Color: \((n, r, g, b)\)
Point Cloud Geometry Compression

- Geometry:
  - Find bounding boxes for objects
  - Oct Tree decomposition to give scalable representation of the objects
Point Cloud Color Compression

- Color attributes compression:

DPCM [Kammer'12]
Colorization [Huang'06]
Octree Based [schnabel'06]
Based on Graph Transform [Zhang'14]
Mapping to jpeg image grid ??

Traverse octree → Write color attributes to an image grid → JPEG Encode

Table 1 scan order for mapping octree centroid colors to image grid (8x10 sample block)

<table>
<thead>
<tr>
<th>Order</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<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>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
<td>21</td>
<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>
</tr>
<tr>
<td></td>
<td>32</td>
<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>
<td>44</td>
<td>45</td>
<td>46</td>
<td>47</td>
</tr>
<tr>
<td></td>
<td>48</td>
<td>49</td>
<td>50</td>
<td>51</td>
<td>52</td>
<td>53</td>
<td>54</td>
<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>
</tr>
</tbody>
</table>

Achieved Compression Ratio

- Original (5 bits colors)
- Proposed (jpeg mapping)

Overall coding gain compared to legacy point cloud codec available in PCL (at comparable objective quality)
Resources

- **Source GitHub:**
  - https://github.com/RufaelDev/pcc-mp3dg

- **MPEG Doc:**
  - Rufael Mekuria, m36527 MPEG Point Cloud Compression: Data Set and Open Source. http://l.web.umkc.edu/lizhu/teaching/2016sp.video-communication/ref/m36527.zip

- **Related Papers:**

Outline

- Lecture 08 ReCap & HW-2
- Video Signal Processing Tricks
  - Motion Vector Coding
  - Intra Prediction
  - De-Blocking Filters
  - Scalable Coding - Prediction from Spatial Layers
- Video Codec Standards and software
Better MV prediction and coding:

- Use median of three neighboring MBs’ MVs
- Prediction error:
  - MV – median (left MV, top MV, top-right MV)
Advanced Prediction (4MV)

- 4MV: Each 8x8 block in a MB can have its own MV
- Suitable when there is complicated motion in the MB
- Need more bits to encode the MVs
- Need to compare the performance of 1 MV and 4MV to select the best mode: more optimizations
MV Coding in AVC/H.264

- **MV Prediction:**

  - Block D  Block B  Block C
  - \( \text{MV}_D \)  \( \text{MV}_B \)  \( \text{MV}_C \)
  - Block A  Current Block

  \[ \text{PMV} = \begin{cases} 
  \text{Median (MV}_A, \text{MV}_B, \text{MV}_C) & \text{(if Block C is available)} \\
  \text{Median (MV}_A, \text{MV}_B, \text{MV}_D) & \text{(otherwise)} 
\end{cases} \]

  Figure 1: motion vector prediction in H.264/AVC

- **MV residual VLC:**

  Code Table defined for MVD in H.264/AVC

<table>
<thead>
<tr>
<th>MVD</th>
<th>Code Word</th>
<th>Code length</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>01x</td>
<td>1</td>
</tr>
<tr>
<td>±1</td>
<td>0010x</td>
<td>3</td>
</tr>
<tr>
<td>±2</td>
<td>0011x</td>
<td>5</td>
</tr>
<tr>
<td>±3</td>
<td>000100x</td>
<td>7</td>
</tr>
<tr>
<td>±4</td>
<td>000101x</td>
<td>7</td>
</tr>
<tr>
<td>±5</td>
<td>000110x</td>
<td>7</td>
</tr>
<tr>
<td>±6</td>
<td>000111x</td>
<td>7</td>
</tr>
<tr>
<td>±7</td>
<td>00001001x</td>
<td>9</td>
</tr>
<tr>
<td>±8</td>
<td>00001000x</td>
<td>9</td>
</tr>
<tr>
<td>±9</td>
<td>00001001x</td>
<td>9</td>
</tr>
<tr>
<td>±10</td>
<td>00001010x</td>
<td>9</td>
</tr>
<tr>
<td>±11</td>
<td>00001011x</td>
<td>9</td>
</tr>
<tr>
<td>±12</td>
<td>00001100x</td>
<td>9</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Motion Vector Prediction & Coding in HEVC

- Spatial and Temporal Candidate MVs

- Prediction logic

Z. Li Multimedia Communciation, 2016 Spring
### MV coding modes

#### 3 coding modes

<table>
<thead>
<tr>
<th></th>
<th>AMVP</th>
<th>Merge</th>
<th>Skip</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax elements</td>
<td>mvp_l0_flag, mvp_l1_flag</td>
<td>merge_flag, merge_idx</td>
<td>cu_skip_flag, merge_idx</td>
</tr>
<tr>
<td>Use of neighbors candidates</td>
<td>Predict motion vector</td>
<td>Copy motion data (motion vector, reference index, direction)</td>
<td>Copy motion data (motion vector, reference index, direction); no residual</td>
</tr>
<tr>
<td>Number of Candidates</td>
<td>Up to 2</td>
<td>Up to 5 (signaled in slice header)</td>
<td></td>
</tr>
<tr>
<td>Spatial</td>
<td>Up to 2 of 5 (scaling if reference index different)</td>
<td>Up to 4 of 5 (no scaling, only redundancy check)</td>
<td></td>
</tr>
<tr>
<td>Temporal</td>
<td>Up to 1 of 2 (if &lt; 2 spatial candidates)</td>
<td>Up to 1 of 2 (always added to list if available)</td>
<td></td>
</tr>
<tr>
<td>Additional</td>
<td>Zero motion vector (if &lt; 2 spatial or temp candidates)</td>
<td>Bi-predictive candidates and zero motion vector</td>
<td></td>
</tr>
</tbody>
</table>
Merge Mode

- Merging MVs from small blocks to form large block shared MV saves bits

Ref:
Outline

- Lecture 08 ReCap & HW-2
- Video Signal Processing Tricks
  - Motion Vector Coding
  - Intra Prediction
  - De-Blocking Filters
  - Scalable Coding - Prediction from Spatial Layers
- Video Codec Standards and software
H.264 Intra-Frame Prediction

- Previous standards: only has DC prediction after transform
- H.264:
  - Spatial-domain prediction
  - More choices
  - Can be used in all slice types
- Four intra prediction modes:
  - Intra_4x4: Predict each 4x4 block separately
    - 9 possible modes
    - Suitable for areas with details
  - Intra_16x16: Predict the MB as a whole unit.
    - 4 possible modes
    - Efficient for smooth areas
  - Chroma prediction:
    - 4 possible modes
  - I_PCM: No prediction, raw samples are sent directly.
    - To limit the maximum number of bits for each block
- Prediction across slice boundary is not allowed.
**H.264 Intra-Frame Prediction**

- **Intra_4x4 Prediction: 9 modes**
  - Small block size enables more accurate spatial prediction.

![Neighbors used for prediction](image)

Current 4x4 block

- **Mode 0**: copy top row
- **Mode 3**: diagonal prediction
- **Mode 4**: copy diagonal

![Prediction Directions](image)  
(Mode 2: DC prediction)
H.265/HEVC Intra Prediction

- Much more modes
  - DC mode: copy DC values from neighbor
  - Planar mode: top row or left col average
  - Angular: pixels on certain line
Outline

- Lecture 08 ReCap & HW-2
- Video Signal Processing Tricks
  - Motion Vector Coding
  - Intra Prediction
  - De-Blocking Filters
  - Scalable Coding - Prediction from Spatial Layers
- Video Codec Standards and software
Deblocking Filter

- Reduce blocking artifact in the reconstructed frames
- Can improve both subjective and objective quality
- Filter in H.261:
  - \([1/4, 1/2, 1/4]\): Applied to non-block-boundary pixels in each block.
  - A low-pass smoothing filter.
- In H.264 (and H.263v2), this is used in the prediction loop to improve motion estimation accuracy. Decoder needs to do the same. Also called loop filter.

Before…. and After
Sample Adaptive Offset (SAO)

A nonlinear amplitude mapping is introduced within the inter prediction loop after the deblocking filter.

- Its goal is to better reconstruct the original signal amplitudes by using a look-up table that is described by a few additional parameters that can be determined by histogram analysis at the encoder side.
- First proposed by Samsung JCTVC-A124, Apr’10, but too complicated

Simplified by MediaTEK

- Band correction: Band Offset (BO)
- Divide all pixels into bands, based on pel values, each band has one offset.
- Extreme Correction: Edge Offset (EO)
- Classify pixels by edge info: each category has one offset.
- Each region can be enhanced by either EO or BO:

  Up to 6% and 17% rate saving for Y and CbCr, with
  Little change of complexity.
How to find the best offset?

- s(k): original pixel          x(k): recon after deblocking filter
- Recon MSE after Deblock Filtering:

\[ \varepsilon_{rec} = \sum_{k \in K} (x(k) - s(k))^2 \]

- After adding an offset \( a_c \) to each category, the MSE is:

\[ \varepsilon_{SAO} = \sum_{c \in C} \sum_{k \in C} ((x(k) + a_c) - s(k))^2 \]

- To find the best offset, taking derivative and setting to 0:

\[ a_c = \frac{1}{N_c} \sum_{k \in c} (s(k) - x(k)). \]

\[ \rightarrow \text{SAO tries to cancel the avg recon error in each category.} \]

Complexity is very low.
Sample Adaptive Offset (SAO) Filtering

- Classify pixels on block edge as one of the four categories

- Offset its pixel value accordingly

![Diagram showing four categories of pixel classification: 0-degree, 90-degree, 135-degree, 45-degree.](image)

- Positive offset
- Negative offset
SAO Effects

Before and After SAO

With SAO

No SAO

Original

Ref:

Outline

- Lecture 08 ReCap & HW-2
- Video Signal Processing Tricks
  - Motion Vector Coding
  - Intra Prediction
  - De-Blocking Filters
  - Scalable Coding - Prediction from Spatial Layers
- Video Codec Standards and software
Scalable video coding is desired in heterogeneous networks:

Types of scalability:

- Temporal Scalability
- Spatial Scalability
- Quality Scalability
Temporal scalability in H.264 SVC

- No change to H.264. The only related change in SVC refers to the signalling of temporal layers.

- Examples:

  **Hierarchical B or P pictures**

  **non-dyadic hierarchical prediction structure**

  **hierarchical prediction structure with a delay of zero**
Spatial scalability in H.264 SVC

- SVC follows the convention of multi-layer coding, which is also used in H.262/MPEG-2 Video, H.263, and MPEG-4 Visual.
- In each spatial layer, motion-compensated prediction and intra prediction are employed as for single-layer coding.
- In addition, SVC provides inter-layer predictions for spatial enhancement layers.
H.264 Scalable Coding Diagram

- **H264 SVC:**
  - Inter-Layer Prediction: motion, modes, texture, residual
  - SNR scalability: Coarse, Medium and Fine Granular Scalability
Spatial scalability in H.264 SVC

- A new MB type is provided: base_mode_flag
- **1. Inter-layer motion prediction:**
  - If base_mode_flag == 1 and the reference layer MB is inter-coded, the MB partition, reference frame index, and MV info are derived from the co-located 8x8 block in the reference layer. The MV is scaled by 2.
- **2. Inter-layer intra prediction:**
  - If base_mode_flag == 1 and the reference layer MB is intra-coded, the prediction for the MB in the enhancement layer is obtained by upsampling the reference 8x8 block (4-tap filter for luma and bilinear filter for chroma)
Spatial scalability in H.264 SVC

3. Inter-layer residual prediction:
   - Can be used for all inter-coded MBs
   - The residual signal of the co-located 8x8 block in the ref layer is upsampled using a bilinear filter and used as prediction for the residual signal of the enhanced layer MB.

Summary: Each spatial enhanced layer is decoded with a single motion compensation loop:
   - For the reference layers, only the intra-coded MBs, the residual blocks and MVs that are used for inter-layer prediction need to be reconstructed.
   - The computationally expensive operations of motion-compensated prediction and the deblocking of inter-picture predicted MBs only need to be performed for the target layer to be displayed.
Spatial scalability in H.264 SVC

Inter-layer prediction examples:

Left: upsampling of intra-coded MB for inter-layer intra prediction,
Middle: upsampling of MB partition for inter-layer prediction of MB modes,
Right: upsampling of residual signal for inter-layer residual prediction.

http://ip.hhi.de/imagecom_G1/savce/index.htm
SNR/Quality scalability in H.264 SVC

- Quality scalability can be viewed as a special case of spatial scalability with same picture sizes for base and enh. Layers:
  - This is called coarse-grain scalability (CGS)

- When utilizing inter-layer prediction, a refinement of texture information is typically achieved by re-quantizing the residual texture signal in the enh layer with a smaller quantization step size than that of the preceding CGS layer.

- The CGS concept only allows a few selected bit rates to be supported in a scalable bit stream.

http://ip.hhi.de/imagecom_G1/savce/index.htm
SNR/Quality scalability in H.264 SVC

- Medium-grain scalability (MGS):

- Differences to the CGS:
  - A modified high-level signalling, which allows a switching between different MGS layers in any access unit
  - Key picture, which allows the adjustment of the trade-off between drift and enh layer coding efficiency.

http://ip.hhi.de/imagecom_G1/savce/index.htm
SNR/Quality scalability in H.264 SVC

- Different trade-offs between enh layer coding efficiency and drifting:

  - (a) **Base layer only control** in MPEG-4 Visual. No drift.
  - (b) **Enh layer only control** in H.262/MPEG-2. Highest efficiency, lowest robustness.
  - (c) **Two loop**: enh layer can be affected by drifting
  - (d) **SVC key picture**: Each picture can select different ref mode. Drifting is limited within 2 key pictures.
SVC Performance


SVC can provide a suitable degree of scalability at the cost of only 10% bit rate increase in comparison to the bit rate of single-layer H.264/AVC coding.

(a) Soccer, CIF 30Hz

(b) Soccer, CIF 30Hz - 4CIF 30Hz

Z. Li Multimedia Communication, 2016 Spring
HEVC Scalability

- **Design Philosophy**
  - Utilize block level prediction mechanism, *ref_idx*, treat across layer prediction as if long range prediction
  - Only need reference layer reconstructed picture and motion vectors, not bit stream syntax

- **Scalability Features**

<table>
<thead>
<tr>
<th>Scalability features</th>
<th>Scalable standard</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SVC</td>
<td>SHVC</td>
</tr>
<tr>
<td>Temporal</td>
<td>X</td>
<td>X (in HEVC)</td>
</tr>
<tr>
<td>Spatial</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>SNR (quality)</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Hybrid codec</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Bit depth</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Color gamut</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Feature</th>
<th>SVC</th>
<th>SHVC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inter-layer prediction signaling</td>
<td>Add flag in macroblock</td>
<td>Reuse ref_idx in PU</td>
</tr>
<tr>
<td>Types of inter-layer prediction supported</td>
<td>Block-level syntax pred., texture, motion, mode, and residual pred.</td>
<td>Texture and motion pred.</td>
</tr>
<tr>
<td>Decoding type</td>
<td>Single-loop</td>
<td>Multi-loop</td>
</tr>
<tr>
<td>Spatial scalability ratio</td>
<td>Limited to 2x and 1.5x in Scalable Baseline</td>
<td>Arbitrary ratio</td>
</tr>
<tr>
<td>Spatial resampling phase</td>
<td>Fixed phase position</td>
<td>Arbitrary phase adjustment</td>
</tr>
<tr>
<td>Backward compatibility</td>
<td>AVC-coded BL only</td>
<td>HEVC or non-HEVC coded BL</td>
</tr>
<tr>
<td>Decoder design</td>
<td>BL decoder needs new API</td>
<td>EL decoder cannot directly reuse AVC decoder</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Can repurpose existing HEVC decoder</td>
</tr>
</tbody>
</table>
SHEVC Performance

- Compare with HEVC Simulcast (2 streams)
  - BL QPs: (26, 30, 34, and 38 for SNR scalability, and 22, 26, 30, and 34 for spatial scalability)
  - EL QPs: Spatial: \{0, +2\}
  - EL QPs: SNR \{-4, -6\}

- Average Performance is tabulated
  - Saving of up to 27% over simulcast
  - The closer the resolution, the better the gain
  - But if two embedded stream frame size far off, e.g., embed CIF within UHD, not worth of it.

### Table VI

<table>
<thead>
<tr>
<th>Seq.</th>
<th>Random access SNR</th>
<th>Low delay SNR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2x</td>
<td>1.5x</td>
</tr>
<tr>
<td>A1</td>
<td>-14.2%</td>
<td>N/A</td>
</tr>
<tr>
<td>A2</td>
<td>-22.6%</td>
<td>N/A</td>
</tr>
<tr>
<td>B1</td>
<td>-23.7%</td>
<td>-35.0%</td>
</tr>
<tr>
<td>B2</td>
<td>-15.2%</td>
<td>-27.0%</td>
</tr>
<tr>
<td>B3</td>
<td>-14.2%</td>
<td>-26.1%</td>
</tr>
<tr>
<td>B4</td>
<td>-17.8%</td>
<td>-29.7%</td>
</tr>
<tr>
<td>B5</td>
<td>-7.8%</td>
<td>-17.3%</td>
</tr>
<tr>
<td>Avg</td>
<td>-16.5%</td>
<td>-27.0%</td>
</tr>
</tbody>
</table>
References


- The Scalable Video Coding Amendment of the H.264/AVC Standard (http://ip.hhi.de/imagecom_G1/savce/index.htm)


Summary

Video Signal Processing II:
- Motion Vector Coding
  - Prediction from spatio-temporal neighboring MVs
  - Sophisticated merging mode for HEVC to combine MVs from smaller coding blocks.
- Intra Prediction
  - Synthesize block data from angular pixel values
  - Many modes to decide from, complexity in R-D optimization
- Deblocking
  - Smoothing by low pass filter across coding blocks
  - SAO – Sample Adaptive Offset, a novel new tool from HEVC with huge coding gains. Post processing of Deblocking
- Scalability
  - Temporal Prediction: B frames, Hierarchical B frames
  - Spatial Prediction: texture, residual prediction
  - SNR/Spatial Scalability: coding the difference between different QP and size layers
  - Temporal Scalability: frame loss induced distortion, how to manage.

Point Cloud Compression
- MPEG work!