EE5359
MULTIMEDIA PROCESSING
Spring 2014

Project Interim Report

Intra Prediction Efficiency and Performance Comparison of HEVC and VP9

Under guidance of
DR K R RAO
DEPARTMENT OF ELECTRICAL ENGINEERING
UNIVERSITY OF TEXAS AT ARLINGTON

Presented by
Shruti Sukumaran
UT ID 1000870264
shruti.sukumaran@mavs.uta.edu
4/26/2014
Contents
1. Overview of HEVC........................................................................................................4
2. Comparison with VP9 – A contender........................................................................5
3. How does HEVC work?...............................................................................................6
4. HEVC intra frame coding..........................................................................................8
5. VP9 Intra-prediction .................................................................................................11
6. Experiments...............................................................................................................14
7. Results for Video Test Sequences [9].........................................................................14

List of Abbreviations
AVC Advanced Video Coding
CABAC Context Adaptive Binary Arithmetic Coding
CTB Coding Tree Blocks
CTU Coding Tree Units
HEVC High Efficiency Video Coding
ISO International Organization for Standardization
ITU-T International Telecommunication Union - Telecommunication Standardization Sector
JCT-VC Joint collaborative team on video coding
JM Joint model software
MC Motion Compensation
MI Mode Info
MPEG Moving Picture Experts Group
MSE Mean square error
MV Motion Vector
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAL</td>
<td>Network Abstraction Layer</td>
</tr>
<tr>
<td>PSNR</td>
<td>Peak signal to noise ratio</td>
</tr>
<tr>
<td>SSIM</td>
<td>Structural similarity index</td>
</tr>
<tr>
<td>TU</td>
<td>Transform Units</td>
</tr>
<tr>
<td>TM</td>
<td>True Motion</td>
</tr>
<tr>
<td>VCEG</td>
<td>Video Coding Experts Group</td>
</tr>
</tbody>
</table>
Abstract

The rapid growth and insatiable demand for consumption of video over the internet continues to stress the limits of the available bandwidth [9]. The emerging of a more efficient next generation video coding standard is of a high demand at present. To address this, ISO/IEC/MPEG/ITU-T VCEG developed a next generation video coding standard called HEVC (High Efficiency Video Coding) starting from current generation codec H.264/AVC [8]. Likewise Google embarked on a project in the late 2011 to develop next generation open source video codec VP9 starting from its predecessor VP8 as the baseline [8]. They seem to be two main contenders for the position of the next state-of-the-art video compression standards[10]. The announced aim of HEVC is to achieve twice more efficient compression compared to H.264/AVC, at the same visual quality and VP9 was developed to get half the bit-rate of VP8 with royalty-free video codec [10]. Intra-prediction is one of the main features which determines the compression efficiency of the entire codec. This proposal is a detailed comparison of intra-prediction modes in HEVC and VP9. Experiments will be carried out using JCT-VC HM [16] [17] and WebM VP9 [19] encoders. The analysis will focus on the intra frame coding and different image performance metrics like MSE, PSNR, bit rate, BD-Metrics[25], SSIM [22][23][24] [25] and hence video quality will be evaluated for high resolution videos.

1. Overview of HEVC

High Efficiency Video Coding (HEVC) is a new Standard for video compression developed by the ISO and ITU-T. The Moving Picture Experts Group (MPEG) and Video Coding Experts Group (VCEG) set up a Joint Collaborative Team on Video Coding (JCT-VC) to create the new standard. HEVC is a joint publication of ISO/IEC and ITU-T, formally known as ISO/IEC 23008-2 and ITU-T Recommendation H.265. The new HEVC standard received first stage approval in January 2013.[1]

The H.264/MPEG-AVC standard successfully achieved an increase of about 50% in coding efficiency compared to its predecessor H.262/MPEG-2 Video [5]. The H.265/MPEG-HEVC standard was designed to be applicable for almost all existing H.264/MPEG-AVC applications, while putting Emphasis on high-resolution video coding. Since the development process of H.265/MPEG, HEVC was also driven by the most recent scientific and technological achievements in the field of video coding. Dramatic bit-rate savings were achieved for substantially the same visual quality, when compared to its predecessor like H.264/MPEG-AVC [5].
2. Comparison with VP9 – A contender

In parallel with the open video coding standardization processes of ITU-T and ISO/IEC, a few companies individually developed their own video codecs, which often were based partly on their own secretly kept technologies and partly on variants of the state-of-the-art technologies used in their standardized counterparts, available at that time. One of these kind of proprietary video codecs is the VP8 codec which was developed privately by On2 Technologies® Inc. that in turn, was later acquired by Google® Inc. Based on P8, Google® Inc. started the development of its successor VP9 in 2011, which was recently announced to be finalized [5].

There seem to be two main contenders for the position of the next state-of-the-art video compression standard: JCT-VC H.265/HEVC and Google VP9. HEVC is being developed by JCT-VC group -the creators of AVC. It is an evolution of AVC concepts with some innovations. On the other hand, VP9 expands techniques used in AVC and VP8 and is very likely to replace AVC at least in the YouTube video service [5].

H.265 and VP9 support 8K content as well and with physical media on the wane, this makes them quite frankly the future of television and video, which is why they are so important [5].

For example, H.264 could grab a 16x16 ‘macroblock’ of pixels and perform nine ‘intra-prediction directions’ - that allowed the pixels to be rebuilt within each block. H.265 can grab 64x64 ‘superblocks’ and perform 35 infra-prediction directions to rebuild them. Like H.264, H.265 varies the size of blocks it takes. For example, it would take much smaller blocks (down to 4x4 pixels) around detailed areas like facial features and much bigger blocks of the sky or a relatively plain background. VP9 is similar on the surface. It can also take 64x64 superblocks, but unlike H.265 these do not need to be square so it can sample 64x32 or 4x8 blocks for greater efficiency. On the flip side it only has 10 prediction modes to rebuild them. Cynics argue VP9 changes H.265 just enough for it to avoid copyright infringement. Needless to say both standards require more computational power than H.264 and VP8 for all this rebuilding [5].
3. How does HEVC work?

Figure 1: HEVC encoder block diagram [11]

Figure 2: HEVC decoder block diagram [11]
An encoding algorithm producing an HEVC compliant bitstream would typically proceed as follows.

- The input video is divided into block-shaped regions called Coding Tree Units (CTUs) and fed in raster-scan order to the HEVC encoder, with the exact block partitioning being conveyed to the decoder [3].
- HEVC is a hybrid encoding standard, i.e. both spatial (Intra) and temporal (Inter) redundancies are exploited for compressing the input video [3].
- The first picture of a video sequence (and the first picture at each clean random access point into a video sequence) is coded using only \textit{intra-picture prediction} (that uses some prediction of data spatially from region-to-region within the same picture, but has no dependence on other pictures) [3].
- For all the remaining pictures or between random access points, typically \textit{inter-prediction} is employed using motion estimation and motion compensation. The encoding process for inter-picture prediction consists of choosing motion data comprising the selected reference picture and motion vector (MV) to be applied for predicting the samples of each block [3].
- After searching for the best Intra and Inter predictions of the current CTU, the residual signal is computed as the difference of the current video data and the prediction [4].
- The residual is transformed by a linear spatial transform and scaled, quantized, entropy coded using the context adaptive binary arithmetic coder (CABAC), and transmitted to the decoder device along with the prediction information [4].
- For generation of identical predictions of the subsequent pictures for future reference, the encoder has a partial decoder in the reconstruction loop [4]. Therefore, the quantized transform coefficients are constructed by inverse scaling and are then inverse transformed to duplicate the decoded approximation of the residual signal [4].
- The residual is then added to the prediction, and the result of that addition may then be fed into one or two loop filters (Deblocking and Sample Adaptive Offset (SAO) filters) to smooth out artifacts induced by block-wise processing and quantization [4].
- The final picture representation (that is a duplicate of the output of the decoder) is stored in a \textit{decoded picture buffer} to be used for the prediction of subsequent pictures [3].
- In general, the order of encoding or decoding processing of pictures often differs from the order in which they arrive from the source; necessitating a distinction between the decoding order (i.e., bitstream order) and the output order (i.e., display order) for a decoder [3].
4. HEVC intra frame coding

Instead of macroblocks like in previous standards, HEVC pictures are divided into so-called coding tree blocks (CTB). CTB appear in the picture in raster order. Depending on the stream parameters, they are 64x64, 32x32 or 16x16. Each CTB can be split recursively in a quad-tree structure, all the way down to 8x8. So for example a 32x32 CTB can consist of three 16x16 and four 8x8 regions. These regions are called coding units, or CUs. CUs are the basic unit of prediction in HEVC. The CUs in a CTB are traversed and coded in Z-order. Figure 3 shows an example of ordering in a 64x64 CTB[6].

Like H.264/AVC, HEVC uses block-based intra prediction to take advantage of spatial correlation within a picture. HEVC follows the basic idea of H.264/AVC intra prediction
but makes it far more flexible. For example, H.264 could grab a 16x16 ‘macroblock’ of pixels and perform nine ‘intra-prediction directions’ - that allowed the pixels to be rebuilt within each block (Figure 5) HEVC can grab 64x64 ‘superblocks’ and perform 35 intra-prediction directions to rebuild them. The 33 luma intra prediction modes for HEVC are as shown in Figure 4. HEVC also includes a planar intra prediction mode, which is useful for predicting smooth picture regions. In planar mode, the prediction is generated from the average of two linear interpolations (horizontal and vertical).

![Figure 4: Available prediction directions in the unified intra prediction in HM 1.0 [11]](image)
Furthermore, intra prediction can be implemented at different block sizes, ranging from 4X4 to 64X64 (whatever size the PU has) (Figure 6).

Figure 5: H.264 Intra-prediction modes [20]

Figure 6: Subdivision of a CTB into CBs (and transform block (TBs)). Solid lines indicate CB boundaries and dotted lines indicate TB boundaries (a) CTB with its partitioning. (b) Corresponding quad tree [3].
5. VP9 Intra-prediction

VP9 divides the picture into 64x64-sized blocks called super blocks - SB for short. Superblocks are processed in raster order: left to right, top to bottom. This is the same as most other codecs. Superblocks can be subdivided down, all the way to 4x4. The subdivision is done with a recursive quadtree just like HEVC. But unlike HEVC, a subdivision can also be horizontal or vertical only. In these cases the subdivision stops. Although 4x4 is the smallest “partition”, lots of information is stored at 8x8 granularities only, a MI (mode info) unit. This causes blocks smaller than 8x8 to be handled as sort of a special case. For example a pair of 4x8 intra coded blocks is treated like an 8x8 block with two intra modes. Figure 7 is an example of partitioning of a super block [7].

Figure 4: VP9 Superblock partitioning
VP9 is similar to HEVC on the surface. It can also take 64x64 superblocks, but unlike H.265 these do not need to be square so it can sample 64x32 or 4x8 blocks for greater efficiency. On the flip side it only has 10 prediction modes to rebuild them. Cynics argue VP9 [19] changes H.265 just enough for it to avoid copyright infringement. Needless to say both standards require more computational power than H.264 and VP8 for all this rebuilding [8]. The encoder block diagram for VP6 is shown in Figure 8.
At block-size 4x4, VP9 supports ten intra prediction modes; DC, Vertical, Horizontal, TM (True Motion), Horizontal Up, Left Diagonal, Vertical Right, Vertical Left, Right Diagonal, and Horizontal Down (the same set defined by VP8). For blocks from 8x8 to 64x64 there is also support for ten intra modes; DC, Vertical, Horizontal, TM, and six angular predictors corresponding, approximately, to angles of 27, 45, 63, 117, 135, and 153 degrees. Furthermore, there is additionally the option of applying a low-pass filter to the prediction that can be signaled in the bitstream [15].
6. Experiments

Analysis will be performed using various image quality measurements such as:

- Objective quality measures-MSE, PSNR [23][24]
- Subjective quality measures- SSIM [25]
- BD-PSNR [25]
- BD-Bit Rate [25]
- Computational Complexity

**MSE and PSNR[23][24]**

MSE and PSNR for a NxM pixel image are defined as

\[
MSE = \frac{1}{M \times N} \sum_{m=1}^{M} \sum_{n=1}^{N} [o(m,n) - r(m,n)]^2
\]

\[
PSNR = 10 \log_{10} \frac{L^2}{MSE}
\]

where ‘o’ is the original image and ‘r’ is the reconstructed image. M and N are the width and height of an image and \(L\) is the maximum pixel value in the NxM pixel image.

7. Results for Video Test Sequences [9]

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Resolution</th>
<th>Frame Rate(fps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BlowingBubbles</td>
<td>416x240</td>
<td>30</td>
</tr>
</tbody>
</table>
Results

Table 1: Table shows the captured metrics for VP9 – 10 frames (Blowingbubbles_30fps_416X240)

<table>
<thead>
<tr>
<th>QP</th>
<th>QP=37</th>
<th>QP=32</th>
<th>QP=27</th>
<th>QP=22</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR(dB)</td>
<td>31.567</td>
<td>34.38</td>
<td>37</td>
<td>41.51</td>
</tr>
<tr>
<td>Bitrate(kbps)</td>
<td>5969.36</td>
<td>6861.776</td>
<td>7384.384</td>
<td>9753.64</td>
</tr>
<tr>
<td>Time(sec)</td>
<td>104.22</td>
<td>117.48</td>
<td>126.30</td>
<td>137.60</td>
</tr>
</tbody>
</table>

Table 2: Table shows the captured metrics for HEVC- 10 frames Blowingbubbles_30fps_416X240

<table>
<thead>
<tr>
<th>QP</th>
<th>QP=37</th>
<th>QP=32</th>
<th>QP=27</th>
<th>QP=22</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR(dB)</td>
<td>38.855</td>
<td>40.992</td>
<td>42.868</td>
<td>43.839</td>
</tr>
<tr>
<td>Bitrate(kbps)</td>
<td>5857</td>
<td>6992.448</td>
<td>8411.844</td>
<td>9227.245</td>
</tr>
<tr>
<td>Time(sec)</td>
<td>132.54</td>
<td>191.34</td>
<td>223.1</td>
<td>257.56</td>
</tr>
</tbody>
</table>

Figure 10: RD Plot - 10 frames Blowingbubbles_30fps_416X240
<table>
<thead>
<tr>
<th>Sequence</th>
<th>Resolution</th>
<th>Frame Rate (fps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cactus</td>
<td>1920×1080</td>
<td>30</td>
</tr>
</tbody>
</table>

**Table 3:** Table shows the captured metrics for VP9 – 10 frames Cactus_30fps_1920×1080

<table>
<thead>
<tr>
<th>QP</th>
<th>PSNR (dB)</th>
<th>Bitrate (kbps)</th>
<th>Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>22</td>
<td>41.122</td>
<td>56539.66</td>
<td>15452</td>
</tr>
<tr>
<td>27</td>
<td>39.123</td>
<td>40335.68</td>
<td>13210</td>
</tr>
<tr>
<td>32</td>
<td>39.123</td>
<td>31092.58</td>
<td>10422</td>
</tr>
<tr>
<td>37</td>
<td>35.55</td>
<td>17030.57</td>
<td>10103</td>
</tr>
</tbody>
</table>

**Table 4:** Table shows the captured metrics for HEVC - 10 frames Cactus_30fps_1920×1080

<table>
<thead>
<tr>
<th>QP</th>
<th>PSNR (dB)</th>
<th>Bitrate (kbps)</th>
<th>Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>22</td>
<td>41.202</td>
<td>55357.09</td>
<td>18732</td>
</tr>
<tr>
<td>27</td>
<td>40.576</td>
<td>48748.65</td>
<td>16958</td>
</tr>
<tr>
<td>32</td>
<td>39.568</td>
<td>40239.89</td>
<td>16221</td>
</tr>
<tr>
<td>37</td>
<td>36.168</td>
<td>17090.58</td>
<td>16031</td>
</tr>
</tbody>
</table>
Figure 11: RD Plot - 10 frames Cactus_30fps_1920×1080

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Resolution</th>
<th>Frame Rate(fps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PeopleOnStreet</td>
<td>2560X1600</td>
<td>30</td>
</tr>
</tbody>
</table>

Table 5: Table shows the captured metrics for HEVC- 10 frames PeopleOnStreet_30fps_2560X1600

<table>
<thead>
<tr>
<th></th>
<th>QP=22</th>
<th>QP=27</th>
<th>QP=32</th>
<th>QP=37</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR(dB)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HEVC</td>
<td>44.8084</td>
<td>41.6998</td>
<td>36.514</td>
<td>38.4149</td>
</tr>
<tr>
<td>VP9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bitrate(kbps)</td>
<td>102898.87</td>
<td>52019.2</td>
<td>40220.45</td>
<td>33006.77</td>
</tr>
<tr>
<td>Time(sec)</td>
<td>2390.967</td>
<td>1723.980</td>
<td>1803.541</td>
<td>1397.272</td>
</tr>
<tr>
<td>QP</td>
<td>QP=22</td>
<td>QP=27</td>
<td>QP=32</td>
<td>QP=37</td>
</tr>
<tr>
<td>--------</td>
<td>-----------</td>
<td>-----------</td>
<td>-----------</td>
<td>-----------</td>
</tr>
<tr>
<td>PSNR (dB)</td>
<td>43.219</td>
<td>40.387</td>
<td>36.514</td>
<td>35.951</td>
</tr>
<tr>
<td>Bitrate (kbps)</td>
<td>99891.504</td>
<td>61032.631</td>
<td>32031.78</td>
<td>18799.884</td>
</tr>
<tr>
<td>Time (sec)</td>
<td>1709.44</td>
<td>1391.31</td>
<td>1304.33</td>
<td>1136.98</td>
</tr>
</tbody>
</table>

Table 6: Table shows the captured metrics for VP9 - 10 frames PeopleOnStreet_30fps_2560X1600

Figure 12: RD Plot - 10 frames PeopleOnStreet_30fps_2560X1600
<table>
<thead>
<tr>
<th>Sequence</th>
<th>QP = 22</th>
<th>QP = 27</th>
<th>QP = 32</th>
<th>QP = 37</th>
<th>BD-BR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cactus</td>
<td>17.3</td>
<td>41.5</td>
<td>50.8</td>
<td>51.3</td>
<td>-40.3</td>
</tr>
<tr>
<td>Traffic</td>
<td>41.0</td>
<td>48.5</td>
<td>51.0</td>
<td>51.3</td>
<td>-50.1</td>
</tr>
<tr>
<td>Blowing Bubbles</td>
<td>6.7</td>
<td>16.1</td>
<td>40.3</td>
<td>41.8</td>
<td>-25.7</td>
</tr>
<tr>
<td>Park scene</td>
<td>39.8</td>
<td>43.2</td>
<td>43.7</td>
<td>47.5</td>
<td>-42.9</td>
</tr>
<tr>
<td>People On Street</td>
<td>16.8</td>
<td>25.7</td>
<td>27.4</td>
<td>35.7</td>
<td>-26.4</td>
</tr>
<tr>
<td>Averages</td>
<td>26.2</td>
<td>37.3</td>
<td>46.45</td>
<td>47.9</td>
<td>-40.5</td>
</tr>
</tbody>
</table>

Figure 7: HEVC Bit-Rate Savings (BD-BR) for the same PSNR

The average BD-BR savings of the HEVC encoder relative to VP9 is 40.5% As it is also observed from Table, the bitrate savings, on average, are increasing along with an increase of quantization parameters for VP9 encoder. Table provides a full summary of the BD-BR results, where negative BD-BR values indicate bitrate savings which indicate the required overhead in bit-rate to achieve the same PSNR-YUV values.

8. Conclusions

The aim of this project is to experiment and present a thorough study, implementation and exhaustive comparison of intra-frame only of the contending video coding standards HEVC and VP9 based on different performance metrics like MSE, PSNR, BD Metrics etc. for different video coding sequences [9]. VP9 needs 40.5% more bit rate than HEVC for the same PSNR. HEVC has 27%-33% more encoding time than VP9.
REFERENCES


[2] Iain Richardson/Vcodex.com “HEVC An Introduction to High Efficiency Video Coding” 2013


[16] HEVC JCT-VC HM reference software https://hevc.hhi.fraunhofer.de/svn/svn_TMuCSoftware/tags/HM-1.0_HM-1.0


[23] MSE http://en.wikipedia.org/wiki/Mean_squared_error


Special Issues on HEVC


Latest Paper on HEVC


[37] MPL Website: http://www-ee.uta.edu/dip/
[38] BD metrics code - http://www.mathworks.com/matlabcentral/fileexchange/27798-bjontegaardmetric/content/bjontegaard.m